ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University
(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-I
Course: COMPUTER PROGRAMMING (MSCS-1-01T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

M.Sc. (Computer Science) Programme Coordinator:

Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:

Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Faculty of School of Business Management & Commerce:

Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Faculty of School of Social Sciences and Liberal Arts:

Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA
A State University Established by Govt. of Punjab vide Act No. 19
of 2019 and Approved Under section 2(f) of UGC

PROGRAMME COORDINATOR
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

COURSE COORDINATOR AND EDITOR:
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Course: Computer Programming	
Course Code: MSCS-1-01T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Understand the basic language implementation techniques.
CO2	Develop C programs to demonstrate the applications of derived data types such as arrays, pointers, strings and functions.
CO3	Understand the concept of object oriented programming language.
CO4	Develop ability to learn and write small programs in C and C++.
CO5	Understand the concepts of OOPs including inheritance.

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

Name of Programme: M.Sc. (Computer Science)
Name of Course: Computer Programming
Course Code: MSCS-1-01T
Semester: 1st

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

OBJECTIVE

After Completion of the course, learners will be able to design, implement, and evaluate a computer-based system, process, components, or program to meet desired needs.

INSTRUCTIONS FOR THE PAPER SETTER/EXAMINER

1. The syllabus prescribed should be strictly adhered to.
2. The question paper will consist of three Sections: A, B, and C. Sections A and B will have four questions from the respective sections of the syllabus and will carry 10 marks each. The candidates will attempt two questions from each section.
3. Section C will have ten short answer questions covering the entire syllabus. Each question will carry 3 marks. All questions of section C will compulsory.
4. The examiner shall give a clear instruction to the candidates to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.
5. The duration of the paper will be three hours.

INSTRUCTIONS FOR THE CANDIDATES:

Candidates are required to attempt any two questions each from the sections A and B of the question paper and all ten short questions from Section C. They have to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.

SECTION - A

Unit I: Basics of Problem Solving in C Language: Problem Solving with Computers, Evolution of C Language, Character Set in C, Tokens, Keywords, Identifier, Constants, Variables, Rules for defining Variables, Data Types in C Language: Basic data type, Derived data type and Enum data type.

Unit II: Operators and Expressions in C: Operators in C, Types of Operator: Arithmetic, Relational, Logical, Comma, Conditional, Assignment, Operator Precedence and Associativity in C, Input and Output Statements, Assignment statements.

Unit III: Control Structure in C: Sequential Flow Statement, Conditional Flow Statement, Decision Control statements: if, if-else, nested-if, else-if ladder. Loop control statements: While, do-while, for loop, Nested of Loops. Case Control Statements: Switch Statement, goto Statement, Break Statement, Continue Statement.

Unit IV: Arrays and Pointers in C: Arrays, Characteristic of Arrays, Representation, Declaration and Initialization of an Array, Types of Arrays: one dimensional, multi-dimensional arrays. Pointer, Pointers Declaration and Initialization, Types of Pointers, Pointer Expressions and Pointer Arithmetic.

SECTION - B

Unit V: Functions and Strings in C: Function in C, Function Declaration and Definition, Types of Functions, Library Vs. User-defined Functions, Function Calling Methods, Function Parameters: Actual Parameter, Formal Parameter, Parameter Passing Techniques: Call by Value and Call by Reference, Recursive Function, Pointers and Functions. Strings: C Strings, Difference between char array and string literal, Traversing String, Accepting string as the input, Pointers with strings, String Functions.

Unit VI: User Defined Data types: Structure, Structure Variables Declaration, Accessing Structure Data Members, Array of Structures, Nested of Structure, Passing structure to function, Structures Limitations, Union, Difference between Structure and Union in C.

Unit VII: Object Oriented Programming: Need of an Object-Oriented Programming, C++ and its Applications, OOPs Concepts in C++: Class, Objects, Encapsulation, Abstraction, Polymorphism, Inheritance, Dynamic Binding and Message Passing. Access Specifiers in C++: Private, Protected and Public.

Unit VIII: Constructor and Inheritance in C: Constructor in C++, Characteristics of Constructors, Difference between constructor and member function, Types of Constructors:

Default Constructor, Parameterized Constructors, Copy Constructors, Dynamic Constructors, Destructor in C++, Difference between Constructor and Destructor. Inheritance in C++, Modes of Inheritance, Type of Inheritance: Single inheritance, Multiple inheritance, Hierarchical inheritance, Multilevel inheritance, Hybrid inheritance.

Reference Books:

- E.Balagurusamy,"Programming in C",Tata McGraw Hill.
- Kamthane,"Programming with ANSI and Turbo C",Pearson Education
- Rajaraman,V,"Fundamentals of Computers" , PHI
- Kanetkar,"Let Us C",BPB Publications.
- HerbertSchildt,"The Complete Reference C++",Tata McGraw-Hill.
- Deiteland Deitel,"C++ How to Program", Pearson Education.
- Robert Lafore,"Object Oriented Programming in C++",Galgotia Publications.
- Bjarne Straustrup,"The C++ Programming Language", Addition-Wesley Publication Co.
- Stanley B. Lippman, Josee Lajoie,"C++ Primer", Pearson Education.
- E.Balagurusamy,"Object Oriented Programming with C++", TataMcGraw-Hill

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- A

UNIT 1: BASIC OF PROBLEM SOLVING IN C LANGUAGES

1.1 Problem Solving with Computers

Computer based problem solving is a systematic process of designing, implementing and using programming tools during the problem solving stage. The following six steps must be followed to solve a problem using computer.

1. **Problem analysis:** It is the process of defining a problem and decomposing overall system into smaller parts to identify possible inputs, processes and outputs associated with the problem.
2. **Program design:** The second stage in problem solving using computer cycle is program design. This stage consists of preparing algorithms, flowcharts and pseudocodes. Generally, this stage intends to make the program more user friendly, feasible and optimized.
3. **Coding:** In this stage, process of writing actual program takes place. A coded program is most popularly referred to as a source code. The coding process can be done in any language (high level and low level). The actual use of computer takes

place in this stage in which the programmer writes a sequence of instructions ready for execution. Coding is also known as programming.

4. **Compilation and Execution:** In step 3 coding is done in either high level language or low level language (assembly language). For the computer to understand these languages, they must be translated into machine level language. The translation process is carried out by a compiler/interpreter (for high level language) or an assembler (for assembly language program).
5. **Debugging and Testing:** Debugging is the process of finding errors and removing them from a computer program, otherwise they will lead to failure of the program. Debugging is generally done by program developer. Further, testing is performed to verify that whether the completed software package functions or works according to the expectations defined by the requirements. Testing is generally performed by testing team which repetitively executes program with intent to find error.
6. **Program Documentation:** The program documentation is the process of collecting information about the program. The documentation process starts from the problem analysis phase to debugging and testing.

1.1 Evolution of C Language

C programming language was developed in 1972 by Dennis Ritchie at bell laboratories of AT&T (American Telephone & Telegraph), located in the U.S.A. Dennis Ritchie is known as the founder of the c language. The root of all modern languages is ALGOL (Algorithmic Language). ALGOL was the first computer programming language to use a block structure, and it was introduced in 1960. In 1967, Martin Richards developed a language called BCPL (Basic Combined Programming Language). BCPL was derived from ALGOL. In 1970, Ken Thompson created a language using BCPL called B. Both BCPL and B programming languages were typeless. After that, C was developed using BCPL and B by Dennis Ritchie at the Bell lab in 1972. So, in terms of history of C language, it was used in mainly academic environments, but at long last with the release of many C compilers for commercial use and the increasing popularity of UNIX, it began to gain extensive support among professionals. The following table highlights the evolution of C Language.

Sr. No.	Development Year	Language Name	Developer Name

1	1960	Algol	International Group
2	1967	BCPL	Martin Richard
3	1970	B	Ken Thompson
4	1972	Traditional C	Dennis Ritchie
5	1978	K & R C	Kernighan & Dennis Ritchie
6	1989	ANSI C	ANSI Committee
7	1990	ANSI/ISO C	ISO Committee

1.3 Character Set in C

The set of characters that are used to represent words, numbers and expression in C language is called C character set. The combination of these characters form words, numbers and expression in C. The characters in C are grouped into the following four categories.

- Letters or Alphabets
- Digits
- Special Characters
- White Spaces

Type of Character	**Characters**
Lowercase Alphabets	a to z
Uppercase Alphabets	A to Z
Digits	0 to 9

Special Characters	` ~ @ ! $ # ^ * % & ( ) [ ] { } < > + = _ – | / \ ; : ‘ “ , . ?
White Spaces	Blank Spaces, Carriage Return, Tab, New Line

1.4 Tokens

The individual elements of a program are called Tokens. In a C program, a number of individual units or elements occur and these elements are termed as C Tokens. In C programming language, the following 6 types of tokens are available:

- Keywords
- Identifiers
- Constant
- Operators
- Strings
- Special Characters

Tokens in C are a fundamental part of the programming language. They are the smallest individual units. Tokens provide functionality for users to interact easily with the compiler.

1.5 Keywords

Keywords are predefined words for a C programming language. All keywords have fixed meaning and these meanings cannot be change. They serve as basic building blocks for program statements. A list of 32 keywords in the C language is given below:

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
Int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

1.6 Identifiers

C identifier is a name used to identify a variable, function, or any other user-defined item. An identifier starts with a letter A to Z, a to z, or an underscore '_' followed by zero or more letters, underscores, and digits (0 to 9). C does not allow punctuation characters such as @, $, and % within identifiers.

Rules for constructing C identifiers

- The first character of an identifier should be either an alphabet or an underscore, and then it can be followed by any of the character, digit, or underscore.
- Identifiers should not begin with any numerical digit.
- Identifiers are case sensitive i.e. both uppercase and lowercase letters are distinct.
- Commas or blank spaces cannot be specified within an identifier.
- Keywords cannot be represented as an identifier.
- The length of the identifiers should not be more than 31 characters.
- Identifiers should be meaningful, short, and easy to read.

1.7 Constants

A constant is basically a named memory location in a program that holds a single value throughout the execution of that program. It can be of any data type- character, string, floating-point, double, and integer, etc. A constant is a value or variable that can't be changed in the program.

There are two ways to define constant in C programming.

- **Using const keyword:** The ‘const’ keyword is used to create a constant of any given datatype in a program. For creating a constant, there is need to prefix the declaration of the variable with the ‘const’ keyword.

  Syntax: const datatype constantName = value;

  Example: const float PI=3.14**;**

- **Using #define pre-processor:** To define the constants ‘#define’ pre-processor directive can also be used. It must be defined in the very beginning of the program as all the preprocessor directives must be defined before the global declaration.

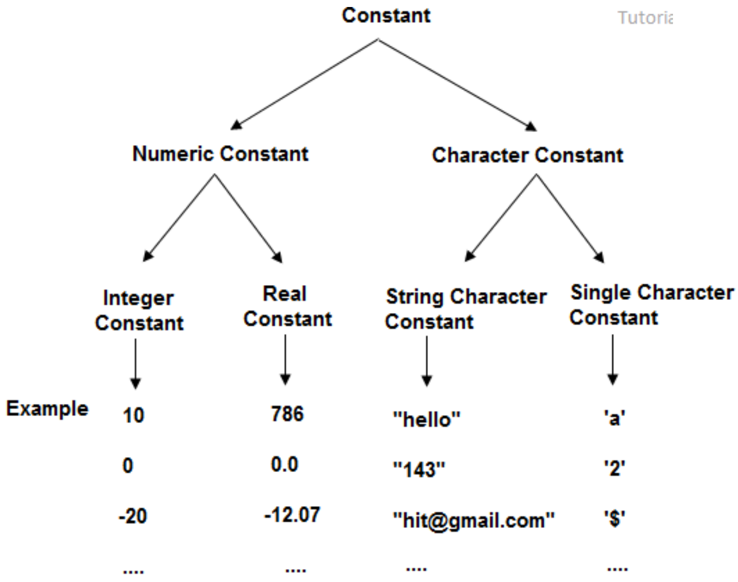
  Syntax: #define identifierName value

  Example: #define PI 3.14

A constant is very similar to variables in the C programming language, but it can hold only a single variable during the execution of a program.

NOTE: Literals are the constant values assigned to the constant variables.

Types of Constant in C



- **Integer Constants:** An integer constant is a numeric constant (associated with number) without any fractional or exponential part. There are three types of integer constants in C programming:
  - Decimal Constant(base 10)
  - Octal Constant(base 8)
  - Hexadecimal Constant(base 16)
- **Floating point/Real constants:** A floating point constant is a numeric constant that has either a fractional form or an exponent form.
  - Examples of Real constants in decimal form are:

    2.2, +8.0, -4.15
  - Examples of Real constants in exponential notation are:

    +1e23, -9e2, +3e-15
- **Character Constants:** The character constants are symbols that are enclosed in one single quotation. The maximum length of a character quotation is of one character only. Example:

  'B'

  '5'

  '+'

- **String Constants:** The string constants are a collection of various special symbols, digits, characters, and escape sequences that get enclosed in double quotations.

  The definition of a string constant occurs in a single line:

  “Programming in C”

- **Escape Sequences or Backslash Character Constants:** These are some types of characters that have a special type of meaning in the C language. These types of constants must be preceded by a backslash symbol so that the program can use the special function in them. Below mentioned is a list of all the special characters used in the C language:

Meaning of Character	**Backslash Character**
Backspace	\b
New line	\n
Form feed	\f
Horizontal tab	\t
Carriage return	\r
Single quote	\’
Double quote	\”
Vertical tab	\v
Backslash	\\
Question mark	\?
Alert or bell	\a
Hexadecimal constant (Here, N – hex.dcml cnst)	\XN
Octal constant (Here, N is an octal constant)	\N

1.8 Variables

A variable is a name given to the memory location that is used to store data. The value of the variable can be changed, and it can be reused several times. It is a way to represent memory location through symbol so that it can be easily identified.

Declaration of Variables: Variables are the storage areas in a code that the program can easily manipulate. Every variable in C language has some specific type- that determines the layout and the size of the memory of the variable, the range of values that the memory can hold, and the set of operations that one can perform on that variable.

***Syntax*:** type variable_list;

***Example*:** **int** a1;
float b1;
char c1;

Here, a1, b1, c1 are variables of int, float and char data types respectively.

Initialization of Variables: Variables can be initialized (assigned an initial value) in their declaration. The initializer consists of an equal sign followed by a constant expression as follows:

***Syntax*:** type variable_name = value;

***Example*:** int a = 31, b = 15; // declaration and initialization of variables a, b

1.9 Rules for defining Variables

- A variable can have alphabets, digits, and underscore.
- No whitespace is allowed within the variable name.
- Variables should not be declared with the same name in the same scope.
- A variable name must not be any reserved word or keyword, e.g. char, float, etc.
- A variable name can start with the alphabet, and underscore only. It can't start with a digit.
- Maximum length of variable is 8 characters depend on compiler and operation system.

For Example

```
int a1; // valid declaration
int 1a; // invalid declaration – the name of the variable should not start using  a number
int roll_no; // valid declaration
int roll$no // invalid declaration – no special characters allowed
char break; // keywords cannot be used as variable name.
int roll no; // invalid declaration – there must be no spaces in the name of the variable
```

NOTE: C is case-sensitive language. Here, MARKS, Marks and marks are three different variables.

```
int MARKS; // it is a new variable
int Marks; // it is a new variable
int marks; // it is a new variable
```

Types of Variables in C: Based on the name and the type of the variable, the variables can be of the following basic types:

- **Global Variable**: A variable that gets declared outside a block or a function is known as a global variable. Any function in a program is capable of changing the value of a global variable. Hence, the global variable will be available to all the functions in the code. Because the global variable in C is available to all the functions, it is declared at the beginning of a block.

  Example:

  ```
  int code=30; //  global variable
  void function1()
    {
      int a=20; //  local variable
    }
  ```

- **Local Variable:** A local variable is a type of variable that is declared inside a block or a function, unlike the global variable. Hence, it is mandatory to declare a local variable in C at the beginning of a given block.

Example:

```
void function1()
 {
int a=10; //  local variable
  }
```

A local variable has to be initialized in a code before we use it in the program.

- **Automatic Variable:** Every variable that is declared inside a block (in the C language) is by default automatic in nature. Automatic variable can be declared explicitly by using the keyword ***auto***.

  Example:

  ```
  void main(){
   int a1=810; //  local variable (implicitly automatic variable)
   auto int b1=510; // an automatic variable
   }
  ```

- **Static Variable:** The static variable in C language is declared using the ***static*** keyword. This variable retains the given value between various function calls.

  Example:

  ```
  void function1()
  {
   int m=10; //  Local variable
  static int n=10; // Static variable
  m=m+1;
  n=n+1;
  printf("%d,%d",m,n);
  }
  ```

  Output:

  Local variable m will print the value 11 for every function call.
  Static variable n will print the value that is incremented in each and every function call i.e. 11, 12, 13 and so on

- **External Variable:** A user can share a variable in multiple numbers of source files in C by using an external variable. The keyword ***extern*** *is used to declare* an external variable.

  Syntax:

  extern int a=10;// external variable (also a global variable)

1.10 Data Types in C Language

Data types in c refer to an extensive system used for declaring variables or functions of different types. The type of a variable determines how much space it occupies in storage and how the bit pattern stored is interpreted.

Types of Data Types in C

Data Type	**Example of Data Type**
Primary/ Basic Data Type	Floating-point, integer, double, character, void
Derived Data Type	Union, structure, array, etc.
Enumerated Data Type	Enums

1.10.1 Basic/Primary Data Types: There are the primitive or primary data types in C programming language:

- **Integer:** Integers are whole numbers that can have both zero, positive and negative values but no decimal values. For example, 0, -5, 10.

```
// C program to print Integer data types
#include <stdio.h>
int main()
{
// Integer variable with positive data.
int a1 = 19;

// integer variable with negative data.
int b1 = -91;

printf("Integer variable with positive data: %d\n", a1);
printf("Integer variable with negative data: %d\n", b1);

return 0;
}
```

Output:

```
Integer variable with positive data: 19

Integer variable with negative data: -91
```

- **Character:** This data type is used to store only a single character. The storage size of the character is 1. It is the most basic data type in C. It stores a single character and requires a single byte of memory in almost all compilers.

```
// C program to print Character data types.
#include <stdio.h>
 int main()
  {
   char ch = 'a';
   char c;
   printf("Value of ch: %c\n", ch);
   ch++;
    printf("Value of ch after increment is: %c\n", ch);

    // c is assigned ASCII values
    // which corresponds to the
    // character 'c'
    // a-->97 b-->98 c-->99
    // here c will be printed
    c = 99;
    printf("Value of c: %c", c);
    return 0;
}
```

Output:

```
Value of ch: a
Value of ch after increment is: b
Value of c: c
```

- **Floating-point:** In C programming float data type is used to store floating-point values. Float in C is used to store decimal and exponential values. It is used to store decimal numbers (numbers with floating point values) with single precision.

```
// C program to print Float data types
#include <stdio.h>
int main()
{
    float a1 = 5.0f;
    float b1 = 12.5f;
      // 2x10^-4
    float c1 = 2E-4f;
      printf("%f\n",a1);
      printf("%f\n",b1);
      printf("%f",c1);
  return 0;
}
```

Output:

```
                    5.000000

                    12.500000

                    0.000200
```

- **Double:** A Double data type in C is used to store decimal numbers (numbers with floating point values) with double precision. It is used to define numeric values which hold numbers with decimal values in C.

```
// C program to print double data types
#include <stdio.h>
 int main()
{
    double a1 = 23125623.00;
    double b1 = 2.267823;
    double c1 = 2312312312.123123;
    printf("%lf\n", a1);
    printf("%lf\n", b1);
    printf("%lf", c1);
    return 0;
}
```

Output:

```
23125623.00

2.267823

2312312312.123123
```

- **Void:** The void data type in C is used to specify that no value is present. It does not provide a result value to its caller. It has no values and no operations. It is used to represent nothing. Void is used in multiple ways as function return type, function arguments as void, and pointers to void.

```
// function return type void

void exit(int check);


// Function without any parameter can accept void.

int print(void);
```

Range of Values of Basic Data Types in C

Data Type	**Format Specifier**	**Minimal Range**	**Memory Size (in bits)**
char	%c	-127 to 127	8
int	%d	-32,767 to 32,767	16 or 32
float	%f	1E-37 to 1E+37 along with six	32

		digits of the precisions here	
double	%lf	1E-37 to 1E+37 along with six digits of the precisions here	64

1.10.2 Derived Data Types

Data types that are derived from fundamental data types are derived types. For example: arrays, pointers, function types, structures, etc. These are the data type whose variable can hold more than one value of similar type. In C language it can be achieve by array.

```
int m[] = {10,20,30};  // valid
int n[] = {100, 'A', "ABC"};   //  invalid
```

The C language supports a few derived data types. These are:

- **Arrays** – The array basically refers to a sequence (ordered sequence) of a finite number of data items from the same data type sharing one common name.
- **Function** – A Function in C language refers to a self-contained block of single or multiple statements. It has its own specified name.
- **Pointers** – The Pointers in C language refer to some special form of variables that one can use for holding other variables' addresses.
- **Unions** – Unions data types are very similar to the structures. It is used to store objects of various different types in the very same location of the memory. It means that in any program, various different types of union members would be able to occupy the very same location at different times.
- **Structures** – A collection of various different types of data type items that get stored in a contagious type of memory allocation is known as structure in C language.

```
struct student
{
int roll_no;
char name[15];
float marks;
}
```

1.10.3 Enum Data Types

Enumeration is a user defined data type in C language. It is mainly used to assign names to integral constants, the names make a program easy to read and maintain. The keyword ‘enum’ is used to declare new enumeration types in C language.

Syntax: enum flag {int_const1, int_const2,.....int_constN};

In the above declaration, enum named as flag is defined containing 'N' integer constants. The default value of int_const1 is 0, int_const2 is 1, and so on. The default value of the integer constants can be changed at the time of the declaration.

```
// Program to demonstrate working of enum data type

#include<stdio.h>
 enum week{Mon, Tue, Wed, Thur, Fri, Sat, Sun};
 int main()
{
    enum week day;
    day = Wed;
    printf("%d",day);
    return 0;
}
```

Output:

```
2
```

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- A

UNIT 2: OPERATORS AND EXPRESSION IN C

2.1 Operators in C:

The operators are simply a symbol that can be used to perform operations. In simpler words, we can also say that an operator is a type of symbols that inform a compiler to perform specific mathematical, conditional, or logical functions. For example, + and - are the operators to perform addition and subtraction in any C program. C has many operators that almost perform all types of operations. These operators are really useful and can be used to perform every operation. Basically, operators serve as the foundations of the programming languages. Thus, the overall functionalities of the C programming language remain incomplete if we do not use operators.

2.2 Types of Operators:

Various types of operators are available in the C language that performs different types of operations.

I. Arithmetic Operators

II. Relational Operators

III. Logical Operators

IV. Comma Operators
V. Conditional Operators
VI. Assignment Operators

2.2.1 Arithmetic Operators:

An arithmetic operator performs mathematical operations such as addition, subtraction, multiplication, division etc on numerical values (constants and variables). We can also say that, it helps a user to perform the mathematical operations as well as the arithmetic operations in a program, such as subtraction (-), addition (+), division (/), multiplication (*), the remainder of division (%), decrement (–), increment (++).

Further, arithmetic operators are divided into two types:

Unary Operators: Operators that operate or work with a single operand are unary operators. For example: Increment(++) and Decrement(–) Operators.

Let's look at an example for demonstrating the working of increment and decrement operator:

// Examples of increment and decrement operators:

```
#include <stdio.h>
int main()
{
   int a = 11, b = 90;
   float c = 100.5, d = 10.5;
   printf("++a = %d \n", ++a);
   printf("--b = %d \n", --b);
   printf("++c = %f \n", ++c);
   printf("--d = %f \n", --d);
   return 0;
}
```

Output:

```
++a = 12
--b = 89
++c = 101.500000
--d = 9.500000
```

In the above code example, the increment and decrement operators ++ and -- have been used as prefixes. Note that these two operators can also be used as postfixes like a++ and a-- when required.

Binary Operators: Operators that operate or work with two operands are binary operators. For example: Addition(+), Subtraction(-), multiplication(*), Division(/) operators. Let’s look at an example of binary Arithmetic operations in C below assuming variable a holds 7 and variable b holds 5.

```
// Examples of arithmetic operators in C
#include <stdio.h>
int main()
{
    int a = 7,b = 5, c;
     c = a+b;
    printf("a+b = %d \n",c);
    c = a-b;
    printf("a-b = %d \n",c);
    c = a*b;
    printf("a*b = %d \n",c);
    c = a/b;
    printf("a/b = %d \n",c);
    c = a%b;
    printf("Remainder when a is divided by b = %d \n",c);
    return 0;
}
```

Output:

```
a+b = 12
a-b = 2
a*b = 35
a/b = 1
Remainder when a divided by b = 2
```

2.2.2 Relational Operators:

Rational Operators are used for the comparison of the values of two operands. For example, checking if one operand is equal to the other operand or not, whether an operand is greater than the other operand or not, etc. Some of the relational operators are (==, >= , <=)(See this article for more reference). In other words, relational operators are specifically used to compare two quantities or values in a program. It checks the relationship between two operands. If the given relation is true, it will return 1 and if the relation is false, then it will return 0. Relational operators are heavily used in decision-making and performing loop operations.

The table below shows all the relational operators supported by C.

Operator	**What it does**	**Example**
==	Equal to	5==5 will be 1
>	Greater than	5>6 will be 0
<	Less than	6<7 will be 1
>=	Greater than equal to	2 >= 1 will be 1
<=	Less than equal to	1 <= 2 will be 1
!=	Not equal to	5 != 6 will be 1

Below is an example showing the working of the relational operator:

// Example of relational operators

```
#include <stdio.h>
int main()
{
int x = 8, y = 10;
printf("%d == %d is False(%d) \n", x, y, x == y);
printf("%d != %d is True(%d) \n ", x, y, x != y);
printf("%d > %d is False(%d)\n ",  x, y, x > y);
printf("%d < %d is True (%d) \n", x, y, x < y);
printf("%d >= %d is False(%d) \n", x, y, x >= y);
printf("%d <= %d is True(%d) \n", x, y, x <= y);
return 0;
}
```

Output:

```
8 == 10 is False(0)
8 != 10 is True(1)
8 > 10 is False(0)
8 < 10 is True(1)
8 >= 10 is False(0)
8 <=10 is True(1)
```

All the relational operators work in the same manner as described in the table above.

2.2.3 Logical Operators:

In the C programming language, we have three logical operators when we need to test more than one condition to make decisions. These logical operators are:

&& (meaning logical AND)

|| (meaning logical OR)

! (meaning logical NOT)

An expression containing a logical operator in C language returns either 0 or 1 depending upon the condition whether the expression results in true or false. Logical operators are generally used for decision-making in C programming.

The table below shows all the logical operators supported by the C programming language.

Operator	**What it does**
&& (Logical AND)	True only if all conditions satisfy.
|| (Logical OR)	True only if either one condition satisfies.
! (Logical Not)	True only if the operand is 0.

Following is the example that easily elaborates the working of the logical operator:-

```
#include <stdio.h>
int main()
{
  int i = 5, j = 5, k = 10, final;
  printf("i is equal to j or k greater than j is is %d \n", (i == j) && (k > j));
  printf("i is equal to j or k less than j is %d \n", (i == j) || (k < j));
   printf("i not equal to j or k less than j is %d \n", (i != j) || (k < j));
   return 0;
}
```

Output:

```
i is equal to j or k greater than j is 1
i is equal to j or k less than j is 1
i not equal to j or k less than j is 0
```

2.2.4 Comma Operators:

Comma Operators are used for separating expressions, variable declarations, function calls etc. It works on two operands. It is a binary operator. Comma acts as a separator.

Syntax of comma operator:-

int a=1, b=2, c=3, d=4;

2.2.5 Conditional Operators:

Conditional or ternary operator is used to construct the conditional expression. A conditional operator pair "?:"

Syntax of Conditional Operators:

exp1 ? exp2 : exp3

Here exp1, exp2, exp3 are expressions.

The Operator ?: works as follows: exp1 is evaluated first. If it is true, then the expression exp2 is evaluated and becomes the value of the expression. If exp1 is false, then exp3 is evaluated and its value becomes the value of the expression.

Example of Conditional Operator:

```
    #include <stdio.h>
    int main()
    {
    int number=13;
    (number>14)? (printf("It is greater than number 14!")) : (printf("It is less than number 14!"));  // conditional operator
    return 0;
}
```

Output:

```
It is less than number 14!
```

If we set the number to 15 then it will give the output⇒ It is greater than number 14!

2.2.6 Assignment Operators:

An assignment operator is mainly responsible for assigning a value to a variable in a program. Assignment operators are applied to assign the result of an expression to a variable. This operator plays a crucial role in assigning the values to any variable. The most common assignment operator is =. C language has a collection of shorthand assignment operators that can be used for C programming. The table below lists all the assignment operators supported by the C language:

Operator	**Example**
=	a=b or b=a
+=	a += b or a = a+b
-=	a -=b or a = a-b
*=	a *= b or a = a*b
/=	a /= b or a = a/b
%=	a %= b or a = a%b

The below example explains the working of assignment operator:

```
#include <stdio.h>
int main()
{
  int a = 99, result;
  result = a;
  printf("Welcome to TechVidvan Tutorials...\n");
  printf("Value of result = %d\n", result);
  result += a;    // or result = result + a
  printf("Value of result = %d\n", result); // After Addition
  result -= a;     // or result = result - a
  printf("Value of result = %d\n", result); // After Subtraction
  result *= a;    // or result = result * a
  printf("Value of result = %d\n", result); // After Multiplication
  result /= a;     // or result = result / a
  printf("Value of result = %d\n", result);
  return 0;
}
```

Output:

```
Welcome to TechVidvan Tutorials…
Value of result = 99
Value of result = 198
Value of result = 99
Value of result = 9801
Value of result = 99
```

Arithmetic Expressions:

An arithmetic expression is an expression that consists of operands and arithmetic operators. An arithmetic expression computes a value of type int, float or double. When an expression contains only integral operands, then it is known as pure integer expression when it contains only real operands, it is known as pure real expression, and when it contains both integral and real operands, it is known as mixed mode expression.

Let's understand through an example.

6*2/ (2+1 * 2/3 + 6) + 8 * (8/4)

Evaluation of expression	Description of each operation
6*2/(2+1 * 2/3 +6) +8 * (8/4)	An expression is given.
6*2/(2+2/3 + 6) + 8 * (8/4)	2 is multiplied by 1, giving value 2.
6*2/(2+0+6) + 8 * (8/4)	2 is divided by 3, giving value 0.
6*2/ 8+ 8 * (8/4)	2 is added to 6, giving value 8.
6*2/8 + 8 * 2	8 is divided by 4, giving value 2.
12/8 +8 * 2	6 is multiplied by 2, giving value 12.
1 + 8 * 2	12 is divided by 8, giving value 1.
1 + 16	8 is multiplied by 2, giving value 16.
17	1 is added to 16, giving value 17.

2.3 Operator Precedence and Associativity

2.3.1 Operator Precedence

Operator Precedence in C is used to determine the sequence in which different operators will be evaluated if two or more operators are present in an expression. The associativity of operators is used to determine whether an expression will be evaluated from left-to-right or from right-to-left if there are two or more operators of the same precedence.

Operator precedence controls how terms in an expression are grouped and how an expression is evaluated. Certain operators take precedence over others. The multiplication operator, for example, takes priority over the addition operator.

For example, x = 2 + 3 * 5;

Here, the value of x will be assigned as 17 and not 20. The “*”operator has higher precedence than the “+” operator. So the first 3 is multiplied by 5 to get 15, and then 2 is added to 15 to result in 17.

2.3.2 Operator Associativity

The direction in which an expression is evaluated is determined by the associativity of operators. Associativity is utilized when two operators of the same precedence exist in an expression. Associativity can be either left to right or right to left.

For example, consider x = 5 / 3 * 3;

Here, the value of x will be assigned as 3 and not 5. ‘*’ operator and ‘/’ operator have the same precedence, but their associativity is from Left to Right. So first 5 is divided by 3 to get 1, and then 1 is multiplied by 3, resulting in 3.

The following table explain the order of Precedence and Associativity in C Language.

Operator	**Order of Precedence**	**Associativity**
.	Direct member selection	Left to right
->	Indirect member selection	Left to right
[]	Array element reference	Left to right
()	Functional call	Left to right
~	Bitwise(1’s) complement	Right to left
!	Logical negation	Right to left
–	Unary minus	Right to left
+	Unary plus	Right to left
—	Decrement	Right to left
++	Increment	Right to left
*	Pointer reference	Right to left
&	Dereference (Address)	Right to left
(type)	Typecast (conversion)	Right to left
sizeof	Returns the size of an object	Right to left

%	Remainder	Left to right
/	Divide	Left to right
*	Multiply	Left to right
–	Binary minus (subtraction)	Left to right
+	Binary plus (Addition)	Left to right
>>	Right shift	Left to right
<<	Left shift	Left to right
>	Greater than	Left to right
<	Less than	Left to right
>=	Greater than or equal	Left to right
<=	Less than or equal	Left to right
==	Equal to	Left to right
!=	Not equal to	Left to right
^	Bitwise exclusive OR	Left to right
&	Bitwise AND	Left to right
||	Logical OR	Left to right
|	Bitwise OR	Left to right
?:	Conditional Operator	Right to left
&&	Logical AND	Left to right

,	Separator of expressions	Left to right
=	Simple assignment	Right to left
/=	Assign quotient	Right to left
*=	Assign product	Right to left
%=	Assign remainder	Right to left
-=	Assign difference	Right to left
+=	Assign sum	Right to left
|=	Assign bitwise OR	Right to left
^=	Assign bitwise XOR	Right to left
&=	Assign bitwise AND	Right to left
>>=	Assign right shift	Right to left
<<=	Assign left shift	Right to left

2.4 Input and Output Statements:

Input and Output statement are used to read and write the data in C programming. These are embedded in stdio.h (standard Input/Output header file). Input means to provide the program with some data to be used in the program and Output means to display data on screen or write the data to a printer or a file.C programming language provides many built-in functions to read any given input and to display data on screen when there is a need to output the result.There are mainly two of Input/Output functions are used for this purpose. These are discussed as:

3.1 Unformatted I/O functions

3.2. Formatted I/O functions

2.4.1. Unformatted I/O functions: There are mainly six unformatted I/O functions discussed as follows:

a) getchar()
b) putchar()
c) gets()
d) puts()
e) getch()
f) getche()
g) getchar()

This function is an Input function. It is used for reading a single character from the keyboard. It is a buffered function. Buffered functions get the input from the keyboard and store it in the memory buffer temporally until you press the Enter key.

The general syntax is as: v = getchar();

where v is the variable of character type.

For example:

```
char n;
n = getchar();
```

A simple C-program to read a single character from the keyboard is as:

```
/*To read a single character from the keyboard using the getchar() function*/
#include <stdio.h>
main()
{
char n;
n = getchar();
}
```

- **putchar():** This function is an output function. It is used to display a single character on the screen. The general syntax is as: putchar(v);

  where v is the variable of character type, For example:

  char n;

putchar(n); A simple program is written as below, which will read a single character using getchar() function and display inputted data using putchar() function:

```
/*Program illustrate the use of getchar() and putchar() functions*/
#include <stdio.h>
main()
{
char n;
n = getchar();
putchar(n);
}
```

- **gets()**: This function is an input function. It is used to read a string from the keyboard. It is also a buffered function. It will read a string when you type the string from the keyboard and press the Enter key from the keyboard. It will mark null character ('\0') in the memory at the end of the string when you press the enter key. The general syntax is as:

  gets(v);

  where v is the variable of character type. For example:

  char n[20];

  gets(n);

  A simple C program to illustrate the use of gets() function:

```
/*Program to explain the use of gets() function*/
#include <stdio.h>
main()
{
char n[20];
gets(n);
}
```

puts()

This is an output function. It is used to display a string inputted by gets() function. It is also used to display a text (message) on the screen for program simplicity. This function appends a newline ("\n") character to the output.

The general syntax is as:

```
puts(v);
or
puts("text line");
```

where v is the variable of character type.

A simple C program to illustrate the use of puts() function:

```
/*Program to illustrate the concept of puts() with gets() functions*/
#include <stdio.h>
main()
{
char name[20];
puts("Enter the Name");
gets(name);
puts("Name is :");
puts(name);
}
```

The Output is as follows:

```
Enter the Name      Geek
Name is:            Geek
```

- **getch():** This is also an input function. This is used to read a single character from the keyboard like getchar() function. But getchar() function is a buffered is function, getchar() function is a non-buffered function. The character data read by this function is directly assigned to a variable rather it goes to the memory buffer, the character data is directly assigned to a variable without the need to press the Enter key.

Another use of this function is to maintain the output on the screen till you have not press the Enter Key. The general syntax is as:

```
v = getch();
```

where v is the variable of character type.

A simple C program to illustrate the use of getch() function:

```
/*Program to explain the use of getch() function*/
#include <stdio.h>
main()
{
char n;
puts("Enter the Char");
n = getch();
puts("Char is :");
putchar(n);
getch();
}
```

The output is as follows:

```
Enter the Char
Char is L
```

getche()

All are same as getch(0 function execpt it is an echoed function. It means when you type the character data from the keyboard it will visible on the screen. The general syntax is as:

```
v = getche();
```

where v is the variable of character type.

A simple C program to illustrate the use of getch() function:

```
/*Program to explain the use of getch() function*/
#include <stdio.h>
main()
{
char n;
puts("Enter the Char");
n = getche();
puts("Char is :");
putchar(n);
getche();
}
```

The output is as follows:

```
Enter the Char L
Char is L
```

Formatted I/O functions

Formatted I/O functions which refers to an Input or Ouput data that has been arranged in a particular format. There are mainly two formatted I/O functions discussed as follows:

scanf()

printf()

scanf()

The scanf() function is an input function. It used to read the mixed type of data from keyboard. You can read integer, float and character data by using its control codes or format codes. The general syntax is as:

```
scanf("control strings",arg1,arg2,..............argn);
                        or
scanf("control strings",&v1,&v2,&v3,................&vn);
```

Where arg1,arg2,……….argn are the arguments for reading and v1,v2,v3,……..vn all are the variables.

The scanf() format code (spedifier) is as shown in the below table:

Example Program:

```
/*Program to illustrate the use of formatted code by using the formatted scanf() function */
#include <stdio.h>
main()
{
char n,name[20];
int abc;
float xyz;
printf("Enter the single character, name, integer data and real value");
scanf("\n%c%s%d%f", &n,name,&abc,&xyz);
getch();
}
```

printf()

This ia an output function. It is used to display a text message and to display the mixed type (int, float, char) of data on screen. The general syntax is as:

```
printf("control strings",&v1,&v2,&v3,................&vn);
                                        or
printf("Message line or text line");
```

Where v1,v2,v3,……..vn all are the variables.

The control strings use some printf() format codes or format specifiers or conversion characters.

Example Program:

```
/*Below the program which show the use of printf() function*/
#include <stdio.h>
main()
{
int a;
float b;
char c;
printf("Enter the mixed type of data");
scanf("%d",%f,%c",&a,&b,&c);
getch();
}
```

2.5 Assignment Statements:

An Assignment statement is a statement that is used to set a value to the variable name in a program. C provides an assignment operator for this purpose, assigning the value to a variable using assignment operator is known as an assignment statement in C. The function of this operator is to assign the values or values in variables on right hand side of an expression to variables on the left hand side.

The syntax of the assignment expression

```
Variable = constant / variable/ expression;
```

The data type of the variable on left hand side should match the data type of constant/variable/expression on right hand side with a few exceptions where automatic type conversions are possible.

Examples of assignment statements:

```
b = c ; /* b is assigned the value of c */
a = 9 ; /* a is assigned the value 9*/
b = c+5; /* b is assigned the value of expr c+5 */
\
```

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- A

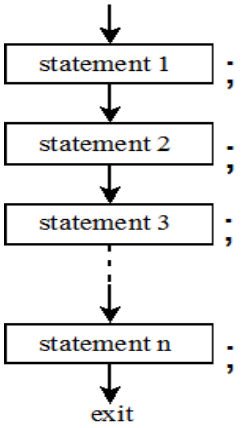
UNIT 3: CONTROL STRUCTURE IN C

3.1 Control Structure:

Control Structures are just a way to specify flow of control in programs. Any algorithm or program can be clearer and understood if they use self-contained modules called as logic or control structures. It basically analyses and chooses in which direction a program flows based on certain parameters or conditions. There are three basic types of flow of control known as Sequential flow, Conditional flow, Iteration flow. The sequential flow is un-conditional flow of control, but the next two are types of conditional statements. The description about each are given below:

3.1.1 Sequential Flow Statement

Sequential flow as the name suggests follows a serial or sequential flow in which the flow depends on the series of instructions given to the computer. Unless new instructions are given, the modules are executed in the obvious sequence. The sequences may be given, by means of numbered steps explicitly. Also, implicitly follows the order in which modules are written. Most of the processing, even some complex problems, will generally follow this elementary flow pattern. The following figure highlights the flow of sequential flow statements.



The Process start with statement 1 follow with statement 2, next follow with statement 3 and same follow till statement n in sequential manner. Therefore when the program which flows only from top to bottom without changing the flow of control are known as sequential control statements.

3.1.2 Conditional Flow Statement

Sometimes, it is desirable to alter the sequence of the statements in the program depending upon certain circumstances. Repeat a group of statements until certain specified conditions are met. This involves a kind of decision making to see whether a particular condition has occurred or not and direct the computer to execute certain statements accordingly. Based on application and the specific requirement, it is necessary to:

(i) To alter the flow of a program

(ii) Test the logical conditions

(iii) Control the flow of execution as per the selection these conditions can be placed in the program using decision-making statements.

In simple words, Control statements in C help the system to execute a certain logical statement and decide whether to enable the control of the flow through a certain set of statements or not. Based on the conditions and flow of execution, control statement is classified into three categories named as:

- Decision Control Statements

- Loop Control Statements
- Case Control Statements

3.2 Decision Control statements:

There come situations when we need to make some decisions, on which we can decide what we should do next. Similar situations arise in programming also where we need to make some decisions and based on these decisions we will execute the next block of code. In conditional control, the execution of statements depends upon the condition-test. If the condition evaluates to true, then a set of statements is executed otherwise another set of statements is followed. This control is also called Decision Control because it helps in making decision about which set of statements is to be executed. In C language, if x occurs then execute y else execute z. There can also be multiple conditions like in C if x occurs then execute p, else if condition y occurs execute q, else execute r. This condition of C else-if is one of the many ways of importing multiple conditions. The Decision Control Statements are used to evaluate the one or more conditions and make the decision whether to execute set of statement or not. Based on the hierarchy of conditions, the Decision Control Statements has five types of control statements:

- if Statement
- if-else Statement
- Nested if-else statement
- else-if Ladder

The detail about each statement is given below:

3.2.1 if statement:

Simple if statements are carried out to perform some operation when the condition is only true. If the condition of the if statement is true then the statements under the block is executed else the control is transferred to the statements outside the block directly and none of the statements will be executed. It is also called a one-way selection statement.

Syntax:

```
If (expression)
{
        //code to be executed
}
```

Following program illustrates the use of if construct in C-Language.

```
#include<stdio.h>
int main()
{
    int num1=1;
    int num2=2;
    if(num1<num2)                //test-condition
    {
        printf("num1 is smaller than num2");
    }
    return 0;
}
```

Output: “num1 is smaller than num2”

In the above program, we have initialized two variables with num1, num2 with value as 1, 2 respectively. Then, we have used if with a test-expression to check which number is the smallest and which number is the largest. We have used a relational expression in if construct. Since the value of num1 is smaller than num2, the condition will evaluate to true. Thus it will print the statement inside the block of If. After that, the control will go outside of the block and program will be terminated with a successful result.

3.2.2 If-else statement:

The single if statement may work pretty well, but in some situations, user may have to execute statements based on true or false under certain conditions and user may want to work with multiple variables or the extended conditional parameters, then the if-else statement is the optimum choice, by using the if statement, only one block of the code executes after the condition is true but by using the if-else statement, there are two possible blocks of code where the first block is used for handling the success part and the other one for the failure condition. It is also called two way selection statement.

Syntax:

```
if(expression)
```

```
{
 //Statements
}
else
{
 //Statements
}
```

As discussed above, the if statement is applicable to make one comparison, and if the user want to make comparison between two variables, then only if-else statement is applicable. The following program illustrates the use of if-else construct in C-Language.

```
#include<stdio.h>
int main()
{
     int num1=1;
     int num2=2;
     if(num1>num2)                    //test-condition
     {
             printf("num1 is Larger");
     }
     else
     {
             printf("num2 is Larger");
     }
     return 0;
}
```

Output: “num2 is Larger”

The above program is start with initialisation of two variable num1 and num2 with value 1 & 2. The if statement evaluate the condition part of both the values and verify that either num1 is greater than num2 or not? The num1 have value 1 and num2 is 2, then num1>num2 condition evaluates to false therefore, the else block is executed and the output of program is num2 is Larger.

3.2.3 Nested If statement:

We already saw how useful if and else statements are, but what if we need to check for more conditions even when one condition is satisfied? Then C Language provides an extended feature as Nested-if statement. Nested if statement in C is the nesting of if statement within another if statement and nesting of if statement with an else statement. Once an else statement gets failed there are times when the next execution of statement wants to return a true statement, there we need nesting of if statement to make the entire flow of code in a semantic order. Nested if statement in C plays a very pivotal role in making a check on the inner nested statement of if statement with an else very carefully. The following syntax highlights the functioning of nested if statement.

Syntax:

```
if (condition1)
{
  // Executes when condition1 is true
  if (condition2)
  {
    // Executes when condition2 is true
  }
}
```

How nested-if statement works?

- The if statement evaluates the test expression inside the parenthesis ().
- If the test expression is evaluated to true, statements inside the body of if are executed.
- If the test expression is evaluated to false, statements inside the body of if are not executed.

Following program illustrates the use of Nested-if construct in C-Language.

```
#include<stdio.h>
void main()
{
        int a, b, c;

        printf("Enter three numbers\n");
scanf("%d %d %d", &a, &b, &c);

if(a > b)
{
        if(a > c)
                        printf("a: %d is largest\n", a);
        else
                        printf("c: %d is largest\n", c);
}
else
{
        if(b > c)
                        printf("b: %d is largest\n", b);
        else
                        printf("c: %d is largest\n", c);


}
}
```

Output:

```
Enter three numbers 89      12      65
Output: a:89 is largest
```

3.2.4 Else-if Ladder: The nested if statement provides the feature to evaluate the expression of three variables, but when the user wants to evaluate among more than three, then C language also provide the feature in form of else-if ladder. The else-if ladder helps user decide from among multiple options. The C/C++ if statements are executed from the top down. As soon as one of the conditions controlling the if is true, the statement associated with that if is executed, and the rest of the C else-if ladder is bypassed. If none of the conditions is true, then the final else statement will be executed. The following syntax highlights the condition evaluation sections of else-if ladder statements.

Suntax:

```
if (Condition1)
{
    Statement1;
}
else if(Condition2)
{
    Statement2;
}
.
.
.
else if(ConditionN)
{
    StatementN;
}
else
{
    Default_Statement;
}
```

The if-else-if ladder statement is an extension to the if-else statement. It is used in the scenario where there are multiple cases to be performed for different conditions. In if-else-if ladder statement, if a condition is true then the statements defined in the if block will be executed, otherwise if some other condition is true then the statements defined in the else-if block will be executed, at the last if none of the condition is true then the statements defined in the else block will be executed. The following programs demonstrate the c-programme to evaluate the expression using else-if ladder statement.

```
#include<stdio.h>
void main ()
{
  int a,b,c,d;
  printf("Enter the values of a,b,c,d: ");
  scanf("%d%d%d%d",&a,&b,&c,&d);
        if(a>b && a>c && a>d)
        {
                printf("%d is the largest",a);
        }
        else if(b>c && b>a && b>d)
        {
                printf("%d is the largest",b);
        }
        else if(c>d && c>a && c>b)
        {
                printf("%d is the largest",c);
        }
        else
        {
                printf("%d is the largest",d);
        }
        }
        }
```

Output:

```
Enter the values of a,b,c,d: 15 20 25 30
30 is the largest
```

3.3 Loop control statements

In computer programming, sometimes programmer have to perform same task again and again on the same data with a few changes. In this situation programmer can either write same code again and again which consumes lots of time and space as well over can use loop to iterate same code to save time and space. Looping Statements in C execute the sequence of statements many times until the stated condition becomes false. A loop in C consists of two parts, a body of a loop and a control statement. The control statement is a combination of some conditions that direct the body of the loop to execute until the specified condition becomes false. The purpose of the C loop is to repeat the same code a number of times. C Language provides the following advantages of with the use of Looping?

- It provides code reusability.
- Using loops, we do not need to write the same code again and again.
- Using loops, we can traverse over the elements of data structures structures (array or linked lists).

Depending upon the position of a control statement in a program, looping statement in C is classified into two categories:

1. Entry controlled loop: In an entry control loop in C, a condition is checked before executing the body of a loop. It is also called as a pre-checking loop.
2. Exit controlled loop: In an exit controlled loop, a condition is checked after executing the body of a loop. It is also called as a post-checking loop.

The control conditions must be well defined and specified otherwise the loop will execute an infinite number of times. The loop that does not stop executing and processes the statements number of times is called as an infinite loop. An infinite loop is also called as an "Endless loop." Following are some characteristics of an infinite loop:

1. No termination condition is specified.
2. The specified conditions never meet.

On the basis of these two categories, in C programming there are four types of loops are discussed below:

- While Loop
- Do-while loop
- For Loop
- Nested Loop

3.3.1 While Loop: While loop is entry controlled loop. In while loop, a condition is evaluated before processing a body of the loop. If a condition is true then and only then the body of a loop is executed.

The syntax of the while loop is:

```
while (testExpression)
{
  // the body of the loop
}
```

The execution flow of while loop is explained below:

- The while loop evaluates the testExpression inside the parentheses ().
- If testExpression is true, statements inside the body of while loop are executed. Then, testExpression is evaluated again.
- The process goes on until testExpression is evaluated to false.
- If testExpression is false, the loop terminates (ends).

While loop is also known as a pre-tested loop. In general, a while loop allows a part of the code to be executed multiple times depending upon a given boolean condition. It can be viewed as a repeating if statement. The while loop is mostly used in the case where the number of iterations is not known in advance. The following program explain the functioning of while loop in c language.

```
#include<stdio.h>
#include<conio.h>
int main()
        {
        int num=1;      //initializing the variable
        while(num<=10)        //while loop with condition
        {
                printf(" %d ",num);
                num++;                  //incrementing operation
        }
        return 0;
        }
```

Output: 1 2 3 4 5 6 7 8 9 10

The above program illustrates the use of while loop to print series of numbers from 1 to 10.

- We have initialized a variable called num with value 1. We are going to print from 1 to 10 hence the variable is initialized with value 1. If you want to print from 0, then assign the value 0 during initialization.
- In a while loop, we have provided a condition (num<=10), which means the loop will execute the body until the value of num becomes 10. After that, the loop will be terminated, and control will fall outside the loop.
- In the body of a loop, we have a print function to print our number and an increment operation to increment the value per execution of a loop. An initial value of num is 1, after the execution, it will become 2, and during the next execution, it will become 3. This process will continue until the value becomes 10 and then it will print the series on console and terminate the loop.

3.3.2 Do-while Loop Statements: Do-while is exit controlled loop. Using the do-while loop, we can repeat the execution of several parts of the statements. The do-while loop is similar to the while loop with one important difference. The body of do-while loop is executed at least once. Only then, the test expression is evaluated. The do-while loop is mostly used in menu-driven programs where the termination condition depends upon the end user. Therefore, it is also called post tested loop.

The execution flow of do-while loop is explained below:

- The body of do...while loop is executed once. Only then, the testExpression is evaluated.
- If testExpression is true, the body of the loop is executed again and testExpression is evaluated once more.
- This process goes on until testExpression becomes false.
- If testExpression is false, the loop ends.

The syntax of the C language do-while loop is given below:

```
do
{
//code to be executed
}while(testExpression);
```

As described above the do-while runs at least once even if the condition is false because the condition is evaluated, after the execution of the body of loop. The following program explain the functioning of do-while loop in c language.

```
// C Program to demonstrate the do-while loop behaviour
// when the condition is false from the start
#include <stdbool.h>
#include <stdio.h>
int main()
   {
  // declaring a false variable
  bool condition = false;
      do
      {
            printf("This is loop body.");
      } while (condition); // false condition

  return 0;
       }
```

Output:

This is loop body.

In the above programme, even when the condition is false at the start, the loop body is executed once. This is because in the do-while loop, the condition is checked after going through the body so when the control is at the start.

- It goes through the loop body.
- Executes all the statements in the body.
- Checks the condition which turns out to be false.
- Then exits the loop.

3.3.3 For Loop Statement: It is also called entry controlled loop. It is also provides a functionality/feature to recall a set of conditions for a defined number of times, moreover, this methodology of calling checked conditions automatically is known as for loop. It is mainly used to traverse arrays, vectors, and other data structures.

```
for(initialization; check/test expression; updation)
{
        // body consisting of multiple statements
}
```

Characteristics of for loop in C:

For loop follows a very structured approach where it begins with initializing a condition then checks the condition and in the end executes conditional statements following with updation of values.

- **Initialization**: This is the first parameter of a fundamental for loop that accepts a conditional variable that iterates the value or helps in checking the condition.
- **Conditional Statement**: It accepts 3 parameters (Initialization, Condition, and Updation) that indicate what condition needs to be followed and checked.
- **Check/Test Condition**: The Second parameter of a fundamental for loop defines the condition that needs to be followed to run the following code statements. In simple terms, if the check expression is true then the iteration of the loop continues otherwise the loop is terminated and further checks (if possible/there) are left unchecked.
- **Updation**: The Third parameter of a fundamental for loop defines the increment or decrement of the conditional variable that will iterate the code according to the condition.

To learn more about when the test expression is evaluated to true and false, the following program demonstrate the flow of for loop.

```
// Print numbers from 1 to 10
        #include <stdio.h>
        int main()
        {
        int i;
        for (i = 1; i < 11; ++i)
        {
        printf("%d ", i);
        }
        return 0;
        }
```

```
        Output: 1 2 3 4 5 6 7 8 9 10
```

Explanation:

1. i is initialized to 1.
2. The test expression i < 11 is evaluated. Since 1 less than 11 is true, the body of for loop is executed. This will print the 1 (value of i) on the screen.
3. The update statement ++i is executed. Now, the value of i will be 2. Again, the test expression is evaluated to true, and the body of for loop is executed. This will print 2 (value of i) on the screen.
4. Again, the update statement ++i is executed and the test expression i < 11 is evaluated. This process goes on until i becomes 11.
5. When i becomes 11, i < 11 will be false, and the for loop terminates.

3.3.4 Nested-Loop Statements: C supports nesting of loops in C. Nesting of loops is the feature in C that allows the looping of statements inside another loop. Let's observe an example of nesting loops in C. Any number of loops can be defined inside another loop, i.e., there is no restriction for defining any number of loops. The nesting level can be defined at n times. User can define any type of loop inside another loop; for example, you can define 'while' loop inside a 'for' loop.

Syntax of Nested loop:

```
Outer_loop
{
  Inner_loop
  {
   // inner loop statements.
   }
    // outer loop statements.
   }
```

Outer_loop and Inner_loop are the valid loops that can be any type of aforesaid loops.

Working of Nested Loop:

- Execution of statement within the loop flows in a way that the inner loop of the nested loop gets declared, initialized and then incremented.
- Once all the condition within the inner loop gets satisfied and becomes true it moves for the search of the outer loop. It is often called a loop within a loop.

Suppose we want to loop through each day of a week for 3 weeks. To achieve this, we can create a loop to iterate three times (3 weeks). And inside the loop, we can create another loop to iterate 7 times (7 days). This is how we can use nested loops. The following program exemplify the used of nested loop to print pattern using for loop statement.

```
// C program to display a triangular pattern
// Number is entered by the user
#include <stdio.h>
int main()
{
  int i, j, n;
  printf("Enter Number :  ");
  scanf ("%d", &n);
  for (i = 1; i <= n; i++) {
    for (j = 1; j <= i; j++) {
      printf("*  ");
    }
```

```
        printf("\n");
        }
      return 0;
         }
```

Output:

```
Enter Number: 4
*
*    *
*    *    *
*    *    *    *
```

3.4 Case control statements:

The statements which are used to execute only specific block of statements in a series of blocks are called case control statements. There are 4 types of case control statements in C language named as:

- Switch Statement
- goto Statement
- break Statement
- continue Statement

3.4.1 Switch statement: Sometimes, there may be requirement to evaluate the multiple expression against different cases. Then switch statement in C tests the value of a variable and compares it with multiple cases. Once the case match is found, a block of statements associated with that particular case is executed. Each case in a block of a switch has a different name/number which is referred to as an identifier. The value provided by the user is compared with all the cases inside the switch block until the match is found. The following syntax explain the method to use switch statement.

Syntax:

```
switch(expression)
{
  case constant-expression  :
    statement(s);
    break; /* optional */
case constant-expression  :
   statement(s);
   break; /* optional */
/* you can have any number of case statements */
default : /* Optional */
statement(s);
}
```

Some major rules must be follow while using a switch statement:

- The expression used in a switch statement must have an integral or enumerated type, or be of a class type in which the class has a single conversion function to an integral or enumerated type.
- You can have any number of case statements within a switch. Each case is followed by the value to be compared to and a colon.
- The constant-expression for a case must be the same data type as the variable in the switch, and it must be a constant or a literal.
- When the variable being switched on is equal to a case, the statements following that case will execute until a break statement is reached.
- When a break statement is reached, the switch terminates, and the flow of control jumps to the next line following the switch statement.
- Not every case needs to contain a break. If no break appears, the flow of control will fall through to subsequent cases until a break is reached.
- A switch statement can have an optional default case, which must appear at the end of the switch. The default case can be used for performing a task when none of the cases is true. No break is needed in the default case.

Let's see a simple example of c language using switch statement.

```
#include<stdio.h>
int main(){
int number=0;
printf("enter a number:");
scanf("%d",&number);
switch(number){
case 10:
        printf("number is equals to 10");
        break;
case 50:
        printf("number is equal to 50");
        break;
case 100:
        printf("number is equal to 100");
        break;
default:
        printf("number is not equal to 10, 50 or 100");
}
return 0;
}
```

Output:

```
        Case 1:
                enter a number:4
                number is not equal to 10, 50 or 100
        Case 2:
                enter a number:50
                number is equal to 50
```

First, the integer expression specified in the switch statement is evaluated. This value is then matched one by one with the constant values given in the different cases. If a match is found, then all the statements specified in that case are executed along with the all the cases present after that case including the default statement. No two cases can have similar values. If the matched case contains a break statement, then all the cases present after that will be skipped, and the control comes out of the switch. Otherwise, all the cases following the matched case will be executed.

3.4.2 goto Statement: The goto statement is known as jump statement in C. As the name suggests, goto is used to transfer the program control to a predefined label. The goto statment can be used to repeat some part of the code for a particular condition. It can also be used to break the multiple loops which can't be done by using a single break statement. However, using goto is avoided these days since it makes the program less readable and complex in large cases.

Syntax of goto Statement:

```
goto label;

... .. ...

... .. ...

label:

statement;
```

The label is an identifier. When the goto statement is encountered, the control of the program jumps to label: and starts executing the code. The following program exemplify the use of goto to print table.

```
//Print a number table
#include <stdio.h>
int main()
{
  int num,i=1;
  printf("Enter the number whose table you want to print?");
  scanf("%d",&num);
  table:
  printf("%d x %d = %d\n",num,i,num*i);
  i++;
  if(i<=10)
  goto table;
}
```

Output:

```
Enter the number whose table you want to print?      10
10 x 1 = 10
10 x 2 = 20
10 x 3 = 30
10 x 4 = 40
10 x 5 = 50
```

```
10 x 6 = 60
10 x 7 = 70
10 x 8 = 80
10 x 9 = 90
                    10   10 = 100
```

3.4.3 Break Statement: The break is a keyword in C which is used to bring the program control out of the loop. The break statement is used inside loops or switch statement. The break statement breaks the loop one by one, i.e., in the case of nested loops, it breaks the inner loop first and then proceeds to outer loops.

Syntax

The syntax for a break statement in C is as follows –

```
        {
                break;
        }
```

The break statement in C programming has the following two usages:

- When a break statement is encountered inside a loop, the loop is immediately terminated and the program control resumes at the next statement following the loop.
- It can be used to terminate a case in the switch statement.

If you are using nested loops, the break statement will stop the execution of the innermost loop and start executing the next line of code after the block. The following two programs demonstrate the use of break statement in both the cases.

```
#include<stdio.h>
int main()
{
int i=1,j=1;//initializing a local variable
for(i=1;i<=3;i++)
{
        for(j=1;j<=3;j++)
        {
                printf("%d &d\n",i,j);
                if(i==2 && j==2)
                {
                        break;//will break loop of j only
                }
                }//end of for loop
return 0;
}
```

Output:

```
1 1
1 2
1 3
2 1
2 2
3 1
3 2
3 3
```

The following program exemplify the use of break in switch statement.

```
#include<stdio.h>
int main(){
int number=0;
printf("enter a number:");
scanf("%d",&number);
switch(number){
case 10:
        printf("number is equals to 10");
        break;
case 50:
        printf("number is equal to 50");
        break;
case 100:
        printf("number is equal to 100");
        break;
default:
        printf("number is not equal to 10, 50 or 100");
}
return 0;
}
```

Output:

Case 1:

```
enter a number:4
number is not equal to 10, 50 or 100
```

Case 2:

```
enter a number:50
number is equal to 50
```

3.4.4 Continue Statement: The continue statement in C language is used to bring the program control to the beginning of the loop. The continue statement skips some lines of code inside the loop and continues with the next iteration. It is mainly used for a condition so that we can skip some code for a particular condition. The continue statement in C programming works somewhat like the break statement. Instead of forcing termination, it forces the next iteration of the loop to take place, skipping any code in between. For the for loop, continue statement causes the conditional test and increment portions of the loop to execute. For the while and do...while loops, continue statement causes the program control to pass to the conditional tests. The syntax and example of continue statement is given below:

Syntax:

```
//loop statements
continue;
//some lines of the code which is to be skipped
```

Continue statement example 1

```
#include<stdio.h>
void main ()
{
   int i = 0;
   while(i!=10)
   {
      printf("%d", i);
      continue;
```

```
     i++;
      }
    }
```

Output:

```
    infinite loop
```

Continue statement example 2

```
#include<stdio.h>
int main()
    {
    int i=1;//initializing a local variable
    //starting a loop from 1 to 10
for(i=1;i<=10;i++)
    {
    if(i==5)
    {//if value of i is equal to 5, it will continue the loop
        continue;
    }
    printf("%d \n",i);
    }//end of for loop
return 0;
    }
```

Output:

```
    1
    2
    3
    4
    6
    7
    8
    9
    10
```

As you can see, 5 is not printed on the console because loop is continued at i==5.

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- A

UNIT 4: ARRAYS AND POINTERS IN C

4.1 Arrays

An array is defined as the collection of similar type of data items stored at contiguous memory locations. Arrays are the derived data type in C programming language which can store the primitive type of data such as int, char, double, float, etc. It also has the capability to store the collection of derived data types, such as pointers, structure, etc. The array is the simplest data structure where each data element can be randomly accessed by using its index number.

For example, to store the marks of a student in 4 different subjects, there is no need to define different variables for the marks in the different subject. Instead of that, array can be used which can store the marks in each subject at the contiguous memory locations.

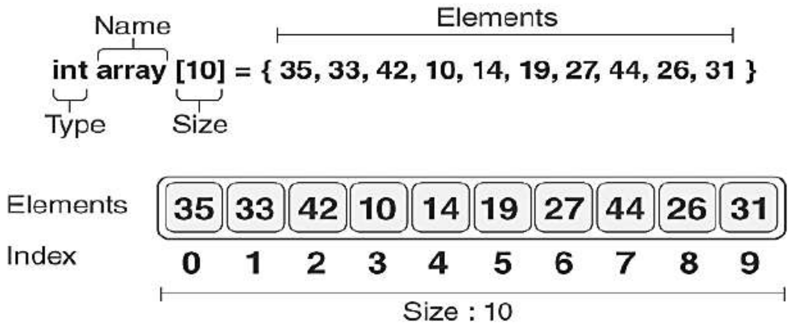
4.1.1 Characteristic of Arrays:

1. Array is a group of variables of similar data types referred to by a single element.
2. Array elements are stored in a contiguous memory location.
3. The size of the array should be mentioned while declaring it.
4. Array elements are always counted from zero (0) onward.
5. Array elements can be accessed using the position of the element in the array.
6. The array can have one or more dimensions.

4.1.2 Representation of an Array:

Arrays always store the same type of values. In the below mentioned example:

- **int** is a data type of Array having size 10.
- Data items stored in an array are known as elements.
- The location or placing of each element has an index value.



4.1.3 Declaration and Initialization of an Array:

To declare an array in C, a programmer specifies the type of the elements and the number of elements required by an array as follows:
Syntax: data_type array_name[array_size];

Example: int marks[4];
In this example, int is the *data_type*, marks are the *array_name*, and 4 is the *array_size*. Hence, marks is the one-dimensional array.

NOTE: The name of the array is also a pointer to the first element of the array.

Initialization of Array:

A) The simplest way to initialize an array is by using the index of each element as mentioned in the following example.

```
//Program to demonstrate the Initialization of Array
#include<stdio.h>
int main()
{
int i=0;
int marks[4];//declaration of array
marks[0]=80;//initialization of array
marks[1]=60;
marks[2]=70;
marks[3]=85;
//traversal of array
for(i=0;i<4;i++){
printf("%d \n",marks[i]);
}//end of for loop
return 0;
}
```

Output:

```
        80
        60
        70
        8585
```

B) Initialize elements of an array at the time of declaration

Syntax: datatype Array_Name[size] = { value1, value2, value3, ….. valueN };

Example: int marks[4]={30,40,50,60};

In such case, there is **no requirement to define the size**. So it may also be written as the following code.

```
        int marks[]={30,40,50,60};
```

```
 //Program to demonstrate the Initialization of Array
#include<stdio.h>
int main(){
int i=0;
int marks[4]={30,40,50,60};
//declaration and initialization of array
 //traversal of array
for(i=0;i<5;i++){
printf("%d \n",marks[i]);
}
return 0;
}
```

Output:

```
30
40
50
60
```

Advantages of Array:

- It is a better version of storing the data of the same size and same type.
- It enables us to collect the number of elements in it.
- Arrays have a safer cache positioning that improves performance.
- Arrays can represent multiple data items of the same type using a single name.

Disadvantages of Array:

- In an array, it is essential to identify the number of elements to be stored.
- It is a static structure. It means that in an array, the memory size is fixed.
- When it comes to insertion and deletion, it is a bit difficult because the elements are stored sequentially and the shifting operation is expensive.

NOTE: Array elements are accessed by using an integer index. Array index starts with 0 and goes till the size of the array minus 1.

4.2 Types of Arrays

There are two types of arrays:

- One-Dimensional Arrays
- Multi-Dimensional Arrays

4.2.1 One-Dimensional Arrays: A one-dimensional array is a kind of linear array. It involves single sub-scripting. The [] (brackets) is used for the subscript of the array and to declare and access the elements from the array.

Syntax: DataType ArrayName [size];

Example: int marks [6];

4.2.2 Multi-Dimensional Arrays: A multi-dimensional array can be termed as an array of arrays that stores homogeneous data in tabular form. Data in multidimensional arrays are stored in row-major order. In multi-dimensional arrays, there are two categories:

- Two-Dimensional Arrays
- Three-Dimensional Arrays

Two-Dimensional Arrays

An array involving two subscripts [] [] is known as a two dimensional array. They are also known as the array of the array. Two-dimensional arrays are divided into rows and columns and are able to handle the data of the table.

Syntax: DataType ArrayName[row_size][column_size];

Example: int a [6][6];

- **Initializing Two-Dimensional Arrays**

1. **First Method**:
   int x[3][4] = {0, 1 ,2 ,3 ,4 , 5 , 6 , 7 , 8 , 9 , 10 , 11}
   The above array has 3 rows and 4 columns. The elements in the braces from left to right are stored in the table also from left to right. The elements will be filled in the array in order, the first 4 elements from the left in the first row, the next 4 elements in the second row, and so on.

2. **Second Method**:
   int x[3][4] = {{0,1,2,3}, {4,5,6,7}, {8,9,10,11}};

```
//Program to demonstrate two-dimensional Array
#include<stdio.h>
int main(){
int i=0,j=0;
int arr[4][3]={{1,2,3},{2,3,4},{3,4,5},{4,5,6}};
//traversing 2D array
for(i=0;i<4;i++){
 for(j=0;j<3;j++){
   printf("arr[%d] [%d] = %d \n",i,j,arr[i][j]);
 }//end of j
}//end of i
return 0;
}
```

Output:

```
arr[0][0] = 1
arr[0][1] = 2
arr[0][2] = 3
arr[1][0] = 2
arr[1][1] = 3
arr[1][2] = 4
arr[2][0] = 3
arr[2][1] = 4
arr[2][2] = 5
arr[3][0] = 4
arr[3][1] = 5
arr[3][2] = 6
```

3. **Third Method:**

```
int x[3][4];

for(int i = 0; i < 3; i++)
{
    for(int j = 0; j < 4; j++)
{
        Scanf("%d", x[i][j]);

    }

   }
```

- **Accessing Two-Dimensional Arrays:** Elements of Two-Dimensional arrays can be accessed using the row indexes and column indexes.

  Example: int x[2][1];

The above example represents the element present in the third row and second column.

Three-Dimensional Arrays

When there is need to create two or more tables of the elements to declare the array elements, then in such a situation three-dimensional arrays is used.

Syntax: DataType ArrayName[size1][size2][size3];

For Example: int a [6][5][5];

- **Initializing Three-Dimensional Array**: Initialization in a Three-Dimensional array is the same as that of Two-dimensional arrays. The difference is as the number of dimensions increases so the number of nested braces will also increase.

Method 1:

```
int x[2][3][4] =    {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10,
                     11, 12, 13, 14, 15, 16, 17, 18, 19,
                     20, 21, 22, 23};
```

Method 2:

```
int x[2][3][4] =    {
                     { {0,1,2,3}, {4,5,6,7}, {8,9,10,11} },
                     { {12,13,14,15}, {16,17,18,19}, {20,21,22,23} }
                     };
```

- **Accessing Three-Dimensional Arrays**: Accessing elements in Three-Dimensional Arrays is also similar to that of Two-Dimensional Arrays. The difference is we have to use three loops instead of two loops for one additional dimension in Three-dimensional Arrays.

```
// C program to print elements of Three-Dimensional Array

#include <stdio.h>

int main(void)
```

```
{
    // initializing the 3-dimensional array
    int x[2][3][2] = { { { 0, 1 }, { 2, 3 }, { 4, 5 } },
                       { { 6, 7 }, { 8, 9 }, { 10, 11 } } };

    // output each element's value
    for (int i = 0; i < 2; ++i) {
        for (int j = 0; j < 3; ++j) {
            for (int k = 0; k < 2; ++k) {
                printf("Element at x[%i][%i][%i] = %d\n", i, j, k, x[i][j][k]);
            }
            }
            }
    return (0);
           }
```

Output:

```
Element at x[0][0][0] = 0
Element at x[0][0][1] = 1
Element at x[0][1][0] = 2
Element at x[0][1][1] = 3
Element at x[0][2][0] = 4
Element at x[0][2][1] = 5
Element at x[1][0][0] = 6
Element at x[1][0][1] = 7
Element at x[1][1][0] = 8
Element at x[1][1][1] = 9
Element at x[1][2][0] = 10
Element at x[1][2][1] = 11
```

Difference between One-Dimensional and Two-Dimensional Array

Parameters	One-Dimensional Array	Two-Dimensional Array
Concept	A one-dimensional array stores a single list of various elements having a similar data type.	A two-dimensional array stores an array of various arrays, or a list of various lists, or an array of various one-dimensional arrays.
Representation	It represents multiple data items in the form of a list.	It represents multiple data items in the form of a table that contains columns and rows.
Dimensions	It has only one dimension.	It has a total of two dimensions.
Total Size (in Bytes)	Total number of Bytes = The size of array * the size of array variable or datatype.	Total number of Bytes = The size of array visible or datatype * size of second index * size of the first index.

4.3 Pointer

The pointer in C language is a variable which stores the address of another variable. This variable can be of type int, char, array, function, or any other pointer. The pointer sizes depend on their architecture.

NOTE: The pointer of type void is called **Void pointer** or **Generic pointer**. Void pointer can hold address of any type of variable.

The following operators are used for the pointers in C language:

Operator	Name	Uses and Meaning of Operator
*	Asterisk	Declares a pointer in a program. Returns the referenced variable’s value.
&	Ampersand	Returns a variable’s address

4.3.1 Pointer Declaration and Initialization

Just like the variables, there is need to declare the pointers in C before we use them in any program.

Syntax:

```
datatype *Pointer_var_name;
```

- The data_type refers to this pointer's base type in the variable of C. It indicates which type of variable is the pointer pointing to in the code.
- The asterisk (*) is used to declare a pointer which is also called the indirection pointer.

Example:

```
 int  *q; // a pointer q for int data type

char  *x; // a pointer x for char data type
```

Pointer Initialization

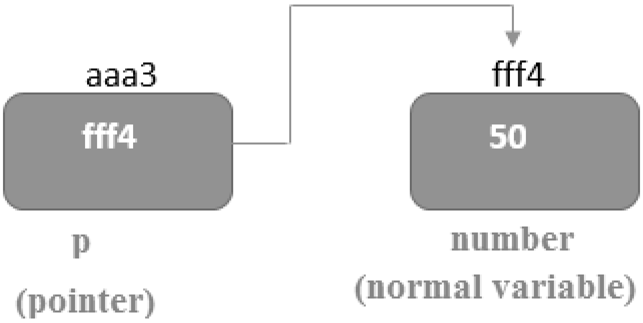
After the pointer declaration, the next step is to initialise the pointers just like the standard variables using an address of the variable. The ampersand (&) operator is used to get the variable's address in the program. If we don't initialise the pointers in any C program and start using it directly, the results will be ambiguous.

Syntax:

```
pointer = &variable;
```

Example:

```
int *p;
p=&number;
```



```
// C program to illustrate Pointers
#include<stdio.h>
 int main(){
 int number=50;
 int *p;
 p=&number; //stores the address of number variable
 printf("Address of number variable is %x \n",p);
 // p prints the address of the number
 printf("Value of p variable is %d \n",*p);
 // prints the value stored in pointer p.
 return 0;
 }
```

OUTPUT:

```
Address of number variable is fff4
Value of p variable is 50
```

4.3.2 Types of Pointers

There are various types of pointers that can be used in the C language.

- **The Null Pointer:** To create a null pointer in C language, assign the pointer with a null value during its declaration in the program. This type of method is especially useful when the pointer has no address assigned to it. The null pointer has a value of 0.

```
//Use of NULL Pointer

#include <stdio.h>
int main()
{
int *a = NULL; // the null pointer declaration
printf("Value of the variable a in the program is equal to :\n%x",a);
return 0;
}
```

Output:

```
Value of the variable a in the program is equal to: 0
```

- **The Void Pointer:** This pointer has no standard data type, and declares it with the use of the keyword ***void***. The void pointer is generally used for the storage of any variable’s address. The void pointer is also known as the generic pointer in the C language.

//Use of void Pointer

```
#include <stdio.h>
int main()
{
void *q = NULL; // the void pointer of the program
printf(“Size of the void pointer in the program is equal to : %d\n”,sizeof(q));
return 0;
}
```

Output:

```
Size of the void pointer in the program is equal to: 4
```

Advantages of using Pointers

- Pointers make it easy for us to access locations of memory.
- Pointers are used for dynamic allocation and deallocation of memory.
- Pointers are also used to create complex data structures, like the linked list, tree, graph, etc.
- Pointers provide an efficient way in which we can access the elements present in the structure of an array.
- Pointers reduce the code and thus improve the overall performance. We can use the pointers for returning various values from any given function.

Disadvantages of using Pointers

- The concept of pointers is a bit tricky to understand.
- Pointers in a program can lead to leakage of memory.
- Memory corruption can occur if an incorrect value is provided to pointers.
- Pointers are comparatively slower than variables in C.
- These can lead to some errors like segmentation faults, accessing of unrequired memory locations, and many more.

Applications of Pointers

There are various uses of the pointers in C language.

- Dynamic allocation of memory: We can allocate the memory dynamically in C language by using the calloc() and malloc() functions along with the pointers.
- Structures, Functions, and Arrays use the pointers in the C. It help in reducing the code and improving the program’s performance.

4.3.3 Pointer Expressions and Pointer Arithmetic

A limited set of arithmetic operations can be performed on pointers. The Pointer Arithmetic is slightly different from the ones that we generally use for mathematical calculations. The operations are:

- Increment/Decrement of a Pointer
- Addition of integer to a pointer
- Subtraction of integer to a pointer
- Subtracting two pointers of the same type
- Comparison of pointers of the same type.
- Assignment of pointers to the same type of pointers.

NOTE: Pointer arithmetic is meaningless unless performed on an array.

// C program to illustrate Pointer Arithmetic

```
#include <stdio.h>

int main()
{

   // Declare an array
   int v[3] = { 10, 100, 200 };

   // Declare pointer variable
   int* ptr;

   // Assign the address of v[0] to ptr
   ptr = v;

   for (int i = 0; i < 3; i++) {

        // print value at address which is stored in ptr
      printf("Value of *ptr = %d\n", *ptr);

        // print value of ptr
      printf("Value of ptr = %p\n\n", ptr);

      // Increment pointer ptr by 1
      ptr++;
      }
   return 0;
        }
```

OUTPUT:

```
Value of *ptr = 10
Value of ptr = 0x7ffe8ba7ec50

Value of *ptr = 100
Value of ptr = 0x7ffe8ba7ec54

Value of *ptr = 200
Value of ptr = 0x7ffe8ba7ec58
```

Pointers and Arrays

- **Array of Pointers:** Pointers are variables which stores the address of another variable. Pointer to an array points the address of memory block of an array variable.

Syntax:

```
datatype *variable_name[size];
```

- **datatype** defines the datatype of variable like int, char, float etc.

- **variable_name** is the name of variable given by user.
- **size** defines the size of array variable.

Example: int (*ptr) [10];

//Program to demonstrate Arrays of Pointers

```
#include <stdio.h>
int main ()
{
  int *arr[3];
  int *a;
  printf( "Value of array pointer variable : %d", arr);
  printf( "Value of pointer variable : %d", &a);
  return 0;
}
```

Output:

```
Value of array pointer variable: 1481173888

Value of pointer variable: 1481173880
```

- **Pointer to an Array:** In a pointer to an array, store the base address of the array in the pointer variable.

  Example: *ptr=arr;
  *ptr=&arr;
  *ptr=&arr[0];

  Here, ptr will store the base address of the array.

//Program to demonstrate Pointer to Arrays

```
#include <stdio.h>
int main()
{
// array declaration and initialization
        int arr[5] = {3, 5, 7, 9, 11}, i;

// both `arr` and `&arr` return the address of the first element of the array.
```

```
        int *ptr = arr;

// printing the elements of array using addition arithmetic on pointer
        for(i = 0; i < 5; i++)
        {
                printf("%d ", *(ptr + i));
        }
        return 0;
        }
```

Output:

```
3 5 7 9 11
```

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- B

UNIT 5: FUNCTIONS AND POINTERS IN C

5.1 Functions in C

Function in C programming is a reusable block of code that makes a program easier to understand, test and can be easily modified without changing the calling program. Functions divide the code and modularize the program for better and effective results. In short, a larger program is divided into various subprograms which are called as functions

There are three components of a C function.

5.1.1 **Function Declaration:** A function must be declared globally in a C program to tell the compiler about the function name, function parameters, and return type. Function declaration means writing a name of a program. It is a compulsory part for using functions in code. In a function declaration, we just specify the name of a function that we are going to use in our program like a variable declaration. The function declarations (called prototype) are usually done above the main () function and take the general form:

Syntax: return_data_type function_name (data_type arguments);

- The **return_data_type**: is the data type of the value function returned back to the calling statement.
- The **function_name**: is followed by parentheses
- **Arguments:** names with their data type declarations optionally are placed inside the parentheses.

Example: consider the following program that shows how to declare a cube function to calculate the cube value of an integer variable

```
#include <stdio.h>
/*Function declaration*/
int add(int a,b);
/*End of Function declaration*/
```

5.1.2 Function Definition: Function definition means just writing the body of a function. A body of a function consists of statements which are going to perform a specific task. A function body consists of a single or a block of statements. It is also a mandatory part of a function. It contains the actual statements which are to be executed. It is the most important aspect to which the control comes when the function is called. Here, we must notice that only one value can be returned from the function.

```
int add(int a,int b)  //function body
          {
          int c;
          c=a+b;
          return c;
          }
```

1. **Function Call:** A function call means calling a function whenever it is required in a program. Whenever we call a function, it performs an operation for which it was designed. A function call is an optional part of a program. Function can be called from anywhere in the program. The parameter list must not differ in function calling and function

declaration. We must pass the same number of functions as it is declared in the function declaration.

```
                              result = add(4,5);
#include <stdio.h>
int add(int a, int b);              //function declaration
int main()
{
        int a=10,b=20;
        int c=add(10,20);   //function call
        printf("Addition:%d\n",c);
        getch();
}
int add(int a,int b)  //function body
{
        int c;
        c=a+b;
        return c;
}
```

Output:

```
Addition:30
```

5.1.3 Library Vs. User-defined Functions

Every C program has at least one function which is the main function, but a program can have any number of functions. The main () function in C is a starting point of a program. In ‘C’ programming, functions are divided into two types:

1. Library functions
2. User-defined functions

The difference between the library and user-defined functions in C is that we do not need to write a code for a library function. It is already present inside the header file which we always include at the beginning of a program.

- **Library Functions:** Library functions are the inbuilt function in C that are grouped and placed at a common place called the library. Such functions are used to perform some specific operations. For example, printf is a library function used to print on the console. The library functions are created by the designers of compilers. All C standard library functions are defined inside the different header files saved with the extension **.h**. We need to include these header files in our program to make use of the library functions defined in such header files. For example, Tt use the library functions such as printf/scanf we need to include stdio.h in our program which is a header file that contains all the library functions regarding standard input/output. The list of mostly used header files is given in the following table.

S.No	**Header file**	**Description**
1	stdio.h	This is a standard input/output header file. It contains all the library functions regarding standard input/output.
2	conio.h	This is a console input/output header file.
3	string.h	It contains all string related library functions like gets(), puts(),etc.
4	stdlib.h	This header file contains all the general library functions like malloc(), calloc(), exit(), etc.
5	math.h	This header file contains all the math operations related functions like sqrt(), pow(), etc.
6	time.h	This header file contains all the time-related functions.
7	ctype.h	This header file contains all character handling functions.

5.1.4 Function calling Methods

A function may or may not accept any argument. It may or may not return any value. Based on these facts, there are four different aspects of function calls. The details of each with suitable example is given below:

Function without arguments and without return value

```
#include<stdio.h>
void printName();
void main ()
{
   printf("Hello ");
   printName();
}
void printName()
{
    printf("Javatpoint");
}
```

OUTPUT:

```
                        Hello Javatpoint
```

Function with arguments and without return value

```
#include<stdio.h>
void sum(int, int);
void main()
{
    int a,b,result;
    printf("\nGoing to calculate the sum of two numbers:");
    printf("\nEnter two numbers:");
    scanf("%d %d",&a,&b);
    sum(a,b);
}
void sum(int a, int b)
{
    printf("\nThe sum is %d",a+b);
}
```

OUTPUT:

```
Going to calculate the sum of two numbers:
Enter two numbers 10 24
The sum is 34
```

Function without arguments and with return value

```
#include<stdio.h>
int sum();
void main()
{
    int result;
    printf("\nGoing to calculate the sum of two numbers:");
    result = sum();
    printf("%d",result);
}
```

```
int sum()
{
   int a,b;
   printf("\nEnter two numbers");
   scanf("%d %d",&a,&b);
   return a+b;
}
```

OUTPUT:

```
                  Going to calculate the sum of two numbers:
                  Enter two numbers 10 24
                  The sum is 34
```

Function with arguments and with return value

```
#include<stdio.h>
int sum(int, int);
void main()
{
   int a,b,result;
   printf("\nGoing to calculate the sum of two numbers:");
   printf("\nEnter two numbers:");
   scanf("%d %d",&a,&b);
   result = sum(a,b);
   printf("\nThe sum is : %d",result);
}
int sum(int a, int b)
{
   return a+b;
}
```

OUTPUT:

```
Going to calculate the sum of two numbers:
Enter two numbers:        10  20
The sum is: 30
```

5.1.5 Function Parameters

A parameter is a way to convey information to a function. Sometimes we also refer to parameters as arguments. Parameters are nothing but the input provided to the function. Function work on these parameters to produce a result or to perform a certain action. The data necessary for the function to perform the task is sent as parameters. Parameters can be actual parameters or Formal Parameters. The key difference between Actual Parameters and Formal Parameters is that Actual Parameters are the values that are passed to the function when it is invoked while Formal Parameters are the variables defined by the function that receives values when the function is called. There is basic two terminology used when using arguments in functions**:**

5.1.5.1 Actual Parameter: Actual parameters are the values, variables or expressions that we pass to a calling function. While defining the actual parameters in the calling function we do not associate data type with the parameters. In a calling function, the actual parameters can be provided either by the programmer at the time of programming or by the user in real-time when the program is being executed.

5.1.5.2 Formal Parameter: Formal parameters are the variable that we pass to function when it is declared or defined in the program. The formal parameters defined during function definition accept the real values in the order they are passed by the calling function. The statements in the function definition access these real values by referring to them with the variable name defined by the formal parameter.

The following program demonstrates the concept of actual and formal arguments:

In this example, the value of variable x is 10 before the function func_1() is called, after func_1() is called, the value of x inside main() is still 10. The changes made inside the function func_1() doesn't affect the value of x. This happens because when we pass values to the functions, a copy of the value is made and that copy is passed to the formal arguments. Hence Formal arguments work on a copy of the original value, not the original value itself, that's why changes made inside func_1() is not reflected inside main().

```
#include<stdio.h>
void func_1(int);
int main()
{
int x = 10;
printf("Before function call\n");
printf("x = %d\n", x);
func_1(x);
printf("After function call\n");
printf("x = %d\n", x);
// signal to operating system program ran fine
return 0;
}
void func_1(int a)
{
 a += 1;
 a++;
 printf("\na = %d\n\n", a);
}
```

5.1.6 Parameter Passing Techniques

In C, a function specifies the modes of parameter passing to it. There are two ways to specify function calls: call by value and call by reference in C. In call by value, the function parameters gets the copy of actual parameters which means changes made in function parameters did not reflect in actual parameters. In call by reference, the function parameter gets reference of actual parameter which means they point to similar storage space and changes made in function parameters will reflect in actual parameters. During the calling of a function, there will be two ways in which we can perform the passing of these arguments to a given function:

Type of Call	**Description**
Call By Reference	The Call by Reference method creates a copy of the address of the given argument into the parameter that is formal in nature. Inside this function, the use of address helps in accessing the actual argument that comes in use in this call. It means that the changes that appear on the parameter are bound to affect the given argument.
Call By Value	The Call by Value method creates a copy of the actual value of the given argument in the parameter of the function that is formal in nature. Here, the changes that appear on the parameter (that exists inside the function) create no effect whatsoever on the available argument.

The C programming, by default, is the Call by Value for passing the arguments. Generally, it means that we cannot make use of the code that exists within a function for altering the arguments that help in calling the function.

```
// C program to show use of call by value
#include <stdio.h>
 void swap(int var1, int var2)
{
  int temp = var1;
  var1 = var2;
  var2 = temp;
}
 int main()
{
  int var1 = 3, var2 = 2;
  printf("Before swap Value of var1 and var2 is: %d, %d\n",
          var1, var2);
  swap(var1, var2);
  printf("After swap Value of var1 and var2 is: %d, %d",
          var1, var2);
  return 0;
}
```

OUTPUT:

```
               Before swap Value of var1 and var2 is: 3, 2
               After swap Value of var1 and var2 is: 3, 2
```

```
// C program to show use of call by Reference
#include <stdio.h>
 void swap(int *var1, int *var2)
{
  int temp = *var1;
  *var1 = *var2;
  *var2 = temp;
}
 int main()
```

```
{
  int var1 = 3, var2 = 2;
  printf("Before swap Value of var1 and var2 is: %d, %d\n",
          var1, var2);
  swap(&var1, &var2);
  printf("After swap Value of var1 and var2 is: %d, %d",
          var1, var2);
  return 0;
}
```

OUTPUT:

```
Before swap Value of var1 and var2 is: 3, 2
After swap Value of var1 and var2 is: 2, 3
```

5.1.7 Recursion in C

Recursion is the process which comes into existence when a function calls a copy of itself to work on a smaller problem. Any function which calls itself is called recursive function, and such function calls are called recursive calls. Recursion involves several numbers of recursive calls. However, it is important to impose a termination condition of recursion. Recursion code is shorter than iterative code however it is difficult to understand.

Recursion cannot be applied to the entire problem, but it is more useful for the tasks that can be defined in terms of similar subtasks. For Example, recursion may be applied to sorting, searching, and traversal problems. Generally, iterative solutions are more efficient than recursion since function call is always overhead. Any problem that can be solved recursively can also be solved iteratively. However, some problems are best suited to be solved by the recursion, for example, tower of Hanoi, Fibonacci series, factorial finding, etc. In the following example, recursion is used to calculate the factorial of a number.

```
 #include <stdio.h>
 int fact (int);
 int main()
 {
 int n,f;
 printf("Enter the number whose factorial you want to calculate?");  scanf("%d",&n);
   f = fact(n);
```

```
    printf("factorial = %d",f);
}
  int fact(int n)
  {
      if (n==0)
      {
          return 0;
      }
      else if ( n == 1)
      {
          return 1;
      }
      else
      {
          return n*fact(n-1);
   }  }
```

Output:

```
Enter the number whose factorial you want to calculate? 5
factorial = 120
```

Memory allocation of Recursive method: Each recursive call creates a new copy of that method in the memory. Once some data is returned by the method, the copy is removed from the memory. Since all the variables and other stuff declared inside function get stored in the stack, therefore a separate stack is maintained at each recursive call. Once the value is returned from the corresponding function, the stack gets destroyed. Recursion involves so much complexity in resolving and tracking the values at each recursive call. Therefore we need to maintain the stack and track the values of the variables defined in the stack.

Advantages of Using Functions

The functions in the C programming language offer the following advantages:

- One of the primary achievements of the C functions is reusability.
- When we make use of the functions, then we can easily avoid the rewriting of the same code/ logic time and again multiple times in any program.

- The calling of C functions may appear as many numbers of times as we want in any program. And we can do so from any place in the program that is given to us.
- We can perform the tracking of a large C program pretty easily if we divide it into various functions.
- However, remember that the function calling always acts as an overhead in the case of a C program.

5.1.8 Pointers and Functions

As discussed earlier we can create a pointer of any data type such as int, char, float. Similarly, we can also create a pointer pointing to a function. The code of a function always resides in memory, which means that the function has some address. We can get the address of memory by using the function pointer. C programming allows to create a pointer pointing to the function, which can be further passed as an argument to the function. We can create a function pointer as follows:

Syntax: (type) (*pointer_name)(parameter);

In the above syntax, **type** is the variable type which is returned by the function, ***pointer_name** is the function pointer, and the **parameter** is the list of the argument passed to the function.

Example: **int** (*ip) (**int**);

In the above declaration, *ip is a pointer that points to a function which returns an int value and accepts an integer value as an argument.

float (*fp) (**float**);

In the above declaration, ***fp** is a pointer that points to a function that returns a float value and accepts a float value as an argument.

NOTE: The declaration of a function is similar to the declaration of a function pointer except that the pointer is preceded by a **'*'**.

```
//Program to demonstrate the Function to Pointers
#include <stdio.h>
int add(int,int);
int main()
{
  int a,b;
  int (*ip)(int,int);
  int result;
```

```
   printf("Enter the values of a and b : ");
   scanf("%d %d",&a,&b);
   ip=add;
   result=(*ip)(a,b);
   printf("Value after addition is : %d",result);
    return 0;
}
int add(int a,int b)
{
    int c=a+b;
    return c;
}
```

Output:

```
Enter the values of a and b: 4 55
Value after addition is: 59
```

5.2 Strings in C

The string can be defined as the one-dimensional array of characters terminated by a null ('\0'). The character array or the string is used to manipulate text such as word or sentences. Each character in the array occupies one byte of memory, and the last character must always be 0. The termination character ('\0') is important in a string since it is the only way to identify where the string ends. When we define a string as char s[10], the character s[10] is implicitly initialized with the null in the memory.

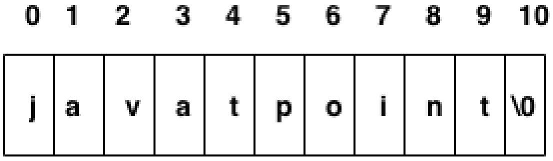
There are two ways to declare a string in c language.

1. By char array
2. By string literal

Let's see the example of declaring **string by char array** in C language.

```
char ch[10]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
```

As we know, array index starts from 0, so it will be represented as in the figure given below:



0	1	2	3	4	5	6	7	8	9	10
j	a	v	a	t	p	o	i	n	t	\0

While declaring string, size is not mandatory. So we can write the above code as given below:

```
char ch[]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n', 't', '\0'};
```

We can also define the **string by the string literal** in C language. For example:

```
char ch[]="javatpoint";
```

In such case, '\0' will be appended at the end of the string by the compiler.

5.2.1 Difference between char array and string literal

There are two main differences between char array and literal.

- We need to add the null character '\0' at the end of the array whereas; it is appended internally by the compiler in the case of the character array.
- The string literal cannot be reassigned to another set of characters whereas; we can reassign the characters of the array.

String Example in C

Let's see a simple example where a string is declared and being printed. The '%s' is used as a format specifier for the string in c language.

```
#include<stdio.h>
#include <string.h>
int main(){
char ch[11]={'j', 'a', 'v', 'a', 't', 'p', 'o', 'i', 'n',          't', '\0'};
 char ch2[11]="javatpoint";
 printf("Char Array Value is: %s\n", ch);
 printf("String Literal Value is: %s\n", ch2);
 return 0;
}
```

Output

```
Char Array Value is: javatpoint
```

String Literal Value is: javatpoint

5.2.2 Traversing String

Traversing the string is one of the most important aspects in any of the programming languages. We may need to manipulate a very large text which can be done by traversing the text. Traversing string is somewhat different from the traversing an integer array. We need to know the length of the array to traverse an integer array, whereas we may use the null character in the case of string to identify the end the string and terminate the loop.

Hence, there are two ways to traverse a string.

- By using the length of string
- By using the null character

- **Using the length of string**

Let's see an example of counting the number of vowels in a string.

```
#include<stdio.h>
void main ()
{
   char s[11] = "javatpoint";
   int i = 0;
   int count = 0;
   while(i<11)
   {
      if(s[i]=='a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'u' || s[i] == 'o')
      {
         count ++;
      }
      i++;
   }
   printf("The number of vowels %d",count);
}
```

Output

```
The number of vowels 4
```

- **Using the null character**

    Let's see the same example of counting the number of vowels by using the null character.

```
#include<stdio.h>
void main ()
    {
   char s[11] = "javatpoint";
   int i = 0;
   int count = 0;
   while(s[i] != NULL)
    {
 if(s[i]=='a' || s[i] == 'e' || s[i] == 'i' || s[i] == 'u' || s[i] == 'o')
      {
         count ++;   }
      i++;
     }
   printf("The number of vowels %d",count);
    }
```

Output

```
The number of vowels 4
```

5.2.3 Accepting string as the input

```
#include<stdio.h>
void main ()
{
   char s[20];
   printf("Enter the string?");
   scanf("%s",s);
   printf("You entered %s",s);
}
```

Output

```
Enter the string? C Programming
You entered C Programming
```

NOTE:

- The compiler doesn't perform bounds checking on the character array. Hence, there can be a case where the length of the string can exceed the dimension of the character array which may always overwrite some important data.
- Instead of using scanf, we may use gets() which is an inbuilt function defined in a header file string.h. The gets() is capable of receiving only one string at a time.

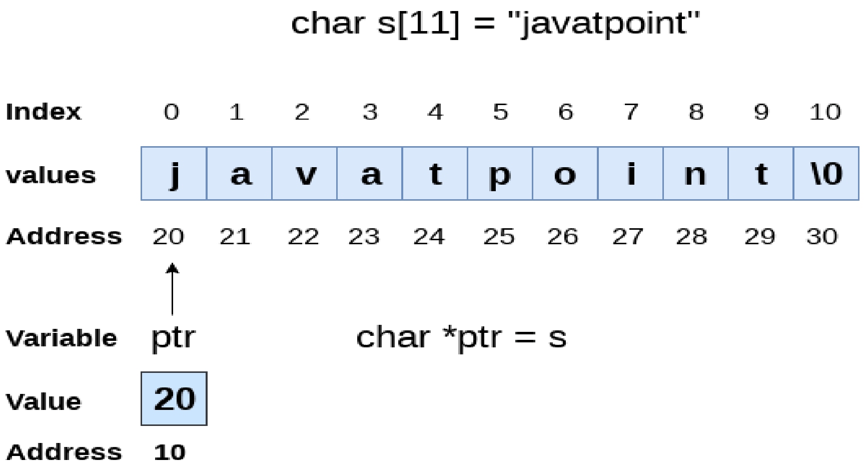
5.2.4 Pointers with strings

We have used pointers with the array, functions, and primitive data types so far. However, pointers can be used to point to the strings. There are various advantages of using pointers to point strings. Let us consider the following example to access the string via the pointer.

```
#include<stdio.h>
void main ()
{
   char s[11] = "javatpoint";
   char *p = s; // pointer p is pointing to string s.
   printf("%s",p);
}
```

Output

Javatpoint



As we know that string is an array of characters, the pointers can be used in the same way they were used with arrays. In the above example, p is declared as a pointer to the array of characters s. P affects similar to s since s is the base address of the string and treated as a pointer internally. However, we cannot change the content of s or copy the content of s into another string directly. For this purpose, we need to use the pointers to store the strings. In the following example, we have shown the use of pointers to copy the content of a string into another.

```
#include<stdio.h>
void main ()
  {
  char *p = " C Programming ";
  printf("String p: %s\n",p);
  char *q;
  printf("copying the content of p into q...\n");
  q = p;
  printf("String q: %s\n",q);
  }
```

Output

```
String p: C Programming
copying the content of p into q...
      String q: C Programming
```

Once a string is defined, it cannot be reassigned to another set of characters. However, using pointers, we can assign the set of characters to the string. Consider the following example.

```
 #include<stdio.h>
 void main ()
 {
   char *p = " C Programming ";
   printf("Before assigning: %s\n",p);
   p = "hello";
   printf("After assigning: %s\n",p);
 }
```

Output

```
Before assigning: C Programming
After assigning: hello
```

5.2.5 String Functions

gets() function: enables the user to enter some characters followed by the enter key. All the characters entered by the user get stored in a character array. The null character is added to the array to make it a string. The gets() allows the user to enter the space-separated strings. It returns the string entered by the user.

Declaration: char[] gets(char[]);

Reading string using gets()

```
#include<stdio.h>
void main ()
{
   char s[30];
   printf("Enter the string? ");
   gets(s);
   printf("You entered %s",s);
     }
```

Output

```
Enter the string?
javatpoint is the best
You entered javatpoint is the best
```

The gets() function is risky to use since it doesn't perform any array bound checking and keep reading the characters until the new line (enter) is encountered

puts() function is very much similar to printf() function. The puts() function is used to print the string on the console which is previously read by using gets() or scanf() function. The puts() function returns an integer value representing the number of characters being printed on the console. Since, it prints an additional newline character with the string, which moves the cursor to the new line on the console, the integer value returned by puts() will always be equal to the number of characters present in the string plus 1.

Declaration: **int** puts(**char**[])

Let's see an example to read a string using gets() and print it on the console using puts().

```
#include<stdio.h>
#include <string.h>
int main(){
char name[50];
printf("Enter your name: ");
gets(name); //reads string from user
printf("Your name is: ");
puts(name);  //displays string
return 0;
}
```

Output:

```
Enter your name: ABC
Your name is: ABC
```

There are many important string functions defined in "string.h" library.

No.	Function	Description
1)	strlen(string_name)	returns the length of string name.
2)	strcpy(destination, source)	copies the contents of source string to destination string.
3)	strcat(first_string, second_string)	concats or joins first string with second string. The result of the string is stored in first string.
4)	strcmp(first_string, second_string)	compares the first string with second string. If both strings are same, it returns 0.
5)	strrev(string)	returns reverse string.
6)	strlwr(string)	returns string characters in lowercase.
7)	strupr(string)	returns string characters in uppercase.

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- B

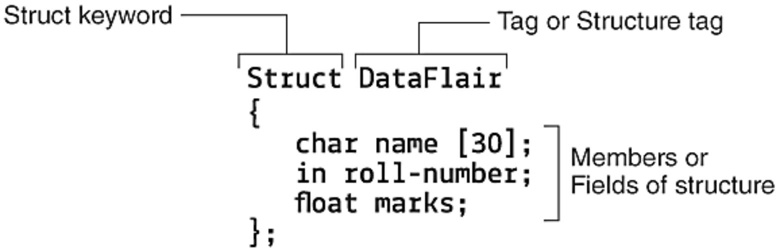
UNIT 6: USER DEFINED DATA TYPES IN C

The data types that are defined by the user depending upon the use case of the programmer are termed user-defined data types. C offers a set of constructs that allow users to define their data types. These user-defined data types are constructed using a combination of fundamental data types and derived data types. Fundamental data types are basic built-in types of C programming language. These are integer data type (int), floating data type (float), and character data type (char). Derived data types are derived from fundamental data types, like functions, arrays, and pointers in the C programming language. For example, an array is a derived data type as it contains elements of a similar data type and acts like a new data type in the C programming language. User-defined data types are created by the user using a combination of fundamental and derived data types in the C programming language. To create a user-defined data type, the C programming language provides following five constructs to define new data type named as:

- Structures (struct)
- Union
- Type definitions (Typedef)
- Enumerations (enum)
- Empty data type (void)

6.1 Structure

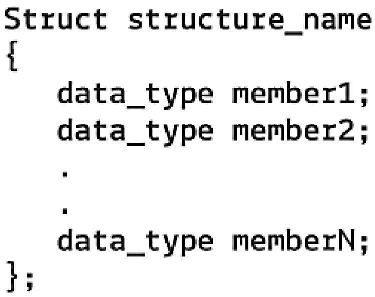
Structures are the user-defined data type, which allow us to collect the group of different data types. Here, all the individual components or elements of structures are known as a member. ‘struct’ keyword is used to create a structure.



```
Struct DataFlair
{
   char name [30];
   in roll-number;
   float marks;
};
```

Structure data type is used to store dates of different attributes of different data types.

Syntax to Define the Structure in C:



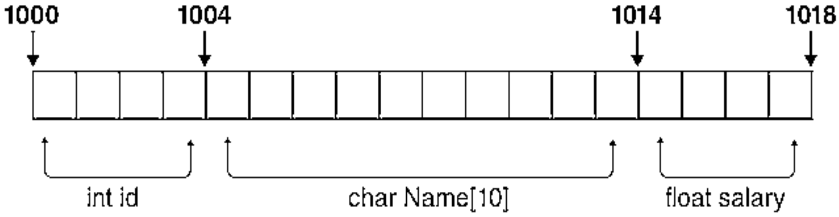
```
Struct structure_name
{
   data_type member1;
   data_type member2;
   .
   .
   data_type memberN;
};
```

Example of Structure in C:

```
struct student
{
int roll_number;
char name[20];
float percentage;
};
```

In this example structure student is keeping the information of a student which consists of three data fields, roll number, name, and percentage. These fields are known as structure elements or members. These elements are of different data types.

Memory Allocation of Structure:



6.1.1 Structure Variables Declaration

A structure variable can either be declared with structure declaration or as a separate declaration like basic types. There are two ways to declare structure variable:

1. By struct keyword within main() function
2. By declaring a variable at the time of defining the structure.

Syntax:

```
struct struct_name var_name;
or
struct struct_name
 {
DataType member1_name;
DataType member2_name;
DataType member3_name;
…
} var_name;
```

// A variable declaration with structure declaration.

```
struct Point
{
  int x, y;
} p1;  // The variable p1 is declared with 'Point'

// A variable declaration like basic data types
struct Point
{
  int x, y;
};

int main()
{
  struct Point p1;
// The variable p1 is declared like a normal variable
}
```

NOTE: Structure members cannot be initialized with declaration.

6.1.2 Accessing Structure Data Members

There are two ways to access structure members:

1. By **.** (member or dot operator)
2. By **-->** (structure pointer operator)

- Structure members are accessed using dot (.) operator.

Syntax:

```
var_name.member1_name;
var_name.member2_name;
```

```
#include <stdio.h>
 struct Point {
   int x, y;
};
int main()
{
   struct Point p1 = { 0, 1 };

   // Accessing members of point p1
   p1.x = 20;
   printf("x = %d, y = %d", p1.x, p1.y);

   return 0;
}
```

Output:

```
x = 20, y = 1
```

6.1.3 Array of Structures

An array of structures in C language can be defined as the collection of multiple structure variables where each variable contains information about different entities. The array of structures in C language is used to store information about multiple entities of different data types. The array of structures is also known as the collection of structures.

//**Program to illustrate an array of structures** #include<stdio.h>

```
#include <string.h>
struct student{
int rollno;
char name[10];
};
int main(){
int i;
struct student st[5];
printf("Enter Records of 5 students");
for(i=0;i<5;i++){
printf("\nEnter Rollno:");
scanf("%d",&st[i].rollno);
printf("\nEnter Name:");
scanf("%s",&st[i].name);
}
printf("\nStudent Information List:");
 for(i=0;i<5;i++){
 printf("\nRollno:%d, Name:%s",st[i].rollno,st[i].name);
 }
  return 0;
 }
```

Output:

```
Enter Rollno:1
Enter Name:Sunny
Enter Rollno:2
Enter Name:Ranjan
Enter Rollno:3
Enter Name:Vishal
Enter Rollno:4
Enter Name:Jhanvi
Enter Rollno:5
Enter Name:Saroj
```

```
Student Information List:
Rollno:1, Name:Sunny
Rollno:2, Name:Ranjan
Rollno:3, Name:Vishal
Rollno:4, Name:Jhanvi
Rollno:5, Name:Saroj
```

6.1.4 Nested Structure

C provides the feature of nesting one structure within another structure by using which, complex data types are created. For example, we may need to store the address of an entity employee in a structure. The attribute address may also have the subparts as street number, city, state, and pin code. Hence, to store the address of the employee, we need to store the address of the employee into a separate structure and nest the structure address into the structure employee. Consider the following program.

```
#include<stdio.h>
struct address
   {
    char city[20];
    int pin;
    char phone[14];
    };
struct employee
    {
    char name[20];
    struct address add;
     };
void main ()
    {
 struct employee emp;
 printf("Enter employee information\n");
 scanf("%s %s %d %s",emp.name,emp.add.city, &emp.add.pin, emp.add.phone);
 printf("Printing the employee information....\n");
 printf("name: %s\nCity: %s\nPincode: %d\nPhone: %s",emp.name,emp.add.city,emp.add.p
in,emp.add.phone);
}
```

Output:

```
Enter employee information
Aman
Chandigarh
160014
8427526788
Printing the employee information....
name: Aman
City: Chandigarh
Pincode: 160014
Phone: 8427526788
```

6.1.5 **Passing structure to function**

Just like other variables, a structure can also be passed to a function. We may pass the structure members into the function or pass the structure variable at once. Consider the following example to pass the structure variable employee to a function display() which is used to display the details of an employee.

```
  #include<stdio.h>
  struct address
     {
     char city[20];
     int pin;
     char phone[14];
      };
  struct employee
      {
     char name[20];
     struct address add;
      };
  void display(struct employee);
  void main ()
```

```
    {
     struct employee emp;
     printf("Enter employee information?\n");
     scanf("%s %s %d %s",emp.name,emp.add.city, &emp.add.pin, emp.add.phone);
     display(emp);
     }
  void display(struct employee emp)
    {
   printf("Printing the details....\n");
   printf("%s %s %d %s",emp.name,emp.add.city,emp.add.pin,emp.add.phone);
   }
```

6.1.6 Structures Limitations

In C language, Structures provide a method for packing together data of different types. A Structure is a helpful tool to handle a group of logically related data items. However, C structures have some limitations.

- The C structure does not allow the struct data type to be treated like built-in data types:
- C Structures do not permit data hiding. Structure members can be accessed by any function, anywhere in the scope of the Structure
- C structures do not permit functions inside Structure
- C Structures cannot have static members inside their body
- C Programming language does not support access modifiers. So they cannot be used in C Structures.
- Structures in C cannot have a constructor inside Structures.

We cannot use operators like +,- etc. on Structure variables. For example, consider the following code:

```
struct number
{
   float x;
};
int main()
{
        struct number n1, n2, n3;
        n1.x = 4;
        n2.x = 3;
        n3 = n1 + n2;
        return 0;
       }

    /*Output:
    prog.c: In function 'main':
    prog.c:10:7: error:
    invalid operands to binary + (have 'struct number' and
    'struct number') n3=n1+n2;
    */
```

But we can use arithmetic operation on structure variables as mentioned in example given below:

```
// Use of arithmetic operator in structure
#include <stdio.h>
struct number {
   float x;
  };
int main()
 {
   struct number n1, n2, n3;
   n1.x = 4;
   n2.x = 3;
   n3.x = (n1.x) + (n2.x);
   printf("\n%f", n3.x);
   return 0;
   }
```

Output:

```
6.000000
```

6.2 Union

Union can be defined as a user-defined data type which is a collection of different variables of different data types in the same memory location. The union can also be defined as many members, but only one member can contain a value at a particular point in time. Union is a user-defined data type, but unlike structures, they share the same memory location.

```
struct abc
{
  int a;
  char b;
}
```

The above code is the user-defined structure that consists of two members, i.e., 'a' of type **int** and 'b' of type **character**. When we check the addresses of 'a' and 'b', we found that their addresses are different. Therefore, we conclude that the members in the structure do not share the same memory location.

```
union abc
{
  int a;
  char b;
}
```

When we define the union, then we found that union is defined in the same way as the structure is defined but the difference is that union keyword is used for defining the union data type, whereas the struct keyword is used for defining the structure. The union contains the data members, i.e., 'a' and 'b', when we check the addresses of both the variables then we found that both have the same addresses. It means that the union members share the same memory location.

NOTE: The size of the union is based on the size of the largest member of the union.

Example:

```
union abc{
int a;
char b;
float c;
double d;
};
```

```
int main()
{
  printf("Size of union abc is %d", sizeof(union abc));
  return 0;
}
```

Output:

```
Size of union abc is 8 bytes
```

As we know, the size of int is 4 bytes, size of char is 1 byte, size of float is 4 bytes, and the size of double is 8 bytes. As the double data type occupies the largest memory among all the four data types, so total 8 bytes will be allocated in the memory. Therefore, the output of the above program would be 8 bytes.

6.3 Difference between Structure and Union in C

Parameter	**Structure**	**Union**
Keyword	**struct** keyword is used to define a Structure.	**union** keyword is used to define a Union.
Internal Implementation	The implementation of Structure in C occurs internally- because it contains separate memory locations allotted to every input member.	In the case of a Union, the memory allocation occurs for only one member with the largest size among all the input variables. It shares the same location among all these members/objects.
Accessing Members	A user can access individual members at a given time.	A user can access only one member at a given time.

Syntax	The Syntax of declaring a Structure in C is:
	struct [structure name]
	{
	type element_1;
		The Syntax of declaring a Union in C is:
		union [union name]
		{
		type element_1;
		type element_2;

	type element_2; . . } variable_1, variable_2, …;	. . } variable_1, variable_2, …;
Size	A Structure does not have a shared location for all of its members. It makes the size of a Structure to be greater than or equal to the sum of the size of its data members.	A Union does not have a separate location for every member in it. It makes its size equal to the size of the largest member among all the data members.
Value Altering	Altering the values of a single member does not affect the other members of a Structure.	When you alter the values of a single member, it affects the values of other members.
Storage of Value	In the case of a Structure, there is a specific memory location for every input data member. Thus, it can store multiple values of the various members.	In the case of a Union, there is an allocation of only one shared memory for all the input data members. Thus, it stores one value at a time for all of its members.
Initialization	In the case of a Structure, a user can initialize multiple members at the same time.	In the case of a Union, a user can only initiate the first member at a time.

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- B

UNIT 7: OBJECT ORIENTED PROGRAMMING CONCEPT

7.1 Need of an Object-Oriented Programming

The earlier approaches to programming were not that good, and there were several limitations as well. Like in procedural-oriented programming, you cannot reuse the code again in the program, and there was the problem of global data access, and the approach couldn’t solve the real-world problems very well.

In object-oriented programming, it is easy to maintain the code with the help of classes and objects. Using inheritance, there is code reusability, i.e., you don’t have to write the same code again and again, which increases the simplicity of the program. Concepts like encapsulation and abstraction provide data hiding as well.

Problems solved by object-oriented approach

- **Code reusability**: No need to write the same code again and again, which increases the program’s simplicity. As the same code can be executed many times in the program.
- **Data hiding**: Data can be hidden from the outside world.

7.2 C++ and its Application:

C++ is a special-purpose programming language developed by **Bjarne Stroustrup** at Bell Labs circa 1980. C++ language is very similar to C language, and it is so compatible with C that it can run 99% of C programs without changing any source of code though C++ is an object-oriented programming language, so it is safer and well-structured programming language than C.

Applications of C++ Programming

As mentioned before, C++ is one of the most widely used programming languages. It has it's presence in almost every area of software development. I'm going to list few of them here:

- **Application Software Development** - C++ programming has been used in developing almost all the major Operating Systems like Windows, Mac OSX and Linux. Apart from the operating systems, the core part of many browsers like Mozilla Firefox and Chrome have been written using C++. C++ also has been used in developing the most popular database system called MySQL.
- **Programming Languages Development** - C++ has been used extensively in developing new programming languages like C#, Java, JavaScript, Perl, UNIX's C Shell, PHP and Python, and Verilog etc.
- **Computation Programming** - C++ is the best friends of scientists because of fast speed and computational efficiencies.
- **Games Development** - C++ is extremely fast which allows programmers to do procedural programming for CPU intensive functions and provides greater control over hardware, because of which it has been widely used in development of gaming engines.
- **Embedded System** - C++ is being heavily used in developing Medical and Engineering Applications like softwares for MRI machines, high-end CAD/CAM systems etc.

7.3 OOPs Concepts in C++

The major purpose of C++ programming is to introduce the concept of object orientation to the C programming language. Object-oriented programming, as the name suggests uses objects in programming. Object-oriented programming aims to implement real-world entities like inheritance, hiding, polymorphism, etc. in programming. The main aim of OOP is to bind together the data and the functions that operate on them so that no other part of the code can access this data except that function. There are some basic concepts that act as the building blocks of OOPs in C++ named as Class, Objects, Encapsulation, Abstraction, Polymorphism, Inheritance, Dynamic Binding and Message Passing, the description of each is given below:

7.3.1 Class

The building block of C++ that leads to Object-Oriented programming is a Class. It is a user-defined data type, which holds its own data members and member functions, which can be accessed and used by creating an instance of that class. A class is like a blueprint for an object. For Example: Consider the Class of Cars. There may be many cars with different names and brands but all of them will share some common properties like all of them will have 4 wheels, Speed Limit, Mileage range, etc. So here, the Car is the class, and wheels, speed limits, and mileage are their properties.

- A Class is a user-defined data type that has data members and member functions.
- Data members are the data variables and member functions are the functions used to manipulate these variables together these data members and member functions define the properties and behavior of the objects in a Class.
- In the above example of class Car, the data member will be speed limit, mileage, etc and member functions can apply brakes, increase speed, etc.

We can say that a Class in C++ is a blueprint representing a group of objects which shares some common properties and behaviors.

A class definition starts with the keyword **class** followed by the class name; and the class body, enclosed by a pair of curly braces. A class definition must be followed either by a semicolon or a list of declarations. For example, we defined the Box data type using the keyword **class** as follows :

```
class Box
  {
  public:
    double length;   // Length of a box
    double breadth;  // Breadth of a box
    double height;   // Height of a box
    };
```

The keyword **public** determines the access attributes of the members of the class that follows it. A public member can be accessed from outside the class anywhere within the scope of the class object.

7.3.2 Objects

An Object is an identifiable entity with some characteristics and behavior. An Object is an instance of a Class. When a class is defined, no memory is allocated but when it is instantiated (i.e. an object is created) memory is allocated. A class provides the blueprints for objects, so basically an object is created from a class. We declare objects of a class with exactly the same sort of declaration that we declare variables of basic types. Following statements declare two objects of class Box:

```
Box Box1;          // Declare Box1 of type Box
Box Box2;          // Declare Box2 of type Box
```

Both of the objects Box1 and Box2 will have their own copy of data members.

Accessing the Data Members: The public data members of objects of a class can be accessed using the direct member access operator (.).

Example:

```
#include <iostream.h>
class Box
   {
   public:
    double length;   // Length of a box
    double breadth;  // Breadth of a box
    double height;   // Height of a box
    };
int main() {
  Box Box1;        // Declare Box1 of type Box
  Box Box2;        // Declare Box2 of type Box
  double volume = 0.0;     // Store the volume of a box here
  // box 1 specification
  Box1.height = 5.0;
  Box1.length = 6.0;
  Box1.breadth = 7.0;
// box 2 specification
  Box2.height = 10.0;
  Box2.length = 12.0;
  Box2.breadth = 13.0;
  // volume of box 1
```

```
   volume = Box1.height * Box1.length * Box1.breadth;
   cout << "Volume of Box1 : " << volume <<endl;
   // volume of box 2
   volume = Box2.height * Box2.length * Box2.breadth;
   cout << "Volume of Box2 : " << volume <<endl;
   return 0;
  }
```

Output:

```
Volume of Box1:210
Volume of Box2: 1560
```

NOTE: Private and Protected members cannot be accessed directly using direct member access operator (.)

7.3.3 Encapsulation

In normal terms, Encapsulation is defined as wrapping up data and information under a single unit. In Object-Oriented Programming, Encapsulation is defined as binding together the data and the functions that manipulate them. Consider a real-life example of encapsulation, in a company, there are different sections like the accounts section, finance section, sales section, etc. The finance section handles all the financial transactions and keeps records of all the data related to finance. Similarly, the sales section handles all the sales-related activities and keeps records of all the sales. Now there may arise a situation when for some reason an official from the finance section needs all the data about sales in a particular month. In this case, he is not allowed to directly access the data of the sales section. He will first have to contact some other officer in the sales section and then request him to give the particular data. This is what encapsulation is. Here the data of the sales section and the employees that can manipulate them are wrapped under a single name “sales section”.

Features of Encapsulation

8 We cannot access any function from the class directly. We need an object to access that function that is using the member variables of that class.

9 The function which we are making inside the class must use only member variables, only then it is called encapsulation.

10 If we don’t make a function inside the class which is using the member variable of the class then we don’t call it encapsulation.

11 Increase in the security of data, to restrict and control the modification of our data members.

In C++, encapsulation can be implemented using classes and access modifiers. The following example explain the concept of Encapsulation:

```
// Encapsulation
#include <iostream>
using namespace std;

class Encapsulation {
private:
    // Data hidden from outside world
    int x;

public:
    // Function to set value of
    // variable x
    void set(int a) { x = a; }

    // Function to return value of
    // variable x
    int get() { return x; }
};

// Driver code
int main()
{
    Encapsulation obj;
    obj.set(5);
    cout << obj.get();
    return 0;
}
```

Output: 5

In the above program, the variable x is made private. This variable can be accessed and manipulated only using the functions get() and set() which are present inside the class. Thus we can say that here, the variable x and the functions get() and set() are bound together which is nothing but encapsulation.

7.3.4 Abstraction:

Data abstraction is one of the most essential and important features of object-oriented programming in C++. Abstraction means displaying only essential information and hiding the details. Data abstraction refers to providing only essential information about the data to the outside world, hiding the background details or implementation. Consider a real-life example of a man driving a car. The man only knows that pressing the accelerator will increase the speed of the car or applying brakes will stop the car but he does not know how on pressing the accelerator the speed is actually increasing, he does not know about the inner mechanism of the car or the implementation of an accelerator, brakes, etc. in the car. This is what abstraction is.

- **Abstraction using Classes**: We can implement Abstraction in C++ using classes. The class helps us to group data members and member functions using available access specifiers. A Class can decide which data member will be visible to the outside world and which is not.
- **Abstraction in Header files**: One more type of abstraction in C++ can be header files. For example, consider the pow() method present in math.h header file. Whenever we need to calculate the power of a number, we simply call the function pow() present in the math.h header file and pass the numbers as arguments without knowing the underlying algorithm according to which the function is actually calculating the power of numbers.

Features of Data Abstraction:

- Helps the user to avoid writing the low-level code
- Avoids code duplication and increases reusability.
- Can change the internal implementation of the class independently without affecting the user.

- Helps to increase the security of an application or program as only important details are provided to the user.
- It reduces the complexity as well as the redundancy of the code, therefore increasing the readability.

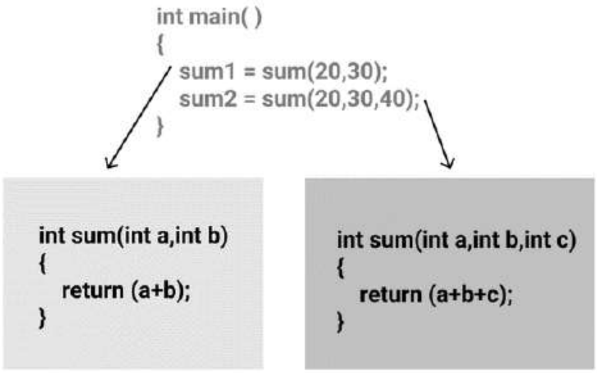
In simple words, Abstraction avoids unnecessary information or irrelevant details and shows only that specific part which the user wants to see.

7.3.5 Polymorphism

The word polymorphism means having many forms. In simple words, we can define polymorphism as the ability of a message to be displayed in more than one form. A person at the same time can have different characteristics. A man at the same time is a father, a husband, and an employee. So the same person possesses different behavior in different situations. This is called polymorphism. An operation may exhibit different behaviors in different instances. The behavior depends upon the types of data used in the operation. C++ supports operator overloading and function overloading.

- Operator Overloading: The process of making an operator exhibit different behaviors in different instances is known as operator overloading.
- Function Overloading: Function overloading is using a single function name to perform different types of tasks. Polymorphism is extensively used in implementing inheritance.

Example: Suppose we have to write a function to add some integers, sometimes there are 2 integers, and sometimes there are 3 integers. We can write the Addition Method with the same name having different parameters, the concerned method will be called according to parameters.



```
int main( )
{
  sum1 = sum(20,30);
  sum2 = sum(20,30,40);
}
```

```
int sum(int a,int b)
{
   return (a+b);
}
```

```
int sum(int a,int b,int c)
{
   return (a+b+c);
}
```

7.3.6 Inheritance

Inheritance supports the concept of “reusability”, i.e. when we want to create a new class and there is already a class that includes some of the code that we want, we can derive our new class from the existing class. By doing this, we are reusing the fields and methods of the existing class. Inheritance allows to define a class in terms of another class, which makes it easier to create and maintain an application. This also provides an opportunity to reuse the code functionality and fast implementation time. When creating a class, instead of writing completely new data members and member functions, the programmer can designate that the new class should inherit the members of an existing class. This existing class is called the **base** class, and the new class is referred to as the **derived** class.

8 **Base and Derived Classes:** A class can be derived from more than one classes, which means it can inherit data and functions from multiple base classes. To define a derived class, we use a class derivation list to specify the base class(es). A class derivation list names one or more base classes and has the form:

class derived-class: access-specifier base-class

Where access-specifier is one of **public, protected,** or **private**, and base-class is the name of a previously defined class. If the access-specifier is not used, then it is private by default.

Access Control in Inheritance: A derived class can access all the non-private members of its base class. Thus base-class members that should not be accessible to the member functions of derived classes should be declared private in the base class. We can summarize the different access types according to - who can access them in the following way:

Access	Public	protected	private
Same class	Yes	yes	yes
Derived classes	Yes	yes	no
Outside classes	Yes	no	no

When deriving a class from a base class, the base class may be inherited through **public, protected** or **private** inheritance. The type of inheritance is specified by the access-specifier as explained above. While using different type of inheritance, following rules are applied :

9 **Public Inheritance**: When deriving a class from a public base class, public members of the base class become public members of the derived class and protected members of the base class become protected members of the derived class. A base class's private members are never accessible directly from a derived class, but can be accessed through calls to the public and protected members of the base class.

10 **Protected Inheritance**: When deriving from a protected base class, public and protected members of the base class become protected members of the derived class.

11 **Private Inheritance**: When deriving from a private base class, public and protected members of the base class become private members of the derived class.

Types Of Inheritance: C++ supports five types of inheritance.

- Single inheritance
- Multiple inheritance
- Hierarchical inheritance
- Multilevel inheritance
- Hybrid inheritance

The details about each section discussed in later sections.

7.3.7 Dynamic Binding and Message Passing

In dynamic binding, the code to be executed in response to the function call is decided at runtime. C++ has virtual functions to support this. Because dynamic binding is flexible, it avoids the drawbacks of static binding, which connected the function call and definition at build time.

Example:

```
// C++ Program to Demonstrate the Concept of Dynamic binding
// with the help of virtual function
#include <iostream>
using namespace std;
class GFG
{
public:
    void call_Function() // function that call print
    {
        print();
    }
    void print() // the display function
    {
        cout << "Printing the Base class Content" << endl;
    }
};
class GFG2 : public GFG // GFG2 inherit a publicly
{
public:
    void print() // GFG2's display
    {
        cout << "Printing the Derived class Content"
             << endl;
    }
   };
int main()
  {
    GFG geeksforgeeks; // Creating GFG's pbject
    geeksforgeeks.call_Function(); // Calling call_Function
    GFG2 geeksforgeeks2; // creating GFG2 object
    geeksforgeeks2.call_Function(); // calling call_Function
                                    // for GFG2 object
    return 0;
    }
```

Output:

```
            Printing the Base class Content
            Printing the Base class Content
```

As we can see, the print() function of the parent class is called even from the derived class object. To resolve this we use virtual functions.

- **Message Passing**

Objects communicate with one another by sending and receiving information. A message for an object is a request for the execution of a procedure and therefore will invoke a function in the receiving object that generates the desired results. Message passing involves specifying the name of the object, the name of the function, and the information to be sent. A message cannot go automatically it creates an interface, which means it creates an interface for an object. The interface provides the abstraction over the message means hide the implementation. So we get to know, An interface is a set of operations that a given object can perform.

All communication between objects is done via message is called message passing, as people exchange information similarly the sending and receiving of information by the object is said to be message passing, to make possible message passing the following things have to be followed to be done :

- User have to create classes that define objects and its behavior.
- Then Creating the objects from class definitions.
- Calling and connecting the communication among objects.

7.3.8 Access Specifiers in C++

Data hiding is an important concept of Object-Oriented Programming, implemented with these Access modifiers' help. It is also known as Access Specifier. Access Specifiers in a class decide the accessibility of the class members, like variables or methods in other classes. That is, it will decide whether the members or methods will get directly accessed by the blocks present outside the class or not, depending on the type of Access Specifier. In a program, we need to create methods or variables that can be accessed by the object of the same class or accessible in the entire program. And Access Modifiers help us to specify that. There are three types of access modifiers in C++:

- Public
- Private
- Protected

To manipulate and fetch the data, a public specifier is used, and to protect the data from outside members, a private specifier is used so that the crucial or sensitive data cannot be tampered with or leaked outside of its block.

Syntax of Declaring Access Specifiers in C++ is given below:

```
class ClassName
{
private:
// Declare private members/methods here.
public:
// Declare public members/methods here.
protected:
// Declare protected members/methods here.
};
```

The following section describes about each access specifier in detail:

7.3.8.1 Public: All the class members declared under the public specifier will be available to everyone. The data members and member functions declared as public can be accessed by other classes and functions too. The public members of a class can be accessed from anywhere in the program using the direct member access operator (.) with the object of that class.

```
// C++ program to demonstrate public
// access modifier
#include<iostream>
using namespace std;
// class definition
class Circle
{
   public:
  double radius;
 double  compute_area()
  {
  return 3.14*radius*radius;
  }
};
// main function
int main()
{
    Circle obj;
    // accessing public datamember outside class
    obj.radius = 5.5;
    cout << "Radius is: " << obj.radius << "\n";
    cout << "Area is: " << obj.compute_area();
    return 0;
}
```

Output:

```
            Radius is: 5.5
            Area is: 94.985
```

In the above example, the data member radius is declared as public so it could be accessed outside the class and thus was allowed access from inside main().

7.3.8.2 Private: The class members declared as private can be accessed only by the member functions inside the class. They are not allowed to be accessed directly by any object or function outside the class. Only the member functions or the friend functions are allowed to access the private data members of the class.

Example:

```
// C++ program to demonstrate private
// access modifier
#include<iostream>
using namespace std;
class Circle
{
    // private data member
    private:
        double radius;
    // public member function
    public:
        double  compute_area()
        {   // member function can access private
            // data member radius
            return 3.14*radius*radius;
        }
};
// main function
int main()
{
    // creating object of the class
    Circle obj;
    // trying to access private data member
    // directly outside the class
    obj.radius = 1.5;
    cout << "Area is:" << obj.compute_area();
    return 0;
}
```

Output:

```
            In function 'int main()':
            error: 'double Circle::radius' is private
            double radius;
                         ^
            error: within this context
            obj.radius = 1.5;
                         ^
```

The output of the above program is a compile time error because we are not allowed to access the private data members of a class directly from outside the class. Yet an access to obj.radius is attempted, but radius being a private data member, we obtained the above compilation error. However, we can access the private data members of a class indirectly using the public member functions of the class.

Example:

```
// C++ program to demonstrate private
// access modifier
#include<iostream>
using namespace std;
class Circle
{
    // private data member
```

```
   private:
      double radius;
   // public member function
   public:
      void compute_area(double r)
      {   // member function can access private
         // data member radius
         radius = r;
         double area = 3.14*radius*radius;
         cout << "Radius is: " << radius << endl;
         cout << "Area is: " << area;
      }
};
// main function
int main()
{
   // creating object of the class
   Circle obj;
   // trying to access private data member
   // directly outside the class
   obj.compute_area(1.5);
   return 0;
}
```

Output:

```
            Radius is: 1.5
            Area is: 7.065
```

7.3.8.3 Protected: The protected access modifier is similar to the private access modifier in the sense that it can’t be accessed outside of its class unless with the help of a friend class. The difference is that the class members declared as Protected can be accessed by any subclass (derived class) of that class as well. This access through inheritance can alter the access modifier of the elements of base class in derived class depending on the mode of Inheritance.

Example:

```
// C++ program to demonstrate
// protected access modifier
#include <bits/stdc++.h>
using namespace std;
// base class
class Parent
{
    // protected data members
    protected:
    int id_protected;
};
// sub class or derived class from public base class
class Child : public Parent
{
    public:
    void setId(int id)
    {
        // Child class is able to access the inherited
        // protected data members of base class
        id_protected = id;
    }
    void displayId()
    {
        cout << "id_protected is: " << id_protected << endl;
    }
};
// main function
int main()
{
    Child obj1;
    // member function of the derived class can
    // access the protected data members of the base class
    obj1.setId(81);
    obj1.displayId();
    return 0;
}
```

Output:

```
                    id_protected is: 81
```

\

MSc (COMPUTER SCIENCE)

COURSE: COMPUTER PROGRAMING

SECTION- B

UNIT 8: CONSTRUCTOR AND INHERITANCE IN C++

8.1 Constructor in C++

While programming, sometimes there might be the need to initialize data members and member functions of the objects before performing any operations. Data members are the variables declared in any class by using fundamental data types (like int, char, float, etc.) or derived data types (like class, structure, pointer, etc.). The functions defined inside the class definition are known as member functions. Suppose you are developing a game. In that game, each time a new player registers, we need to assign their initial location, health, acceleration, and certain other quantities to some default value. This can be done by defining separate functions for each quantity and assigning the quantities to the required default values. For this, we need to call a list of functions every time a new player registers. Now, this process can become lengthy and complicated.

Therefore, C++ provides feature of constructor to do the same, a constructor is a member function of a class that has the same name as the class name. It helps to initialize the object of a class. It can either accept the arguments or not. It is used to allocate the memory to an object of the class. It is called whenever an instance of the class is created. It can be defined manually with arguments or without arguments. There can be many constructors in a class. It can be overloaded but it cannot be inherited or virtual. There is a concept of copy constructor which is used to initialize an object from another object.

Syntax of Constructor:

```
class class_name
{
  private:
  // private members
  public:
  // declaring constructor
  class_name({parameters})
  {
    // constructor body
  }
};
```

In the above syntax, we can see the class has the name class_name and the constructor have also the same name. A constructor can have any number of parameters as per requirements. Also, there is no return type or return value of the constructor.

8.2 Characteristics of Constructors in C++:

- A constructor can be made public, private, or protected per our program's design. Constructors are mostly made public, as public methods are accessible from everywhere, thus allowing us to create the object of the class anywhere in the code. When a constructor is made private, other classes cannot create instances of the class. This is used when there is no need for object creation. Such a case arises when the class only contains static member functions (i.e., the functions which are independent of any class object and can be accessed using a class name with scope resolution operator).

- A constructor in C++ cannot be inherited. However, a derived class can call the base class constructor. A derived class (i.e., child class) contains all members and member functions (including constructors) of the base class.
- Constructor functions are not inherited, and their addresses cannot be referenced.

8.3 Difference between constructor and member function:

- Constructor name must be the same as class name but functions cannot have the same name as the class name.
- Constructors do not have a return type whereas functions must have a return type.
- Constructors are automatically called when an object is created.
- A member function can be virtual, but there is no concept of virtual constructors.
- Constructors are invoked at the time of object creation automatically and cannot be called explicitly using class objects.

8.4 Types of Constructor in C++

There are four types of constructors in C++, Default Constructors, Parameterized Constructors, Copy Constructors, Dynamic Constructors. The detail of each is given below:

8.4.1 Default Constructor: Default constructor is also known as a zero-argument constructor, as it doesn't take any parameter. It can be defined by the user if not then the compiler creates it on his own. Default constructor always initializes data members of the class with the same value they were defined.

Syntax of Default Constructor:

```
class class_name
{
  private:
  // private members

  public:

  // declaring default constructor
  class_name()
  {
    // constructor body
  }

};
```

Example of default constructor:

```
#include <iostream>
using namespace std;
class Person
{
  // declaring private class data members
private:
  string name;
  int age;

public:

  // declaring constructor
  Person()
  {
      cout<<"Default constructor is called"<<endl;
      name = "student";
      age = 12;
  }
  // display function to print the class data members value
  void display()
  {
      cout<<"Name of current object: "<<name<<endl;
      cout<<"Age of current object: "<<age<<endl;
  }
};
int main()
{
    // creating object of class using default constructor
    Person obj;
    // printing class data members
```

```
   obj.display();
   return 0;
}
```

Output:

```
Default constructor is called
Name of current object: student
Age of current object: 12
```

In the above code, we have created a class with two data members. Declared a default constructor which always initializes objects of a class with the same name and age. In the main function, we have created an object of the class and printed its data member values by using the display function.

8.4.2 Parameterized Constructor:

Parameterized constructor is used to initialize data members with the values provided by the user. This constructor is basically the upgraded version of the default constructor. We can define more than one parameterized constructor according to the need of the user, but we have to follow the rules of the function overloading, like a different set of arguments must be there for each constructor.

Syntax

```
class class_name{
  private:
  // private members

  public:

  // declaring parameterized constructor
  class_name(parameter1, parameter2,...)
  {
    // constructor body
  }
  };
```

Code to understand the working of the parameterized constructor

```
#include <iostream>
using namespace std;
class Person
{
  // declaring private class data members
private:
  string name;
  int age;
public:

  // declaring parameterized constructor of three different types
  Person(string person_name)
  {
      cout<<"Constructor to set name is called"<<endl;
      name = person_name;
      age = 12;
  }
  Person(int person_age)
  {
      cout<<"Constructor to set age is called"<<endl;
      name = "Student";
      age = person_age;
  }
  Person(string person_name, int person_age)
  {
      cout<<"Constructor for both name and age is called"<<endl;
      name = person_name;
      age = person_age;
  }
  // display function to print the class data members value
  void display()
  {
      cout<<"Name of current object: "<<name<<endl;
      cout<<"Age of current object: "<<age<<endl;
      cout<<endl;
    }
   };
int main()
    {
    // creating objects of class using parameterized constructor
    Person obj1("First person");
    // printing class data members for first object
    obj1.display();
    Person obj2(25);
    // printing class data members for second object
    obj2.display();
    Person obj3("Second person",15);
    // printing class data members for third object
    obj3.display();
    return 0;
    }
```

Output:

```
Constructor to set name is called
Name of current object: First person
Age of current object: 12

Constructor to set age is called
Name of current object: Student
Age of current object: 25

Constructor for both name and age is called
Name of current object: Second person
Age of current object: 15
```

In the above code, we have created three types of the parametric constructor, one for initialization of name only, second to initialization of age only, and third to initialize both name and age. In the main function, we have created three different types of objects and initialized them in different ways, and printed values for each of them.

8.4.3 Copy Constructor

If we have an object of a class and we want to create its copy in a new declared object of the same class, then a copy constructor is used. The compiler provides each class a default copy constructor and users can define it also. It takes a single argument which is an object of the same class.

Syntax of Copy Constructor:

```
class class_name
{
        private:
        // private members
        public:
        // declaring copy constructor
class_name(const class_name& obj)
{
        // constructor body
}
};
```

In the above syntax, we created a copy constructor which takes an object of the same class as a parameter but it is declared constant and passed as a reference because when an argument is passed as a function parameter it creates a copy for it, to create that copy compiler will again call the copy constructor, means it will call the same function and for that call again there will be a call to create copy which will take this process in never ending recursion of creating copies. To prevent such conditions we pass it as a reference.

Example of copy constructor:

```
#include <iostream>
using namespace std;
class Person
{
  // declaring private class data members
private:
  string name;
  int age;
public:
  Person(string person_name, int person_age)
```

```
    {
      cout<<"Constructor for both name and age is called"<<endl;
      name = person_name;
      age = person_age;
    }
  Person(const Person& obj)
    {
      cout<<"Copy constructor is called"<<endl;
      name = obj.name;
      age = obj.age;
    }
  // display function to print the class data members value
  void display()
    {
      cout<<"Name of current object: "<<name<<endl;
      cout<<"Age of current object: "<<age<<endl;
      cout<<endl;
  }
  };
int main()
  {
    // creating objects of class using parameterized constructor
    Person obj1("First person",25);
    // printing class data members for first object
    obj1.display();
    // creating copy of the obj1
    Person obj2(obj1);
    // printing class data members for second object
    obj2.display();
    return 0;
   }
```

Output:

```
            Constructor for both name and age is called
            Name of current object: First person
```

```
Age of current object: 25

Copy constructor is called
Name of current object: First person
Age of current object: 25
```

In the above code, we have created a class and defined two types of constructors in it, the first is a parameterized constructor and another is a copy constructor. Parameterized constructor is used to create an object then by using the copy constructor we create a copy of it and stored it in another object.

8.4.4 Dynamic Constructor

When memory is allocated dynamically to the data members at the runtime using a new operator, the constructor is known as the dynamic constructor. This constructor is similar to the default or parameterized constructor; the only difference is it uses a new operator to allocate the memory.

Syntax of Dynamic Constructor:

```
class class_name
{
  private:
  // private members
  public:
  // declaring dynamic constructor
  class_name({parameters})
  {
    // constructor body where data members are initialized using new operator
  }
  };
```

Example of dynamic constructor:

```
#include <iostream>
using namespace std;
class Person
{
  // declaring private class data members
private:
  int* age;
public:
  Person(int* person_age)
  {
      cout<<"Constructor for age is called"<<endl;
      // allocating memory
      age = new int;
      age = person_age;
  }
  // display function to print the class data members value
  void display()
  {
      cout<<"Age of current object: "<<*age<<endl;
      cout<<endl;
  }
  };
int main()
  {
    // creating objects of class using parameterized constructor
    int age = 25;
    Person obj1(&age);
    // printing class data members for first object
    obj1.display();
    return 0;
   }
```

Output:

```
                        Constructor for age is called
                        Age of current object: 25
```

In the above code, we have created a class with a dynamic constructor. In the main function, we have created an object and initialized it using a dynamic constructor, where we have given memory dynamically using a new operator.

8.8 Destructor in C++

Destructor is just the opposite function of the constructor. A destructor is called by the compiler when the object is destroyed and its main function is to deallocate the memory of the object. The object may be destroyed when the program ends, or local objects of the function get out of scope when the function ends or in any other case. Destructor has the same as of the class with prefix tilde(~) operator and it cannot be overloaded as the constructor. Destructors take no argument and have no return type and return value.

Syntax of the Destructor:

```
class class_name
 {
 private:
 // private members
 public:
 // declaring destructor
 ~class_name()
 {
   // destructor body
  }
  };
```

In the above syntax, we can see the class has the name class_name and the destructor also has the same name, in addition there is a tilde(~). Also, there is no return type and return value of the destructor.

Important Points about the Destructor

- Destructor are the last member function called for an object and they are called by the compiler itself.
- If the destructor is not created by the user then compile creates or declares it by itself.
- A Destructor can be declared in any section of the class, as it is called by the compiler so nothing to worry about.
- As Destructor is the last function to be called, it should be better to declare it at the end of the class to increase the readability of the code.
- Destructor is just the opposite of the constructor as the constructor is called at the time of the creation of the object and allocates the memory to the object, on the other side the destructor is called at the time of the destruction of the object and deallocates the memory.

8.6 Difference between Constructor and Destructor in C++:

A constructor allows an object to initialize some of its value before it is used. A destructor allows an object to execute some code at the time of its destruction. The following example demonstrate the concept of Constructor and Destructor in C++.

Example/Implementation of Constructor and Destructor

```
#include <iostream>
using namespace std;
class Z
{
public:
    // constructor
    Z()
    {
        cout<<"Constructor called"<<endl;
    }
    // destructor
    ~Z()
    {
        cout<<"Destructor called"<<endl;
    }
    };
int main()
    {
    Z z1;   // Constructor Called
    int a = 1;
    if(a==1)
    {
        Z z2;  // Constructor Called
    }  // Destructor Called for z2
     } //  Destructor called for z1
```

Output:

```
                        Constructor called
                        Constructor called
                        Destructor called
                        Destructor called
```

The following table also differentiate the concept of Constructor and Destructor.

Sr. No.	Constructor	Destructor
1.	Constructor helps to initialize the object of a class.	Whereas destructor is used to destroy the instances.
2.	It is declared as **className(arguments if any){Constructor's Body }.**	Whereas it is declared as ~ **className(no arguments){ }.**
3.	Constructor can either accept arguments or not.	While it can't have any arguments.
4.	A constructor is called when an instance or object of a class is created.	It is called while object of the class is freed or deleted.
5.	Constructor is used to allocate the memory to an instance or object.	While it is used to deallocate the memory of an object of a class.
6.	Constructor can be overloaded.	While it can't be overloaded.
7.	The constructor's name is same as the class name.	Here, its name is also same as the class name preceded by the tiled (~) operator.

8.7 Inheritance in C++

The capability of a class to derive properties and characteristics from another class is called Inheritance. Inheritance is one of the most important features of Object-Oriented Programming. Inheritance is a feature or a process in which, new classes are created from the existing classes. The new class created is called "derived class" or "child class" and the existing class is known as the "base class" or "parent class". The derived class now is said to be inherited from the base class. When we say derived class inherits the base class, it means, the derived class inherits all the properties of the base class, without changing the properties of base class and may add new features to its own. These new features in the derived class will not affect the base class. The derived class is the specialized class for the base class.

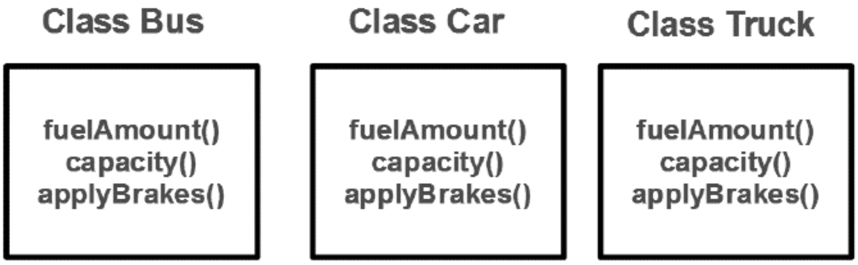
- **Sub Class**: The class that inherits properties from another class is called Subclass or Derived Class.

- **Super Class**: The class whose properties are inherited by a subclass is called Base Class or Superclass.

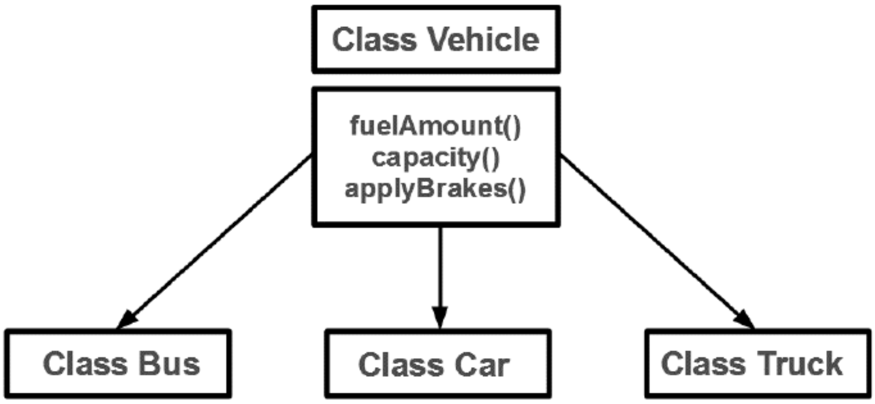
Reusability using Inheritance

C++ strongly supports the concept of reusability. The C++ classes can be reused in several ways. Once a class has been written and tested, it can be adapted by another programmer to suit their requirements. This is basically done by creating new classes, reusing the properties of the existing ones. The mechanism of deriving a new class from an old one is called inheritance. The old class is referred to as the base class and the new one is called the derived class or subclass. A derived class includes all features of the generic base class and then adds qualities specific to the derived class.

Example: Consider a group of vehicles. You need to create classes for Bus, Car, and Truck. The methods fuelAmount(), capacity(), applyBrakes() will be the same for all three classes. If we create these classes avoiding inheritance then we have to write all of these functions in each of the three classes as shown below figure:



The above process results in duplication of the same code 3 times. This increases the chances of error and data redundancy. To avoid this type of situation, inheritance is used. If we create a class Vehicle and write these three functions in it and inherit the rest of the classes from the vehicle class, then we can simply avoid the duplication of data and increase re-usability. Look at the below diagram in which the three classes are inherited from vehicle class:



Using inheritance, we have to write the functions only one time instead of three times as we have inherited the rest of the three classes from the base class (Vehicle).

Implementing inheritance in C++: For creating a sub-class that is inherited from the base class we have to follow the below syntax.

Syntax:

```
class  <derived_class_name> : <access-specifier> <base_class_name>
      {
      //body
       }
```

Note: A derived class doesn’t inherit access to private data members. However, it does inherit a full parent object, which contains any private members which that class declares.

Example of Inheritance:

```
class ABC : private XYZ            //private derivation
{                            }
class ABC : public XYZ            //public derivation
{                            }
class ABC : protected XYZ        //protected derivation
{                            }
class ABC: XYZ                     //private derivation by default
{                            }
```

8.8 Modes of Inheritance

There are three modes of inheritance that is publicly, privately, and protected. If we are not writing any access specifiers then by default it becomes private.

- **Public Mode**: If we derive a subclass from a public base class. Then the public member of the base class will become public in the derived class and protected members of the base class will become protected in the derived class.
- **Protected Mode**: If we derive a subclass from a Protected base class. Then both public members and protected members of the base class will become protected in the derived class.
- **Private Mode**: If we derive a subclass from a Private base class. Then both public members and protected members of the base class will become Private in the derived class.

Note: The private members in the base class cannot be directly accessed in the derived class, while protected members can be directly accessed. For example, Classes B, C, and D all contain the variables x, y, and z in the below example. It is just a question of access.

```
// C++ Implementation to show that a derived class
// doesn’t inherit access to private data members.
// However, it does inherit a full parent object.
class A
{
public:
    int x;
protected:
    int y;
private:
    int z;
};
class B : public A
{
    // x is public
    // y is protected
    // z is not accessible from B
};
class C : protected A
{
    // x is protected
    // y is protected
    // z is not accessible from C
};
class D : private A // 'private' is default for classes
{
    // x is private
    // y is private
    // z is not accessible from D
};
```

The below table summarizes the above three modes and shows the access specifier of the members of the base class in the subclass when derived in public, protected and private modes:

Base Class Member Access Specifier	Mode of Inheritance		
	Public	Protected	Private
Public	Public	Protected	Private
Protected	Protected	Protected	Private
Private	Not accessible (Hidden)	Not accessible (Hidden)	Not accessible (Hidden)

8.9 Types of Inheritance

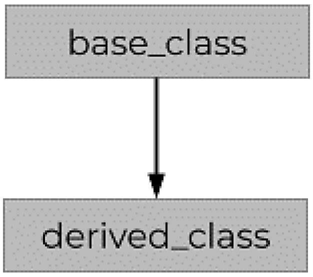
During inheritance, the data members of the base class get copied in the derived class and can be accessed depending upon the visibility mode used. The order of the accessibility is always in a decreasing order i.e., from public to protected. C++ supports five types of inheritance:

- Single inheritance
- Multiple inheritance
- Multilevel inheritance
- Hierarchical inheritance
- Hybrid inheritance

The detail of each type is given below:

8.9.1 Single Inheritance

Single Inheritance is the most primitive among all the types of inheritance in C++. In this inheritance, a single class inherits the properties of a base class. All the data members of the base class are accessed by the derived class according to the visibility mode (i.e., private, protected, and public) that is specified during the inheritance.



Syntax of Single Inheritance is given below:

```
class base_class_1
{
   // class definition
};
class derived_class: visibility_mode base_class_1
{
   // class definition
};
```

Description: A single derived_class inherits a single base_class. The visibility_mode is specified while declaring the derived class to specify the control of base class members within the derived class.

Given below is a complete Example of Single Inheritance.

```
#include <iostream>
#include <string>
using namespace std;
class Animal
{
  string name="";
  public:
  int tail=1;
  int legs=4;
};
class Dog : public Animal
{
  public:
  void voiceAction()
  {
    cout<<"Barks!!!";
  }
  };
int main()
  {
  Dog dog;
  cout<<"Dog has "<<dog.legs<<" legs"<<endl;
  cout<<"Dog has "<<dog.tail<<" tail"<<endl;
  cout<<"Dog ";
  dog.voiceAction();
}
```

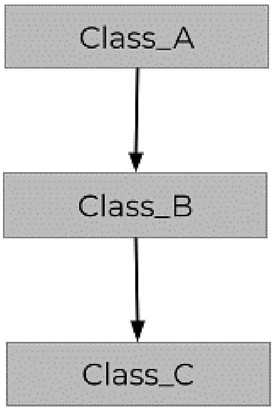
Output:

```
        Dog has 4 legs
        Dog has 1 tail
        Dog Barks!!!
```

Explanation: We have a class Animal as a base class from which we have derived a subclass dog. Class dog inherits all the members of the Animal class and can be extended to include its own properties, as seen from the output.

8.9.2 Multilevel Inheritance

The inheritance in which a class can be derived from another derived class is known as Multilevel Inheritance. Suppose there are three classes A, B, and C. A is the base class that derives from class B. So, B is the derived class of A. Now, C is the class that is derived from class B. This makes class B, the base class for class C but is the derived class of class A. This scenario is known as the Multilevel Inheritance. The data members of each respective base class are accessed by their respective derived classes according to the specified visibility modes.



Syntax of Multilevel Inheritance is given below:

```
class class_A
{
   // class definition
};
```

```
class class_B: visibility_mode class_A
{
   // class definition
};
class class_C: visibility_mode class_B
{
   // class definition
};
```

Description: The class_A is inherited by the sub-class class_B. The class_B is inherited by the subclass class_C. A subclass inherits a single class in each succeeding level.

Example of Multilevel Inheritance is given below:

```
#include <iostream>
#include <string>
using namespace std;
class Animal
{
  string name="";
  public:
  int tail=1;
  int legs=4;

};
class Dog : public Animal
{
  public:
  void voiceAction()
  {
    cout<<"Barks!!!";
  }
};
class Puppy:public Dog{
  public:
  void weeping()
  {
    cout<<"Weeps!!";
  }
};
int main()
{
  Puppy puppy;
cout<<"Puppy has "<<puppy.legs<<" legs"<<endl;
cout<<"Puppy has "<<puppy.tail<<" tail"<<endl;
cout<<"Puppy ";
puppy.voiceAction();
cout<<" Puppy ";
puppy.weeping();
}
```

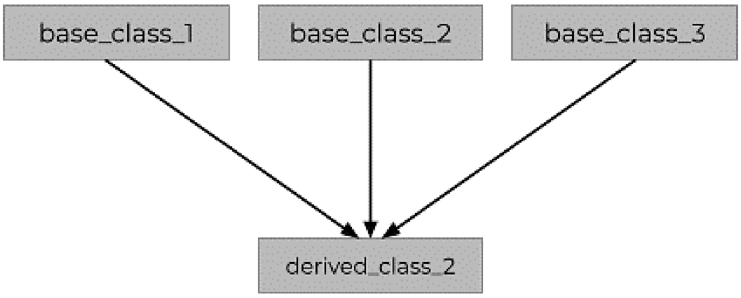
Output:

```
Puppy has 4 legs
Puppy has 1 tail
Puppy Barks!!! Puppy Weeps!!
```

Here we modified the example for Single inheritance such that there is a new class Puppy which inherits from the class Dog that in turn inherits from class Animal. We see that the class Puppy acquires and uses the properties and methods of both the classes above it.

8.9.3 Multiple Inheritance

The inheritance in which a class can inherit or derive the characteristics of multiple classes, or a derived class can have over one base class, is known as Multiple Inheritance. It specifies access specifiers separately for all the base classes at the time of inheritance. The derived class can derive the joint features of all these classes and the data members of all the base classes are accessed by the derived or child class according to the access specifiers.



Syntax of Multiple Inheritance is given below:

```
class base_class_1
{
   // class definition
};
class base_class_2
{
   // class definition
};
class derived_class: visibility_mode_1 base_class_1, visibility_mode_2 base_class_2
{
   // class definition
};
```

Description: The derived_class inherits the characteristics of two base classes, base_class_1 and base_class_2. The visibility_mode is specified for each base class while declaring a derived class. These modes can be different for every base class.

Example of Multiple Inheritance is given below:

```
#include <iostream>
using namespace std;
//multiple inheritance example
class student_marks {
protected:
int rollNo, marks1, marks2;
public:
void get()
{
cout << "Enter the Roll No.: "; cin >> rollNo;
cout << "Enter the two highest marks: "; cin >> marks1 >> marks2;
   }
};
class cocurricular_marks
{
protected:
int comarks;
public:
void getsm() {
cout << "Enter the mark for CoCurricular Activities: "; cin >> comarks;
}
};
//Result is a combination of subject_marks and cocurricular activities marks
class Result : public student_marks, public cocurricular_marks
{
  int total_marks, avg_marks;
  public:
  void display()
{
    total_marks = (marks1 + marks2 + comarks);
    avg_marks = total_marks / 3;
    cout << "\nRoll No: " << rollNo << "\nTotal marks: " << total_marks;
    cout << "\nAverage marks: " << avg_marks;
  }
};
int main()
{
  Result res;
res.get(); //read subject marks
res.getsm(); //read cocurricular activities marks
res.display(); //display the total marks and average marks
}
```

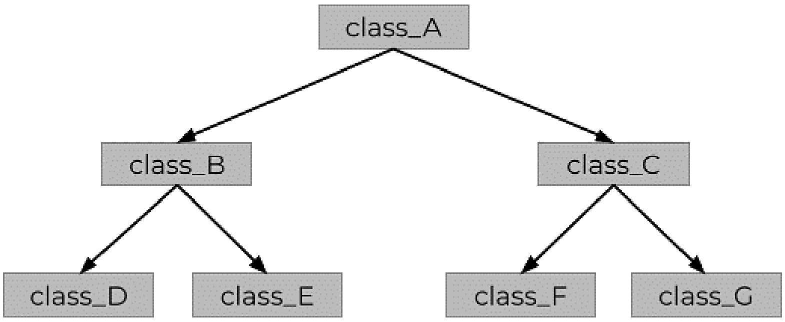
Output:

```
Enter the Roll No.: 25
Enter the two highest marks: 40 50
Enter the mark for CoCurricular Activities: 30

Roll No: 25
Total marks: 120
Average marks: 40
```

Explanation: In the above example, we have three classes i.e. student_marks, cocurricular_marks, and Result. The class student_marks reads the subject mark for the student. The class cocurricular_marks reads the student's marks in co-curricular activities.

8.9.4 Hierarchical Inheritance

The inheritance in which a single base class inherits multiple derived classes is known as the Hierarchical Inheritance. This inheritance has a tree-like structure since every class acts as a base class for one or more child classes. The visibility mode for each derived class is specified separately during the inheritance and it accesses the data members accordingly.



Syntax of Hierarchical Inheritance is given below:

```
class class_A
{
   // class definition
};
class class_B: visibility_mode class_A
{
   // class definition
};
class class_C : visibility_mode class_A
{
   // class definition
};
class class_D: visibility_mode class_B
{
   // class definition
};
```

```
class class_E: visibility_mode class_C
{
   // class definition
};
```

Description: The subclasses class_B and class_C inherit the attributes of the base class class_A. Further, these two subclasses are inherited by other subclasses class_D and class_E respectively.

Example of Hierarchical Inheritance is given below:

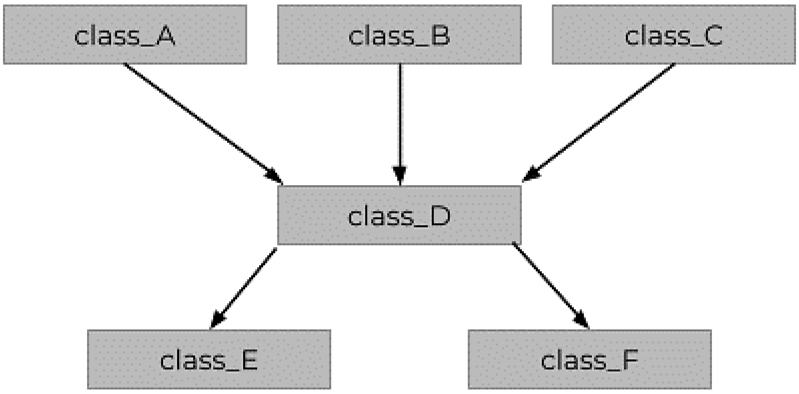
```
// C++ program to implement
// Hierarchical Inheritance
#include <iostream>
using namespace std;
// base class
class Vehicle {
public:
   Vehicle() { cout << "This is a Vehicle\n"; }
};
// first sub class
class Car : public Vehicle {
};
// second sub class
class Bus : public Vehicle {
};
// main function
int main()
{
   // Creating object of sub class will
   // invoke the constructor of base class.
   Car obj1;
   Bus obj2;
   return 0;
}
```

Output:

```
This is a Vehicle
This is a Vehicle
```

8.9.5 Hybrid Inheritance

Hybrid Inheritance, as the name suggests, is the combination of two or over two types of inheritances. For example, the classes in a program are in such an arrangement that they show both single inheritance and hierarchical inheritance at the same time. Such an arrangement is known as the Hybrid Inheritance. This is arguably the most complex inheritance among all the types of inheritance in C++. The data members of the base class will be accessed according to the specified visibility mode.



Syntax of Hybrid Inheritance is given Below:

```
class class_A
{
   // class definition
};
class class_B
{
   // class definition
};
class class_C: visibility_mode class_A, visibility_mode class_B
{
   // class definition
};
class class_D: visibility_mode class_C
{
   // class definition
};
class class_E: visibility_mode class_C
{
   // class definition
};
```

Description: The derived class class_C inherits two base classes that are, class_A and class_B. This is the structure of Multiple Inheritance. And two subclasses class_D and class_E, further inherit class_C. This is the structure of Hierarchical Inheritance. The overall structure of Hybrid Inheritance includes more than one type of inheritance.

Example of Hybrid Inheritance is given below:

```
// C++ program for Hybrid Inheritance
#include <iostream>
using namespace std;
// base class
class Vehicle {
public:
   Vehicle() { cout << "This is a Vehicle\n"; }
};
// base class
class Fare {
public:
   Fare() { cout << "Fare of Vehicle\n"; }
};
// first sub class
class Car : public Vehicle {
};
// second sub class
class Bus : public Vehicle, public Fare {
};
// main function
int main()
{
   // Creating object of sub class will
```

```
    // invoke the constructor of base class.
    Bus obj2;
    return 0;
}
```

Output:

```
                        This is a Vehicle
                        Fare of Vehicle
```

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University

(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-I
Course: DATA BASE MANAGEMENT SYSTEM (MSCS-1-02T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

M.Sc. (Computer Science) Programme Coordinator:
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:
Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Business Management & Commerce:
Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Social Sciences and Liberal Arts:
Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA
A State University Established by Govt. of Punjab vide Act No. 19
of 2019 and Approved Under section 2(f) of UGC

PROGRAMME COORDINATOR
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

COURSE COORDINATOR & CONTENT WRITER:
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Course: DBMS	
Course Code: MSCS-1-02T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Understand the fundamental elements of database management system.
CO2	Understands the three level architecture of DBMS and mapping between these levels.
CO3	Familiar with the hierarchical model, network model, entity relationship model and relational model.
CO4	Acquire knowledge of normalization technique that reduces data redundancy and eliminates undesirable characteristics like Insertion, Update and Deletion Anomalies.
CO5	Apply SQL and PL/SQL to solve problems

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

Name of Programme: M.Sc. (Computer Science)
Name of Course: DBMS
Course Code: MSCS-1-02T
Semester: 1st Semester

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

SECTION A

Unit I: Introduction of DBMS: Database Approach, Characteristics of a Database Approach, Database System Environment. Roles in Database Environment: Database Administrators, Database Designers, End Users, Application Developers. Database Management Systems: Definition, Characteristics, Advantages of Using DBMS Approach, Classification of DBMSs.

Unit II: Three Level Architecture of DBMS: Database Schema and Database Instance, Mapping Between Different Views, Data Independence–Physical and Logical data Independence, Difference between logical data independence and physical data independence, Components of a DBMS, Data Dictionary, DBMS Languages

Unit III: Data Models: Classification of Data Model, Hierarchical Model, Network Model, Entity Relationship Model, Database Conceptual Modeling by E-R model: Concepts, Entities and Entity Sets, Attributes, Mapping Constraints, E-R Diagram, Weak Entity Sets, Strong Entity Sets, Comparison between Data Models.

Unit IV: Relational Data Model: Concepts and Terminology. Constraints: Integrity Constraints, Entity and Referential Integrity constraints, Keys: Super Keys, Candidate Keys, Primary Keys, Secondary Keys and Foreign Keys. Relational Algebra: Basic Operators, Additional Operators. Relational Calculus: Tuple Relational Calculus and Domain Relational Calculus, Difference between relational algebra and relational calculus.

SECTION B

Unit V: Normalization: Functional Dependency, Full Functional Dependency, Partial Dependency, Transitive Dependency, Normal Forms– 1NF, 2NF, 3NF, BCNF, Multi-valued Dependency, Join Dependency and Higher Normal Forms-4NF, 5NF.

Unit VI: Transaction Management and Concurrency Control: ACID Properties. Database Protection: Security Issues, Discretionary Access Control-Granting and Revoking Privileges. Database Concurrency: Problems of Concurrent Databases, Serializability and Recoverability, Concurrency Control Methods-Two Phase Locking, Time Stamping. Deadlock, Database security and integrity, Different Methods of Database Security, Database Recovery: Recovery Concepts, Recovery Techniques-Deferred Update, Immediate Update, Shadow Paging.

Unit VII: Introduction to SQL: Introduction to SQL*PLUS, Data types, Parts of SQL: Data Definition Language, Data Manipulation Language, Data Control Language, and Transaction Control Language. SQL Operators, SQL Functions, Joins, Roll up operation, Cube operation, Nested query, Subquery, View, Disadvantages of SQL

Unit VIII: Introduction to PL/SQL: Difference between SQL and PL/SQL, Block structure of PL/SQL, Architecture of PL/SQL, Elements and Data Types of PL/SQL, Variables and Constants of PL/SQL, Control structures of PL/SQL, Cursors, Exception handling, Subprograms, Stored packages, Triggers

Reference Books:

- Elmasry Navathe,“Fundamentals of Database System”,Pearson Education.
- Oracle SQL Complete Reference”,Tata McGraw-Hill.
- T.Connolly,CBegg,“Database Systems”,Pearson Education.
- Jeffrey D.Ullman,"Principles of Database Systems",Galgotia Publications.
- Henry F.Korth,A.Silberschhatz,“Database Concepts,"Tata McGraw Hill.
- C.J.Date,"An Introduction to Database Systems”,Pearson Education.

DBMS (CREDITS: 2)

Total Marks: 100

External Marks: 70

Internal Marks: 30

Credits:2

Pass Percentage: 40%

The programs in lab will be based on the contents covered in the theory syllab

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 1: INTRODUCTION OF DATABASE MANAGEMENT SYSTEM

1. INTRODUCTION

The exponential growth of information technology and its dependency in different sectors of society results in collection of hugs data. The large data collection has to be stored in such a way that is should be retrieved and processed as per the requirement of the user. Traditionally, data was manually maintained, stored in fields, updated and retried manually. The system was worked with very small amount of data which was isolate handled by single user. With the increase in size of data and access of multiple users for single source of data, manually management of such data was nearly impossible in such a scenario, the concept of database management system was originated. The goal of the database management system is to store information in such a way so that it can be access with ease. The database management system is aimed to perform basic operation like: storing, retrieval, sorting, searching, and deletion of records in database. It plays a critical role in almost all area where computer systems are used for information processing like business, engineering, medical, defence, education, library etc. The database sheared among different users. Sometime, it is called mediator between user and data as it communicate between user and data. It responses to the user with results after processing query raised by the user. Before going into more technical detail of DBMS, let us ho through the basic concepts:

1.1 DATABASE CONCEPTS

The conceptual understating of database is required to go through two elementary database concepts: Data and Information

I. ***Data: It is defined as representation of facts, concepts, and instruction in a from which is suitable for communication, interpretation by human or computer.***

- Data can be recorded and have meaning.
- Examples of data are: height, weights, prices, costs, names of things, marks, image, and sound.
- In a formal manner, data is suitable for understanding and processing's.

Data can be represented with different character set and format which are stipulated in table given below:

Representation of Data		
Sr. No.	**Format**	**Character set**
1	Alphabet	(A-Z a-z)
2	Digits	(0-9)
3	Special characters	(+,-,*,/,@,#)

4	picture	Picture in jpeg, Gif, Tiff, etc format
5	Sound	Sound in mp3, mp4 format

Table 1.1 Representations of Data (Format Character Set)

In a any information processing system like database management system, data us considered as raw material or figure as it itself is not significant. It requires to be processed to come up with suitable fact and figure which is called as information.

II. ***Information: It is defined as processed from of data which has significance in decision making or performing some action. In another words, information is data that has been converted into some useful form.***

The information is a result of processing of data according to specific requirement. In hugs data collection user is asked to make query to fetch required information. The following logical diagram demonstrated the same concept:

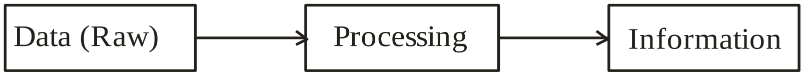


Fig. 1.1: Data Processing for Information

It is very important to throw light on the basic difference between these two elements: Data and information, as these terms are quite miss used among beginners. They sometime us information on the place of data but both information and data are different from each other. The tabular representations of fact as shown below very well explained the differences.

Sr. No	Data	Information
1	Data is raw facts and unorganized figure that need to be processed.	Information is processed from of data which may be further processed to gain knowledge.
2	Data is useless until it is organized and does not convey any message.	Information is useful and conveys meaningful message.

3	Data is used as input for any data processing application.	Information is the result of any data processing application.
4	Decision making is not recommended on data as data may or may not be meaningful .That is why: data not help in decision making.	Information is always meaningful and plays vital role in any decision making process.
5	Data is available in unorganised and un specific format.	Information is always required in organised and in specific format.
6	Data is a collection of atomic levels of pieces. It collectively represents different fact and figure.	Information is organized collection of data and is always represent about specific entity.
7	Data itself has no significance in business as data is not in the form that can be interrelated.	Information is interrelated to data and has strong significance in business.
8	Data representation order is not significant as it may be in any order. It has no effect of meaning	Information must be in specific order otherwise It may have different meaning.
9	The data cannot be interpreted as it is very difficult to understand. It may have different meaning for different person in different situation.	Information is concrete in nature and easy to Understand. It has same meaning for everyone in any situation.
10	Example: Data may have figure like 20, 30, 50, 70 that it is a raw figure which has no significance.	Example: Data is proceed and associate with some meaningful facts like 20 year old,30 kg weight,50 gm Gold, 70 km/hr, etc May be processed from of data. Now

		these figures have some meaning.

Table 1.2: Different between Data and Information

Importance of information in organization: the organisation has maintained data of activities conducted during the session which include sale purchase data, human resource data, store data, etc as shown below in logical diagram.

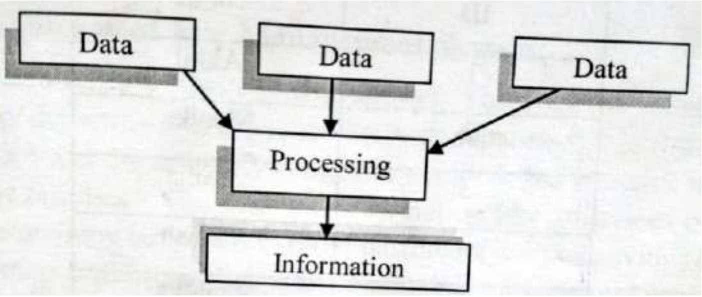


Fig. 1.2: Organisational Data Processing Conceptual Model

Every organisation has data processing systems that applied on different data to fetch information for smooth working of their organization. The information may be significant to the organisation for many purpose some of these are listed below which emphasis that information is very important for smooth working of any organisation.

Based on the above points, we conclude that the information help in planning, the action in the process of running and protecting the system.

- To gain information about the organisational sale and purchase.
- To access information of employee in the organisation.
- To know about the future predictions of the organisation.
- To know about the surroundings and whatever is happening in the society and universe.
- To keep the system up to data.

1.2 TRADITIONAL FILE MANAGEMENT SYSTEM

The file management system is a traditional approach to store and mange data in files. it is early day approach when records are stored in different files with different format. Each department in organisation have own file storage system where specific application are designed to process these applications. The department have their own set rule to store and retrieve data from file system. The system to handle these file was called file management system. Such system are file department and incompatible to other file system. It means file system of one department may not work for file processing of another department. But such system is preferably good as compare to manual file management.

Key points:

- File processing system is a simple computer file system.
- It is a group of files storing data of an organization.
- File are in the form of text. Even records are also in text form.
- Each file is independent from one another.
- Each file is called a *falt file.*
- It uses hard disk or CD to store the data.
- File are designed by using programs written in programs written in programming languages such as C, C++.
- Searching is very difficult. Searching will start and continue till find the result.
- If data is very large then searching will take long time.

ID	**Name**
1	Akhil
2	Monika
3	Aastha
4	Ankush
5	Radhika

Formula for searching = n+1/2

- File are suitable when number of store items is small.
- It is not suitable when we have to perform data processing.
- As a system became more complex, file processing system presented many limitations and were difficult to maintain.

Example : To understand file management system in details we are taking practical example. Let us consider a business organisation where different department are organised and performed different tasks. Suppose department are Sale and Purchase Department, Inventory, Control Department, HR Department and Production unit. Every department have own system to store information in files. The following diagram shows how file management system works and manages data is different files.

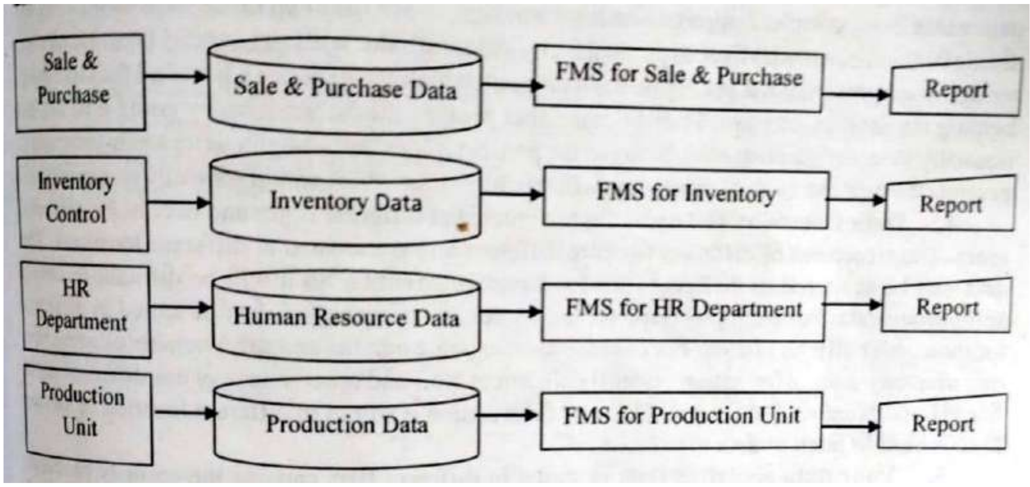


Fig. 1.3: File Management system in a business Organisation

It is very much clear form the above diagram that in file management system every department has separate storage of data and specific application for processing. In such as system inter-departmental access is not possible and there is duplication of organisational data. So the file management system has many limitations which are addressed in new concept, called, database management system.

Limitation/ Disadvantages of file Management System:

1. **Duplication of Data (Data Redundancy):** The file are created according to the application and every department in organistion have separate file system. So in that case the repetition of information about an entity cannot be avoided. For instance in Bank, the files are maintained about the customer. The personal information like addresses of customer holdings savings account and also the address od the customers will be present in file maintaining the current account. Even in case if same customers

have a saving account and current account his address will be present at two places. There is duplication of data as files are not shear able among different applications.

2. **Data Inconsistency and Inflexibility:** Data isolation limited the flexibility of file processing system in providing users with ad-hoc information requests. Data inconsistency means data about same entity stored in different files are not up-to-date and is not identical at same time. It is due to duplication of data which leads to greater problem than just wasting the storage. Same data which has been repeated at several places may not match after it has been updated at some places. For example: Suppose the customer requests to change the address for his account in the Bank and the Program is executed to update the saving bank account file only but his current bank account file is not updated. Now the addresses of the same customer have two addresses stored in two different locations that are called data inconsistency.

3. **Difficulty in Accessing Data:** In file management system, the program is designed for generating ad hoc reports. It means that program is for not general purpose and is data dependent. For example: Suppose administrator want to see list of all the customers holding the saving banks account who lives in particular city. Administrator will not have any program already written to generate that list but say he has a program which can generate a list of all the customers holding the savings account. Then he can either provide the information by going thru the list manually to select the customers living in the particular locality or he can write a new program to generate the new list. Both of these ways will take large time which would generally be impractical.

4. **Data Isolation:** The data files are created at different times and may be by different users. The structures of different files are different and are located at different locations. The data will be scattered in different files for a particular entity. So it will be difficult to obtain appropriate data. For example: Suppose the address of an employee may be stored in different location under different fields. For instance to store house number and street number of employee, one user may store information under (HNo, Street No.) and other one may use different name like (House Number and Street). This

way information is stored in different location of similar kind is hard to fetch as data is isolated.

5. **Poor data security:** Data is stored in different files causing the security problem. The data should be protected from unauthorized users. Every user should not be allowed to access every data.

6. **Difficult to Show Data According to User:** In file processing system, it was difficult to determine relationships between isolated data in order to meet user requirements.

7. **Concurrency Problems:** When more than one user are allowed to process the database. If in that environment two or more users try to update a shared data element at about the same time then it may result into inconsistent data. In case of file management system such concurrent access to data is hard to implement.

8. **Data in separated files:** Data is in more than one file and it is difficult to take data from more than one files.

9. **Data Dependence:** Data dependence means it is impossible to change storage structure without affecting the application program. If the format of a certain record was changes, the code in each file containing that format must be updated.

10. **Incompatible File Formats:** Each programmer stores the data in the file in the format as per the choice as there is no standard file format for storing the file. It becomes very difficult to handle the different files in different format.

The database management systems are designed to overcome above listed problems along with other advanced database concepts. The following section is designed to address issues like Database, DBMS, Difference between Database and DBMS, Characteristics of DBMS, Functions of DBMS, Advantages and Disadvantages of DBMS.

1.3 DATABASE

Database: It is a computer based record keeping system whose over all purpose is to record and maintains data. It is designed to hold bundle of organizational data. It holds the records, fields, and cells of data.

The database stores the known facts that can be recorded and that have implicit meaning. Data is represented in database in different levels of abstraction in its architecture.

Typically, there are three levels: External, Conceptual, and internal. The following diagram show how data is represented in different levels:

- **External Level:** It defines how user views data. Single user may have multiple views.
- **Conceptual Level:** It is a communication medium between external and internal level. Its representation is unique regardless of external level and internal level.
- **Internal Level:** It defines how data is physically stored.

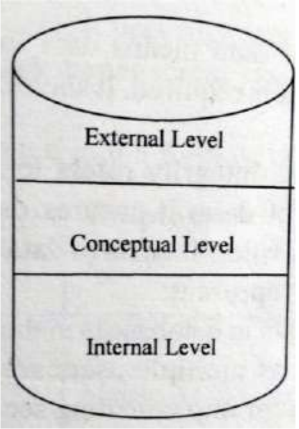


Fig. 1.4: Database Abstraction Levels

Key Points:

- The database is a shared collection of logically related data, designed to meet the information needs of an organization.
- It is a computer based record keeping system. Its overall purpose is to record and maintain the information.
- The database is a single large repository of data which can be used simultaneously by many departments and users.
- It holds not only the organisation's operational data but also a description of the data. It is also defined as a self-describing collection of integrated records.

- The description of the data is known as the Data Dictionary or Meta Data (the 'data about data").
- We can perform many operations on database such as:
  a) To add new operation.
  b) To modify/ edit the existing information.
  c) To remove/delete the unwanted information.
  d) To retrieve/ view the stored information.
  e) Arrange the information in a desired manner.
- Database is manages by an individual or group called Database Administrator (DBA), Who is responsible for designing, creating and maintaining the database to satisfy the needs of the users.
- All access to database is automated by special software called Database Management system (DBMS).
- The tern database is generally confused with DBMS. The database is a concept to representation data whereas DBMS an application program to provide access on database Both together represent Database System.

Characteristics of Database:

Database has some Characteristics in order to meet the standards which are as follows:

1. **Data sharing:** Database should ne capable to be shared among different users and applications.
2. **Persistence:** Persistence of data means data in a database exist permanently and available in time whenever it is required. It should live beyond the scope of the process that created it.
3. **Integrity/Correctness**: Data integrity refers to the property of data which enforce constrains to safe format of data. It ensures data should in a uniform format and implemented with integrity rules. It ensures data should be correct with respect to the real world entity that they represent.

4. **Security:** The security of data in database is in the top of priority. It should be protected from unauthorized access as multiple users are sharing database. Database should have their own mechanism for implementing security.
5. **Consistency:** The consistence of data is must whenever more than one data element in a database represents related real world values. The values should be consistent with respect to the relationship.
6. **Non-Redundancy**: The data in database should not be duplicated as no two data items in a database should represent the same real world entity. The non-redundancy helps to reduce size of the database and avoid inconsistency of data.
7. **Independence:** The database has three different levels (External, Conceptual, Internal) to represent data. These levels should be independent of each other so that the changes in one level should not affect the other levels.

1.4 DATABASE MANAGEMENT SYSTEM (DBMS)

DBMS: A database management system is a collection of interrelated data and a set of programs to access those data. The interrelated data is called database which is a shared collection of logically related data, designed to meet the information needs of an Organisation.

The primary goal of a DBMS is to provide methods to store and retrieve database information that is both convenient and efficient. Database sustems are designed to manage large bodies of information. In addition, the database system must ensure the safety of the information stored. Despite system crashes or attempts at unauthorized access. If data is to be shared among several users, the system must avoid possible anomalous results.

Key Points:

- DBMS is a software system that allows user to create , maintain and delete a database. It provides controlled access to the data.
- It centralzed the database.
- It is a computerized system which maintains the data.

- DBMS is an intermediate between programs and data. It is used to make information from data.
- DBMS is a collection of programs which are required to perform different task on database. It perform various operation on data like defining structure of data, accepting data, format data as per user requirement, hide data, allow concurrent access, backup and provide security to data.
- DBMS ensures the privacy of data. It prevent data from unauthorized users.
- Commercially available database management systems in the market are dbase, Foxpro, Oracle etc.
- In DBMS, data can be represented in the dorm of tables.

Employee

Ename	Empno	Job	Sal	Deptno
Nidhi	6258	Clerk	900	20
Aastha	6388	Manager	1500	30
Manmeet	6765	Clerk	1050	10
Navreet	6800	Analyst	1100	30

Entity : Employee

Attributes : Ename, Empno, Job, Sal, Deptno

Record : collection of related data i.e. Ename, Empno, Job, Sal, deptno

- In DBMS, each user can view data according to his/ her choice. Two users can use the same portion of data at the same time different forms.
- DBMS is used to create the reports and matahematical functions for the users.

1.4.1 Characteristics of Database Management system

A database management system is designed to define, manipulate, retrieve and manage data in a database. It generally manipulates the data itself, the data format, filed names, record structure and file structure. It also defines rules to validate and manipulate this data. The modern DBMS has the following characteristics:

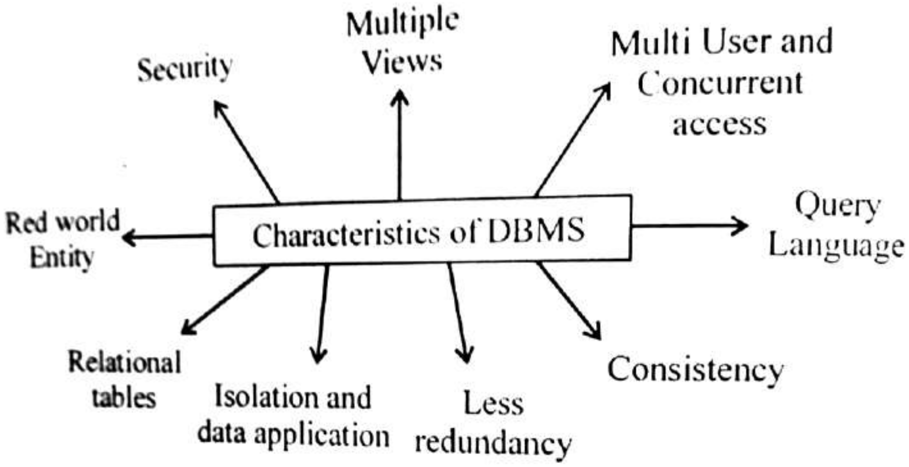
1. **Real world entity:** The DBMS is designed to represent real world entities consist of feature and behaviour of real world object. The DBMS have constructs that can easily define real world entity.

2. **Relational tables:** The database contains tables which are mapped with entities. These entity tables are related with other tables to define relational. This eases the concept of data saving. A user can understand the architecture of database just by looking at table names.

3. **Isolation and data application:** The DBMS is designed to isolate data form other complication working of the system as data in preserve into database. The application programs are written to access data

4. **Less redundancy :** DBMS Follows rules of normalization, which splits a relation when any of its attributes is having redundancy in values. Following normalization, which itself is a mathematically rich and scientific process, make the entire database to contain as less redundancy as possible.

5. **Consistency:** DBMS always enjoy the state of consistency where the previous from of data storing application like file processing does not guarantee this. Consistency is a state where every relation database remains consistent. There exist methods and techniques, which can detect attempt of leaving database in inconsistent state.



6. **Query language:** DBMS is equipped with query language, which makes it more efficient to retrieve and manipulate data. A user can apply as many any different filtering options, as her on she wants. Traditionally it was not possible where file-processing system was used.

7. **Multiuser and concurrent access**: DBMS support multi-user environment and allows them to access and manipulate data in parallel. Though there are restriction on transactions when they attempt to handle same data item, but users are always unaware of them.

8. **Multiple views**: DBMS offers multiples view for different users. A user who is in sales department will have a different view of database than a person working in production department. The enables user to have a concentrate view of database according to their requirements.

9. **Security**: Features like multiple views offers security at some extent where users are unable to access data of other user and department. DBMS offers methods to impose constraints while entering data into database and retrieving data at later stage. DBMS offers many different levels of security features, which enables multiple users to have different view with different features, for example, a user in sales department cannot see data of purchase department is one thing, additionally how much data of sales department he can see, can also be managed. Because DBMS offer many different features, for examples, a user in sales department cannot see data of purchase department is one thing, additionally how much data of sales department he can see, can also be managed. Because DBMS is not saved on disk as traditional file system it is very hard for a theif to break the code.

1.4.2 Operations/Functions of Database Management System

A DBMS is an intermediate between user and database. The DBMS provides multiple useful interface to interact with database. There are several function that a DBMS performs to ensure data integrity and consistency of data in the database. The following are some important function of data management system.

1. **Data Dictionary Management :** Data Dictionary is where the DBMS stores definitions of the data elements and their relationships i.e. metadata. It is often hidden from the user and is used by Database Administrators and Programmers. It also shows which program use which piece of database and record.

   The DBMS uses different function to look up the required data along with relationships into data Dictionary. Whenever a request is made for a particular data in database then DBMS programs access data dictionary. The function removes structural and data dependency and provides the user with data abstraction.

2. **Data Storage Management** : The data storage management is one of core function of DBMS which is used for the storage of data and any related data entry forms, report definitions, data validation rules, procedural code, screen definition and structures. The DBMS manage data in such a way that users do not need to know how data is stored or manipulated.

3. **Data Transformation and Presentation :** The data transformation and presentation is one of integrated function of DBMS as it is helpful to store data in simple format and display information in uniform format. The function exits to transform any data entered into required data structures. By using the data transformation and presentation function, the DBMS can determine the difference between logical and physical data formats.

4. **Security Management:** The security management is implemented in DBMS at different levels. Security management sets rules that determine specific user that are allowed to access the database. Users are given a username and password or sometimes through biometric authentication. DBMS must monitor user request. It can reject the request which break the security rules.

5. **Data Definition:** DBMS must be able to accept data definition commands. These commands are: create Alter and Drop.

6. **Data Manipulation**: DBMS must be able to handle the request from user to retrieve update and delete data. These commands are: Select, insert, Update and Delete.

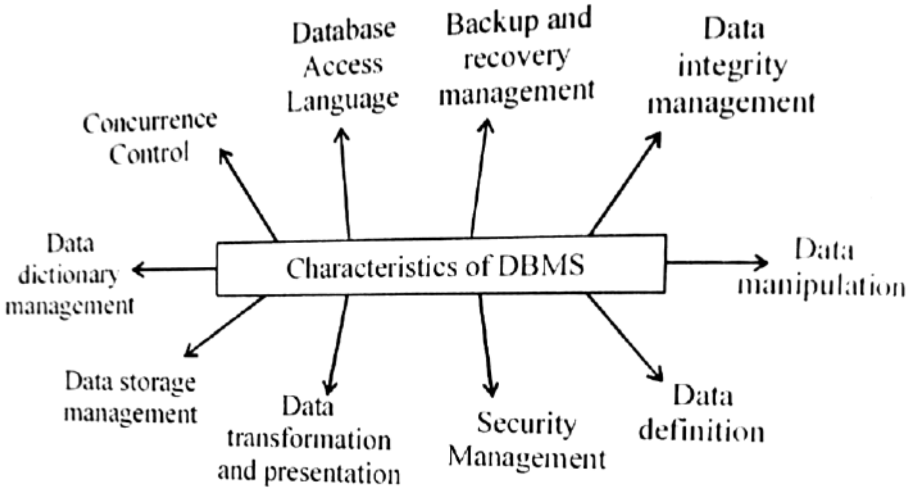


Fig. 1.6: Operations/Function of Database Management System

7. **Data Integrity Management:** Data integrity and data consistency are the core function of DBMS to provide security to data. The DBMS enforces these simultaneously without affecting the integrity of the database. The DBMS enforces these rules to reduce things such as data redundancy, which is when data is stored in more than one place unnecessarily, and maximizing data consistency, making sure database is returning correct/same answer each time for same question asked.

8. **Backup and Recovery Management:** Backup and recovery is done by DBMS to safeguard the old data so that unwanted damage to data can be recovered. Backup management refers to the data safety and integrity; for example backing up

document files. Similarly recovery of data can be implemented to go back to check the previous status of the data. DBMS software component " transaction manager" is used to recover the data which was lost due to some misshapen.

9. Database Access Languages and User Interfaces : The DBMS provides multiple user interfaces to meet different requirements of the end user in different network environments DBMS may provides different terminals, web interfaces, etc. DBMS also provides user interface to interact with database. It is not feasible to provide everything in the form of drop down menu so DBMS supports a SQL (structured query language) language which is a non procedural language. The use of SQL language makes it easy for user to seek information according to the requirement. User can seek information by providing command to DBMS query processor which arranges data to the user.

10. Concurrency control : since DBMS support sharing of data among multiple users they must provide a mechanism for managing concurrent access to the database. DBMS ensure that the database kept in accurate state.

1.4.3 Advantages of Database Management System

1. Minimal Redundancy/Eliminate Duplication: In non-database system each application program has its own private files. In this case, the duplicated copies of the same data are created in many places. In DBMS, all data of an organization is integrated into a single database file. The data is recorded in only one place in the database and it is not duplicated. Centralized control of unnecessary duplication of data. It also reduce the total amount of data storage. It also eliminates the extra processing required to trace the results.

2. **Data Integrity:** In database management system, data in database is stored in tables. A single database contains multiple tables and relationships can be created between tables (or associated data entities). This makes easy to retrieve and update data. Data store in database is accurate and consistent. Integrity of data means that data in

database is always accurate, such that incorrect information cannot be stored in database. If the system "crashes", we can retrieve the data easily

3. **Improved Data Consistency:** By controlling the data redundancy, the data consistency is obtained. If a data item appears only once, any update to its value has to be performed only once and the updated value is immediately available to all users. If the DBMS has controlled redundancy, the database system enforces consistency.

4. **Data in Shared Form:** In DBMS, data can be shared by authorized users of the organization. The database administrator manages the data and gives rights to users to access the data. Many users can be authorized to access the same piece of information simultaneously. The remote users can also share same data. Similarly, the data of same database can be shared between different application programs. The DBMS allows the sharing of data under its control by any number of application, programs or users. There is no need to insert the data separately by each department or user. Data enter by one user can be share by all the users.

5. **Enforcement of Standards:** DBMS is enforced laws and policies in the form of standards which helps in maintaining database. Data is stored according to the standards and in uniform pattern The common standards can be implemented to all databases.

6. **Data Security:** Data must not be accessed by unauthorized persons with the help of DBMS we can ensure that proper access procedures can be implemented. Different level of security could be implemented for various types of data and operations.

7. **Solving Enterprise Requirement than Individual Requirement:** The DBMS is designed for general purpose and it is developed that person with different technical skills can use it as per the requirements. Since many types of users with varying level of technical knowledge use a database, a DBMS should provide a variety of user interface. The overall requirements of the enterprise are more important than the individual user requirements. So the DBA (Database Administrator) O can structure the database system to provide an overall service,

8. **Providing Backup and Recovery:** The DBMS provide the backup and recovery system so that data can be stored for future. It is also stored to manage the unwanted lose Similarly recovery policies are used which is automatically create the backup of data and restore data if required A DBMS must provide facilities for recovering from hardware or software failures.

9. **Cost of Developing and Maintaining System is Lower:** The cost of using DBMS is low as it requires basic resources which are generally available in the organisation. The cost involved in developing and maintaining the whole system is low whoever shitting manual data to electronic data may include labour cost which is addition to system cost.

10. **Concurrency Control:** The DBMS have control over the concurrent access to database It provides a common interface to perform access to database. The concurrent users may access data at same time being sharing by the DBMS. The DBMS designed polices such that one user access data other cannot perform updated task. At a time only one user is allowed to perform update operation other user need to wait for update the common data.

11. **Flexible System:** DBMS is a flexible system as it is designed for general purpose. The DBMS can handle small to large database and it can be redesigned to meet the requirements of the users. Moreover Database used by one system can be transformed into another system.

12. **Better Services to the User:** Because data are integrated into a single database, complex requests can be handled much more rapidly, then if the data were located in separate, non-integrated files. In many businesses, faster response means better customer service.

13. **Tools for Report Writing:** Most of the DBMS provide the report writer tools used to create reports. The users can create very easily and quickly. Once a report is created, it can be used may times and it can be modified very easily. The created reports are also saved along with database and behave like a software component.

14. **Controlled Concurrency:** In a computer file-based system, if two users are allowed to access data simultaneously, it is possible that they will interfere with each other. For example, if both users attempt to perform update operation on the same record, then one may overwrite the values recorded by the other. Most database management systems have sub-systems to control the concurrency so that transactions are always recorded with accuracy.

15. **Application Program/Data Independence:** The separation of data structure of database from the application program that uses the data is called data independence. In DBMS, we can easily change the structure of database without modifying the application program. Data is independent from one level to another level.

16. **Improved Decision Making Process:** Better-managed data and improved data access make it possible to generate better-quality information, on which better decisions are based. The quality of the information generated depends on the quality of the underlying data. Data quality is a comprehensive approach to promoting the accuracy, validity, and timeliness of the data. While the DBMS does not guarantee data quality, it provides a framework to facilitate data quality initiatives.

17. **Improvement in End-User Productivity:** The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

18. **Application Development Ease:** The application programmer need not build the functions for handling issues like concurrent access, security, data integrity, etc. The programmer only needs to implement the application business rules. This brings in application development ease. Adding additional functional modules is also easier than in file-based systems.

19. **Data Atomicity:** A transaction in commercial databases is referred to as atomic unit of work. For example, when you purchase something from a point of sale (POS) terminal, a number of tasks are performed such as;

- Company stock is updated.

- Amount is added in company's account.
- Sales person's commission increases etc.

All these tasks collectively are called an atomic unit of work or transaction. These tasks must be completed in all, otherwise partially completed tasks are rolled back. Thus through DBMS, it is ensured that only consistent data exists within the database.

20. **No Data Isolation:** Data is stored in uniform format so there is no need to make different programs for each data.

21. **Advanced Capabilities:** DBMS also provides advance capabilities for online access and reporting of data through Internet. Today, most of the database systems are online. The database technology is used in conjunction with Internet technology to access data on the web servers

1.4.4 Disadvantages of Database Management System

In contrast to the lots of advantages, there are few disadvantages as well which are discussed below:

1. **Complexity:** The database designed is no of the major challenging task in DBMS as it is complex, difficult, and time-consuming. DBMS is an extremely complex piece of software database designer's developers DBA and end users must understand this functionality to take full advantage of it. Failure to understand the system can lead to bad design decisions.

2. **Large Size:** The complexity and breadth of functionality makes the DBMS an extremely large piece of software, occupying many megabytes of space and requiring huge amounts of memory to sun efficiently

3. **Performance:** File Based system is written for a specific application a result performance is generally very good. However, the DBMS is written to be more general to cater for many applications rather than just one. So performance is very poor.

4. **Higher Impact of Failure:** In most of the organizations, all data is integrated into a single database. If database is corrupted due to power failure or it is corrupted on the storage media, then our valuable data may be lost or whole system stops.

All users and applications rely on the availability of the DBMS the failure of any component can bring operations to a hall.

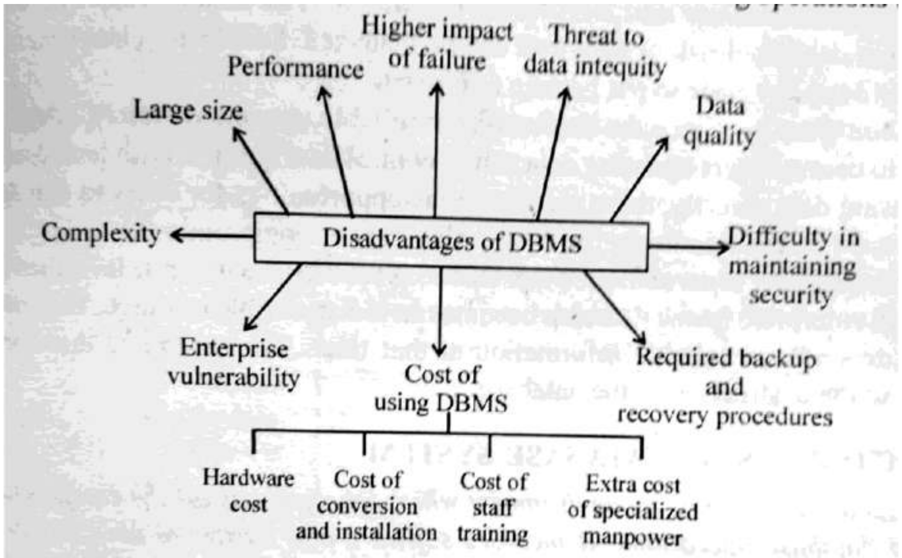


Fig. 1.7: Disadvantages of Database Management System

5. **Cost of using DBMS**

(a) **Hardware Cost:** If we want to implement DBMS then we need DBMS software which is very expensive. We need to upgrade the hardware the processing overheads to implement the security, integrity and sharing of data make the additional cost.

(b) **Cost of Conversion and Installation:** DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features. Cost of DBMS and extra hardware may be insignificant compared with the cost of converting existing applications to run on the new DBMS and hardware. This cost includes cost of training staff to use these new systems and employment of specialist staff for help. That is way some organisations feel tied to their current systems and cannot switch to modern database technology.

(c) **Cost of Staff Training:** DBMS is a complex system which demands specialized users. The user need to get training which ultimately added to the total coast.

(d) **Extra Cost of Specialized Manpower:** The DBMS is managed by skilled people includes DBA, Programmer, and data entry staff.

6. **Threat to Data Integrity:** Database is shared among different users and concurrent access is permitted in DBMS. So there is always a threat to data integrity, especially when there is transition failure.

7. **Difficulty in Maintaining Security:** Data is reserved in common place and different users are accessing data with different security levels. The user management and security access is a challenging task which requires attention. Access policy design for secure access is difficult to maintain.

8. **Required Backup and Recovery Procedures:** The database need to be backed up in time so that unwanted risk of data lose can be managed. DBMS requires special extra hard disk space and special place to put backup of old data.

9. **Data Quality:** Since the database is accessible to users remotely, adequate controls are needed to control users updating data and to control data quality. With increased number of users accessing data directly, there are enormous opportunities for users to damage the data. Unless there are suitable controls, the data quality may be compromised.

10. **Enterprise Vulnerability:** When DBMS is used in an enterprise level then centralizing all data of an enterprise in one database becomes an indispensable resource. The survival of the enterprise depends on reliable information at that time. The enterprise therefore becomes vulnerable to the destruction of the database.

1.5 COMPONENTS OF DATABASE SYSTEM

The database system is an environment which incorporates all the components required to execute database operations. It includes software and hardware used in functioning of database.

The database system is designed to provide an environment that is both convenient and efficient to perform different task on data. There are five major components in the database system environment which are as follows:

1. ***Data (Data, Metadata)***

2. ***Software (DBMS and Application Programs)***

3. ***Hardware***

4. ***Users***

5. ***Procedure***

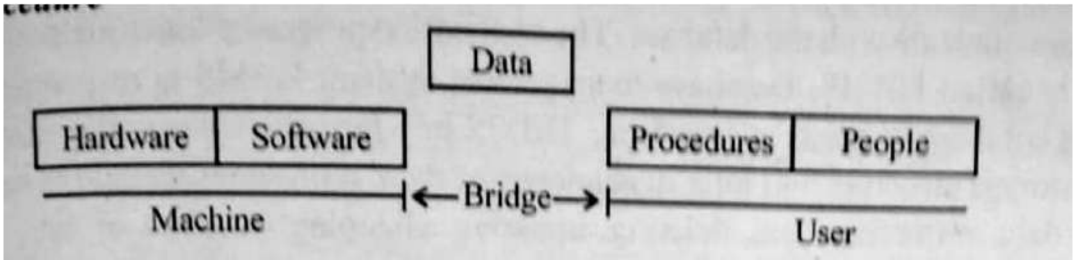


Fig. 1.8: Components of Database System

The following diagram shows these components and interaction with each other.

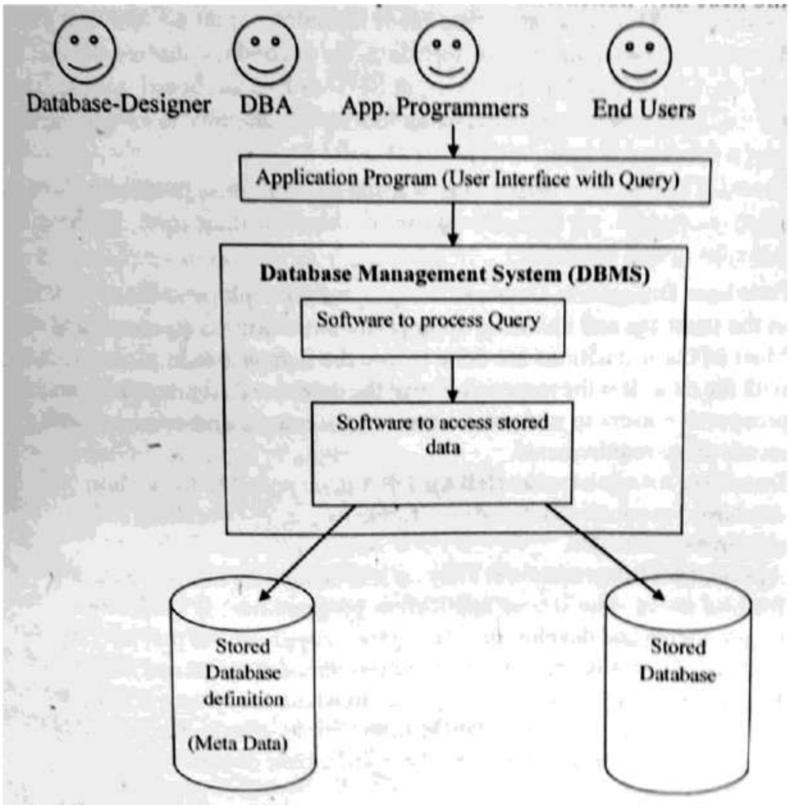

Database-Designer
DBA
App. Programmers
End Users
Application Program (User Interface with Query)
Database Management System (DBMS)
Software to process Query
Software to access stored data
Stored Database definition
(Meta Data)
Stored Database

Fig. 1.9: Database System Environments

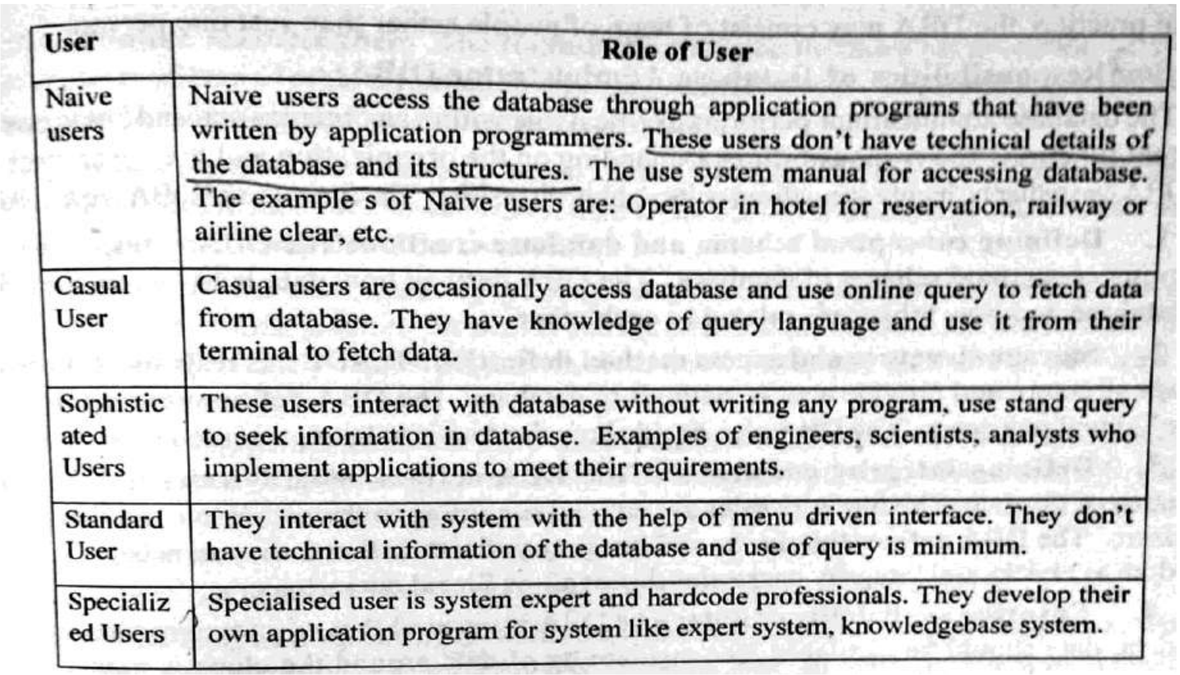
1. **Data (Data, Meta data):** The data is a fact and figure which is impotent to be stored in database. A database represents data and method used to preserve data in database. It designed to store data in such as way so that it can easily sharable and is being integrated. It also contains meta-data which is an information stored in catalo about the data. It is also called data about data or data dictionary which means it has information about data which stored in database. It defines the definition and representation information of data.

2. **Software (DBMS and Application Programs):** The software is actual DBMS. All requests from users for access to the database are handled by the DBMS. DBMS allows the users to communicate with the database. The application programs which are designed to handle database is called DBMS, Database management system. DBMS is responsible for smooth work with database and acts as interface. DBMS interface shields complex detail of data like physical storage structure and inter dependency of data. It provides method to handle data like insertion data, retrieving data, deleting, updating, changing structure of data, etc. It is also categorized into two components: Software to process Queries and Software to access stored data. The Software to process Queries deals with user interface and interaction to sort the query raised by the user whereas software to access data acts as an interface with physical database.

3. **Hardware:** The hardware component includes actual computer hardware used for keeping and accessing the database. To store data, the secondary storage devices are used such as Magnetic Disk, CD/DVD. Input and output devices like keyboard, mouse, scanner reader, monitors are used. Data processing hardware includes computer processor which plays significant role to support a database system.

4. **Users:** The users are person who is using the system as per their role or requirement. In typical database system, we categorized user on basis of their role. We have four types of users which are discussed below:

- **Database Designers:** Database designers are people who identify data to be stored in the database and choosing appropriate structures to represent and store the data Most of these functions are done before the database is implemented and populated with the data. It is the responsibility of the database designers to communicate with all prospective users to understand their requirements and come up with a design that meets these requirements.
- **Database Administrator (DBA):** DBA is responsible for authorizing access to the database, for coordinating and monitoring its use, and acquiring software and hardware resources as needed.
- **Application Programmers:** They are responsible for developing application program for end users. The job of application programmers is to determine the end use requirements and develop specifications for applications that meet the requirements. They implement the specifications as programs, then test and debug the programs.
- **End Users:** End users are those people to whom the system is designed. These use are actually accessing the database from their terminals. The end users are classified on the basis of knowledge of database and extent of use.



User	Role of User
Naive users	Naive users access the database through application programs that have been written by application programmers. These users don't have technical details of the database and its structures. The use system manual for accessing database. The example s of Naive users are: Operator in hotel for reservation, railway or airline clear, etc.
Casual User	Casual users are occasionally access database and use online query to fetch data from database. They have knowledge of query language and use it from their terminal to fetch data.
Sophisticated Users	These users interact with database without writing any program, use stand query to seek information in database. Examples of engineers, scientists, analysts who implement applications to meet their requirements.
Standard User	They interact with system with the help of menu driven interface. They don't have technical information of the database and use of query is minimum.
Specialized Users	Specialised user is system expert and hardcode professionals. They develop their own application program for system like expert system, knowledgebase system.

Table 1.3: Different End Users and their Rolls

The database system have different users as listed above, but role of DBA is significant in many aspects. We discuss the role of DBA in 1.6 in detail.

5. **Procedures:** Procedures refer to the instructions (rules) that govern the design and use of the database. The users of the system and the staff that manage the database require documented procedures to use or run the system. The followings are some instructions so that we can follow procedure systematically:

(i) Log on to the DBMS

(ii) Start and stop the DBMS

(iii) Make backup copies of the database

(iv) Handle hardware or software failures.

(v) Change the structure of a table.

1.6 DBA (DATABASE ADMINISTRATOR)

- The DBA is a person or group of persons who control the right to access to the data and over all maintain policies to ensure softy and smoothly working of the system.
- DBA controls the design and use of database. DBA is responsible for implementing the database system within an organization.
- DBA provides a necessary technical support for implementing policy for smooth working of the database.
- DBA is responsible for evaluation, selection and implementation of DBMS package.
- The DBA has to perform number of important task like authorizing access to database, coordinating and monitoring different types of users, handling software and hardware issues.
- Database Administrator's job requires a high degree of technical expertise.
- In practice, the DBA may consist of team of people rather than just one person.

Functions Responsibilities of Database Administrator (DBA)

The database administrator performs a critical role within an organization and has to perform different functions and responsibilities Depending on the organization and the department, the role DBA can either be highly specialized or incredibly diversified. The functions of DBA are as follows:

1. **Defining conceptual schema and database creation:** The DBA is responsible for designing conceptual schema of database. The DBA defined how data is to be represented in the database and how tables are related to each other.

2. **Storage structure and access method definition:** The DBA is responsible to define storage structure and provide access methods to database. The DBA defines access policies to an individual or a group. The DBA also decide how the data is to be represented in the database.

3. **Defining integrity constraints:** The DBA defines integrity rules to ensure the accuracy of the data. The integrity rules are defined according to the nature and requirement of the users. The DBA defines the checks and integrity policies so that users can access data with freedom as checks and integrity constrains do not allow illegal operations.

4. **Ensuring availability of data:** The DBA is ensured that whenever request is made for data, data should be available. The availability of data around the clock is possible with appropriate steps to take backup and switching load among different systems. The DBA defines policies such that data must be available in time.

5. **Deciding backup and recovery methods:** The backup and recovery methods are very crucial for the safety of the data as system accidental failure can happen any time. The DBA decides which data is to be backed up and when. DBA defines policy for backup so that data loss can be avoided.

6. **Granting authorisation to the users:** The DBA defines list of users with access level so that data can be access by the authorised users only. The authorisation of user is monitored and updated by DBA time to time as one user may changes his access level.

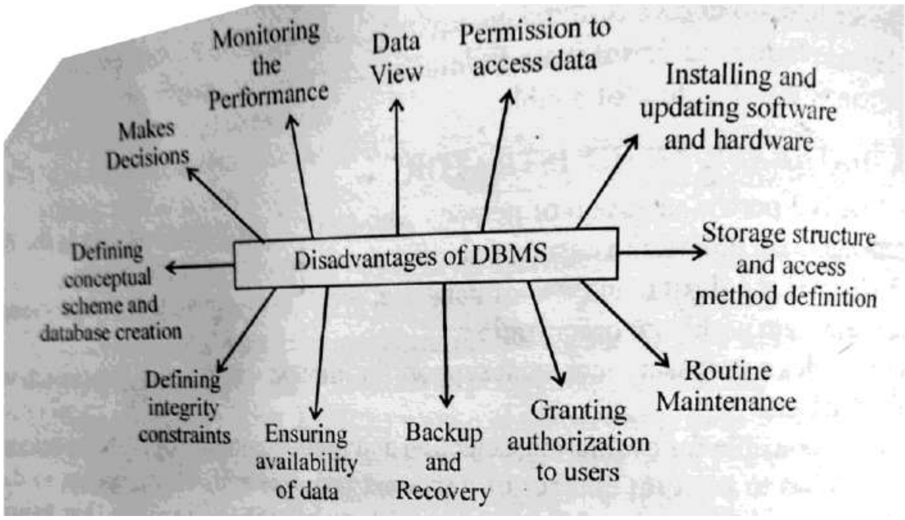

Monitoring the Performance
Data View
Permission to access data
Installing and updating software and hardware
Makes Decisions
Disadvantages of DBMS
Storage structure and access method definition
Defining conceptual scheme and database creation
Defining integrity constraints
Ensuring availability of data
Backup and Recovery
Granting authorization to users
Routine Maintenance

Fig. 1.10: Responsibilities of DBA

7. **Routine maintenance:** The routine maintenance includes up gradation of system, updating user profiles and other information regarding access policies. The Database Administrator understands the following routine maintenance activities:

- ***When transaction rollbacks occur*** DBA decides what to do when a transaction rollback. If such incidence occurs, then DBA checks the updating records and decide about re-do or undo of the transaction.
- ***When the database is out of system disk space:*** The database is stored into physical memory which in limited in size. When the data is about to reach maximum limit in disk space then DBA decide whether to erase some unwanted data or add new hard disk into system.
- ***When unique constraints have been violated:*** The access to data in database in done through some unique constrains whenever such rule is violated then DBA has to look into the matter The unique constraints policies are updated as their reports of violation such that violations can be avoided in future.
- When not to shut down the database while the application is running

8. **Installing and updating software and hardware:** The DBA is the person who is authorised to installed software into the database machine. The DBA decides the need of updating the hardware

9. **Permission to Access Data:** DBA gives permission to user to use database. Only authorised user can access data.

10. **Data view:** DBA can create different views of data that can be shown to different users.

11. **Monitoring the performance.** DBA is responsible for overall performance of the system. To improve the performance DBA regular monitor the system performance.

12. **Makes Decisions:** It is the DBA's job to decide exactly what information is to be held in the database.

1.7 COMPARISON OF FILE MANAGEMENT SYSTEM WITH DATABASE MANAGEMENT SYSTEM

Sr. No.	**Concepts**	**File Management System**	**Database Management System**
1.	Redundancy	Data redundancy (duplication) is possible. Data duplication is a commonly visible in file management system as multiple files are stored at different location.	Data redundancy (duplication) is not possible. In database management system, minimum redundancy occurred as duplication is avoidable.
2.	Consistency	Data is duplicated into number of files and consistency is a hard job in file management system. Consistency ensures that data at different location about single entity should be	Database management system avoids duplication as a result consistency can be avoided. Moreover data

		same with time.	
3.	Data Isolation	The data isolation is preserve in the file management system as data is stored in different file which are hardly associate with each other.	The data is stored into tables which are linked with each other so the isolation of data is not available.
4.	Standards enforcement	In an organization, every department has their own file system and format so uniform format and standards cannot be implemented. Moreover application programs are file dependent so new standard enforcement is hard to implement.	DBMS is enforced laws and policies in the form of standards which helps in maintaining database. Data is stored according to the standards and in uniform pattern so any programs are file dependent so new change in standard policies do not affect the working of the database.
5.	Data security	Data security is implemented at file level, whereas user level security itself is not feasible.	Data security is implemented at different levels and data security policies are upgradable to meet

			the requirements.
6.	Application dependency	The applications are developed according to the data file, it means we have to make certain changes whenever application applied on different format data file.	The application programs in DBMS are independent on database. Data independency is enforced in the system which helps the programmer to write general purpose programs
7.	Multiple Access	The multiple access to data file is not permitted in file management system.	In database management system, multiple access is permitted.
8.	Concurrency Problem	There is concurrency problem in file management system as multiple accesses is not desirable.	DBMS is designed to meet concurrent access which means more than one user can access same data without any problem.
9.	Technical platform	In file management system, every department developed their own applications to access data files. Generally applications are developed	DBMS provides integrated program kit which is developed using common language. Examples of DBMS are Oracle,

		in C, C++, COBOL etc.	Sequel & Foxpro etc.
10.	Real world modelling	Real world modelling is not possible as files management system stored data in files. The real world modelling requires object based data representation which is not possible in file based system.	Real world modelling is done through object representation in DBMS as data along with other attributes can be stored.
11.	No. of Files	There are less number of files as compared to DBMS	There are more number of files.
12.	Cost	It is cheaper as compared to DBMS.	It is costly
13.	Structure	It has simple structure	It is complex structure.
14.	Flexibility	It is less flexible as compared to DBMS.	It is more flexible.
15.	Efficiency	When the volume of data increases, its efficiency decreases	Volume of data not affect its working capability.

1.8 CATEGORIES OF DBMS

The classification of a database management system (DBMS) is greatly influenced by the underlying computing system on which it runs, in particular of computer architecture such as parallel, networked or distributed. However, the DBMS can be classified according to the number of users, the database site locations and the", expected type and extent of use.

1.8.1 Centralized DBMS

1. In centralized .database systems, the database system, application programs, and user-interface all are executed on a single system and dummy terminals are connected to it.
2. It is physically confined to a single location.
3. The processing power of single system is utilized and dummy terminals are used only to display the information.
4. As the personal computers became faster, more powerful, and cheaper, the database system started to exploit the available processing power of the system at the user's side, which led to the development of client/server architecture,
5. In client/server architecture, the processing power of the computer system at the user's end is utilized by processing the user-interface on that system.
6. The centralised database system .consists of a single processor together with its associated data storage devices and other peripherals.
7. The system offers data processing capabilities to users who are located either at the same site, or, through remote terminals, at geographically dispersed sites.-
8. The management of the system and its data are controlled centrally form any one or central site.

The following diagram 2,6 show the centralized DBMS.

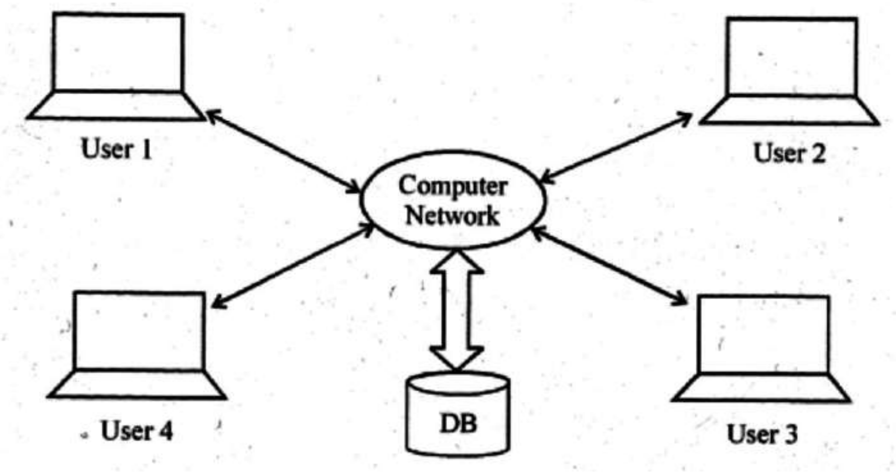


Fig. 2.6: Centralized DBMS

Advantage of a Centralized DBMS

1. **Centralized control:** The organization can exert centralized management and control over the data by Database Administrator (DBA). The database administrator is the focus of centralized control./
2. **Shared data:** A database allows the sharing of data under its control by any number of application programs or users.
3. **Reduction of redundancies: Centralized** control of data by DBA avoids unnecessary duplication of data and effectively reduces the total amount of data storage required. It also eliminates the extra processing necessary to trace the required data in a large mass of data.
4. **Integrity:** Centralized control can also ensure that adequate checks are incorporated in the DBMS to provide data integrity. Data integrity means that the data contained in the database is both accurate and consistent. Therefore data values being entered for storage could be checked to ensure that they fall within a specified range and in the correct format.
5. **Security:** Data is a vital importance to an organization and may be confidential. Such confidential data must not be accessed by unauthorized person.
6. **Data Independence:** Data independence allows dynamics changes and growth potential.
7. **Operations:** Most of the functions such as update, backup, query, control access and so on, are easier to accomplish in a centralised database system.
8. **Size of the Database:** The size of the database and the computer on which it resides need not have any bearing on whether the database is centrally located.

Disadvantage of a Centralized DBMS

1. **Problems associated with centralization:** Several problems are associated

with centralization like networking the excessive load on the system at the central site would likely causes all accesses to be delayed etc.

2. **Cost of software and migration:** The cost of purchasing or developing the software, the hardware has to be upgraded to allow for the extensive programs and the work spaces required for their execution and storage. The processing overhead is also added by implement security integrityof data causes a degradation of the response and through put times. It is also added the cost of migration from .a traditionally separate application environment to an integrated one.
3. **Complexity of backup and recovery:** The centralization reduces duplication, the lack of duplication required that the database be adequately backed up so that in the case of failure the data can be recovered. Backup and recovery operations are fairly complex in a DBMS environment.
4. **Server Down:** When the central site computer or database system goes down, then every user is blocked from using the system until the system comes back.
5. **Communication costs:** The communication costs from the terminals to the central site can be expensive.

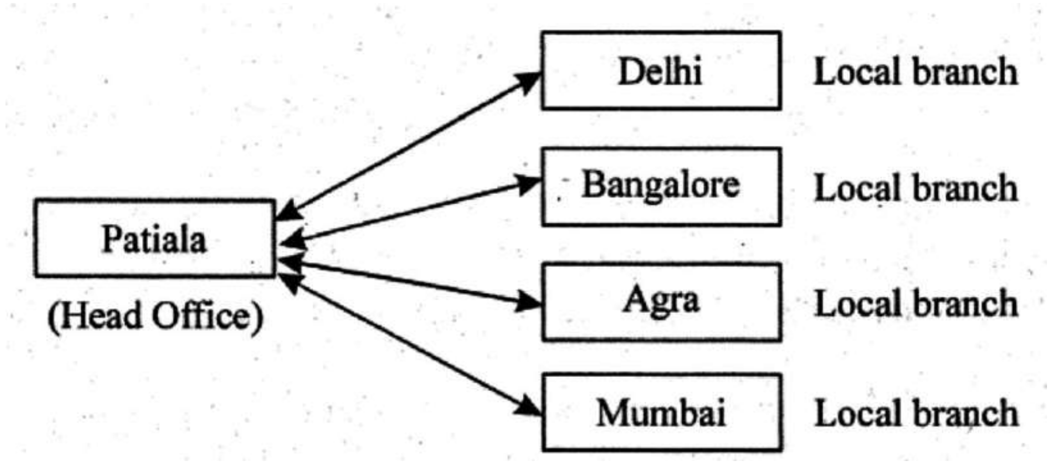
1.8.2 Parallel DBMS

1. Parallel database systems architecture consists of a multiple central processing units (CPUs) and data storage disks in parallel.
2. They improve processing and input/output (I/O) speeds.
3. Parallel DBMS are used in the applications that have to. query extremely large databases or that have to process an extremely large number of transactions per second.

The following diagram 2.7 shows the parallel DBMS.

- Shared data storage disk
- Shared memory
- Hierarchical

- Independent resources



Advantages of-a Parallel DBMS.

1. Parallel database systems are very useful for the applications that have to query extremely large databases.
2. In a parallel database system, the throughput and the response time are very high. Throughtput is number of tasks completed in -given time duration. Response time is amoung of time required by single task for completion.

Disadvantages of a Parallel DBMS

1. In a parallel DBMS, there is a .startup cost associated with initiating a single process and the startup-time may overshadow the processing time, affecting speedup adversely.
2. In parallel DBMS, the processes access the shared resources which slow down the result.

1.8.3 Distributed DBMS

1. Distributed DBMS consist soft of a single logical database that is spilt into number of fragments.
2. Distributed database systems are similar to client/server architecture in a number of ways.
3. Both typically involve the use of multiple computer systems and enable users to access data remote system.

4. Distribute database system broadens the extent to which data can be shared well beyond that which can be achieved with the client/server system.

Following diagram 2.8 shows the distributed DBMS architecture.

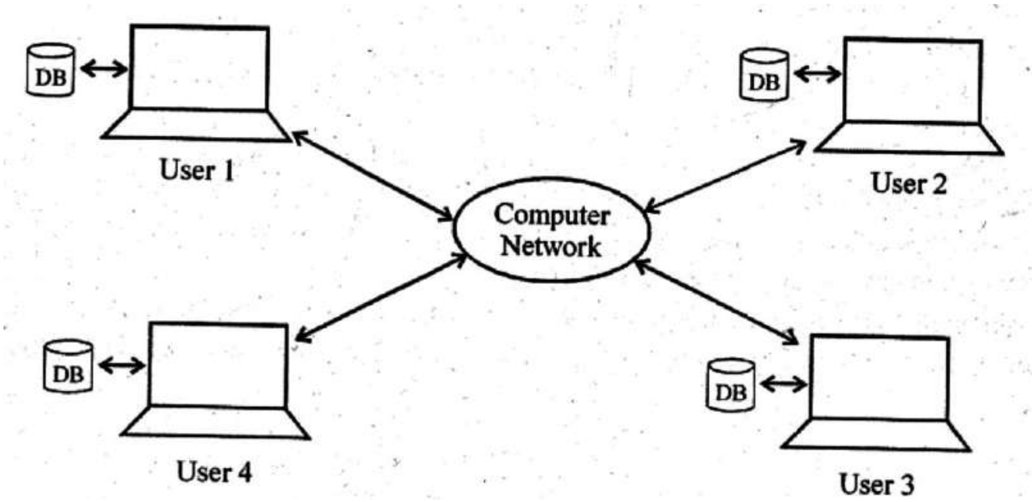


Fig. 2.8: Distributed DBMS

Advantages

1. **Efficiency and better Performance:** Distributed database architecture provides greater efficiency and better performance.
2. **Response time:** The response time and throughput is high as data is available at different places.
3. **Custom-built Machine:** The server database machine can be custom-built or tailored to the DBMS function and thus can provide better DBMS performance.
4. **Customized user interface:** The client application-database might be a personnel workstation tailored to the needs of the .end users and thus able to provide better interfaces, high availability, faster responses and overall improved ease of use to the user.
5. **Shearing of Database:** A single database on server can be shared across several distinct client application systems.
6. **Adding new location:** It causes less impact on ongoing operations when adding new locations. As data volumes and transaction rates increase, users

can grow the system incrementally.

7. Local autonomy: Distributed database system provides local autonomy.

Disadvantage of Distributed DBMS

The recovery from failure is more complex in distributed database systems than in centralized systems.

1.8.4 Client/Server Database System

1. Client/server architecture of database system has two logical components namely client, and server.
2. Clients are generally personal computers or workstations whereas server is large workstations, mini range computer system or a mainframe computers system.
3. The server computer is called backend and the client's computer is called front-end. These server and client computers are connected into a network.
4. The applications and tools of DBMS act as clients, making requests for its. services.
5. DBMS software resides on the server.
6. The DBMS, in turn, processes these requests and returns the results to the client(|).
7. The client/server architecture is a part of the open systems architecture in which all computing hardware, operating systems, network protocols and other software are interconnected as a network and work in concert to achieve user goals.
8. It is well suited for online transaction processing and decision support applications, which tend to generate a number of relatively short transactions and require a high degree of concurrency.

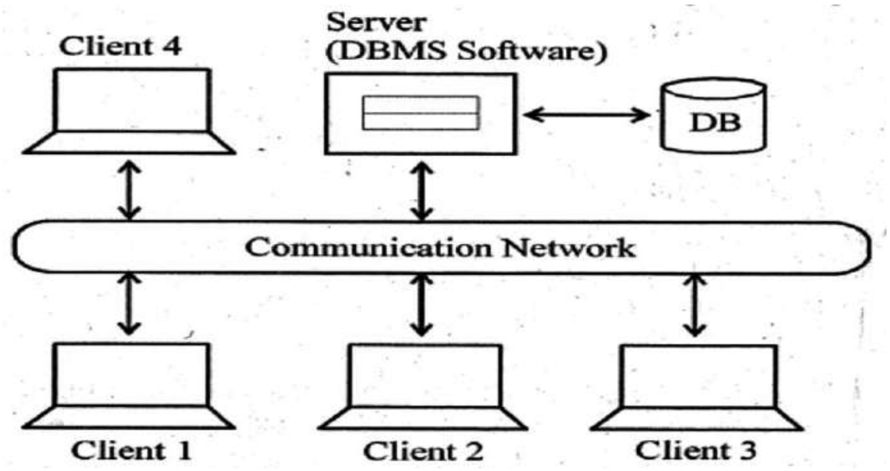


Fig. 2.9: Client Server DBMS

As shown in Fig. 2.9, the client/server database architecture consists of three components namely, client applications, a DBMS server and a communication network interface. The client applications may be tools, user-written applications or vendor-written applications. They issue SQL statements for data access. The DBMS server stores the related software; processes the SQL statements and returns results. The communication network interface enables client applications to connect to the server, send SQL statements and receive results or error messages or error return codes after the server has processed the SQL statements. In client/server database architecture, the majority of the. DBMS services are performed on the server.

Advantages of Client/Server DBMS

1. **Less expensive:** Client-server system has less expensive platforms to support^ applications that had previously, been running only on large and expensive mini or mainframe computers.

2. **Menu-drive interface:** Clients offer icon-based menu-driven interface, which is superior to the traditional command-line, dumb terminal interface typical of mini and mainframe computer systems.

3. **Flexible and productive environment:** Client-server database system is more flexible as compared to the centralised system. Client/server environment facilitates in more productive work by the users and making better use of existing data.

4. **Response time and throughput:** The client server model is based on

request and reply model when a machine make request to the server then server immediately reply so Response time and throughput is high.

5. **Custom-built Servers:** The database server machine can be custom-built or tailored to the DBMS function-and thus can provide a better DBMS performance.

6. **Custom build Client machine:** The client application database might be a personnel workstation, tailored to the needs of the end users and thus able to provide better interfaces, high availability, faster responses and overall improved ease of use to the user.

7. **Powerful Single Server:** A single database on server can be shared across several distinct client application systems.

Disadvantages of Client/Server DBMS

1. **High set cost:** The setup cost is high which include labour or programming, cost is high in client/server environments, particularly in initial phases.

2. **Lack of management tools:** There is a lack of management tools for diagnosis, performance monitoring and tuning and security control, for the DBMS, client and operating systems and networking environments.

Questions

1. What do you mean by data? How is it different from information, explain by example?

2. What is database system? What are four components of database system?

3. What are advantages of database system?

4. What is DBMS? What are the advantages and disadvantages offered by such system?

5. What are the main responsibilities of DBA? Explain.

6. What do you mean by file system? Explain it limitations.

7. Compare file management system with database management system.

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 2: DBMS ARCHITECTURE

2.INTRODUCTION

Architecture of Database Management System: *The architecture of DBMS is a framework for describing database concepts and specifies the structure of the database* *system. It describes Junctions of each component and describes how these components communicate with each other in a logical manner.*

The database management system Is a sophisticated software application which is designed to provide interface to the user so that user can perform different operations on database with ease. The design of a database management system highly depends on its architecture. It can be centralized or decentralized or hierarchical depending upon the type of applications. Its architecture can be single tier or multi-tier. The multi-tier architecture divides the database management system into related but independent different modules which can be independently modified, altered, changed or replaced. In case of multi-tier architecture, the best suitable architecture is 3-tier architecture.

2.1 THREE LEVEL ARCHITECTURE OF DBMS

Database management system is described in three different levels which have separate functioning and working. These three different levels are named as **external level, conceptual level** and **internal level.** These levels are shown in a serial view of the architecture:

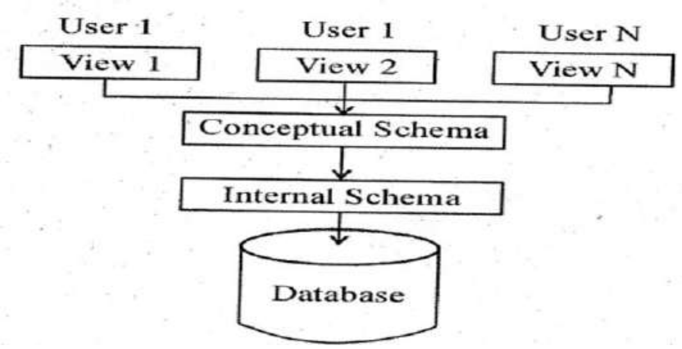


Fig. 2.1: Three Level Architecture (an Aerial view)

2.1.1 Objectives of Architecture

1. It should provide an interface to make changes into the structure of database without changing the application program at external schema.
2. Each user should be able to change the way he view the data and his change should not affect other users.
3. User should not directly deal with the physical database storage.

4. Users are independent of the storage complexities like indexing constraints etc. of the database

6. The conceptual structure of the database has no effect due to the change of the physical storage devices.

7. DBA should be able to change the storage structure and conceptual structure without affecting user's and his view level.

The core objective to design three levels is to provide data independence and physical independence. It is required to provide an easy to use interface to the end users.

2.1.2 External Level/External View

The external level is more concerned with the way in which the data is viewed by individual users. It is closer to the user and provides an interface to interact with the database. Each user has different requirement of the data so DBMS presents each user with a shared or single view or schema of the data. In external level, the different views may have different representations of the same data. For example, one user may view date in the form as (day-month-, year) while another may view as (year-month-day). Similar one view of data may show detail of employee with fields (Name, DOB, Address) and other view may show employee detail with salary (Name, DOB, Basic Pay, HRA, DA). The external view is user specific and provides an abstraction of data which helps to user to view important data and hide additional information.

Characteristics/Functions/Key Points of External Level

1. The external level is at the highest level of database abstraction where only those data is visible to the user which is concerned to the user at that time.

2. External view is user's view of database. It may provide limited and complete access to the database.

3. External schema consists of definition of logical records and their relationships in the external view.

4. External level is also known as view level and closest to the end users. It acts-as an interface to access data. User need not to know the details of data structure and physical storage.
5. External level provides the way in which individual users can view data according to his/her requirements i.e. one user may view data in the form (day, month, year) while another user may view data as (year, month, day).
6. Same database can have different views for different users.
7. Its core purpose is to provide user friendly interface to the end user.

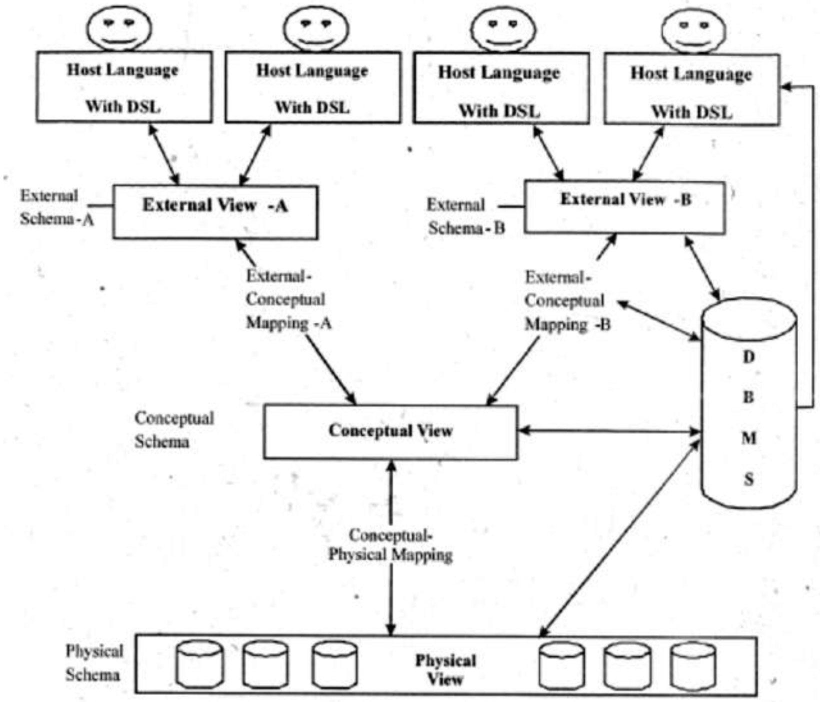


Fig. 2.2: Three Level Architecture with Different Schemas

2.1.3 Conceptual Level/Community User View/Logical Level

Conceptual Level represents the entire database. Conceptual schema describes the records and relationship included in the Conceptual view. The external level is concerned with individual user view whereas the conceptual level represents community user view. The conceptual schema hides the details of physical structure and concentrates on describing entities, data type, relationships, user operations and constraints. The view is normally more stable than the other two views. The ultimate objective of the conceptual schema is to describe the complete enterprise-not just its data but also how that data is used, how it flows from point to point within the enterprise.

Characteristic/Functions/Key Points of Conceptual Level

1. Conceptual level is also known as middle level. It is created and maintained by DBA.
2. The conceptual schema hides the details of physical structure and concentrates on describing data type entities, their attributes and relationships, user operations.
3. It implements constraint on fie data.
4. At this level, different security and integrity rules can be imposed on data.
5. The semantic information about the data can be represented in conceptual view.
6. Different types of validation checks to retain data consistency and integrity are enforced at conceptual level.
7. It describes what data is stored in database and relationship among database.

2.1.4 Internal Level/Storage View/Physical Level

The internal level is closest to the physical storage which is concerned with the way in which the data is actually stored. The internal view is described by means of the internal schema, which not only defines the various stored record types but also specifies the indexes are in and so on.

Characteristic/Functions/Key Points of Internal Level

1. It is the physical representation of data.
2. It describes how the data is stored in database. It manages storage space allocation for data.
3. It concern with the physical implementation of the database to achieve optimal runtime performance and space utilization.
4. Record description for storage with stored sizes for data items.

5. Access path e.g. specification of primary and secondary keys, index and pointers.
6. Data compression and encryption techniques.
7. Optimization of the internal structures.
8. It builds the indexer, retrieve the data and so on.

2.1.5 Database Schema and Database Instance

While working with any data model, it is necessary to distinguish between the overall design or description of the database (database schema) and the database itself. The database schema is also known as intension of the database, and is specified while designing the database.

1. Schema

A schema is plan of the database that gives the names of the entities and attributes and the relationship among them. A schema includes the definition of the database name, the record type and the components that make up the records. Alternatively, it is defined as a framework into which the values of the data items are fitted. The values fitted into the framework changes regularly but the format of schema remains the same.

Key Points of Schema

- The plan or scheme of the database is known as Schema.
- It gives the names of the entities, attributes and relationship among them.
- It is the framework into which the values of data items are fitted.
- Overall description of database is known as database schema.

Types of Schema

Generally, a schema can be partitioned into/three categories which are as follows:

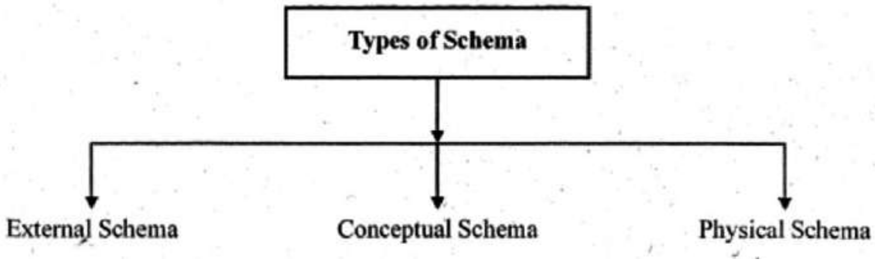


Fig. 2.3: Types of Schema

(a) ***External Schema:*** The external schema is concerned with the description of external view, correspond to different view according to the requirements of the users.

(b) ***Conceptual Schema:*** The conceptual schema is concerned with the description of all the entities, attributes and relationships along with the constraints. The logical (conceptual) schema is concerned with exploiting the data structures offered by the DBMS so that the schema becomes understandable to the computer.

(c) ***Physical Schema:*** The physical schema is concerned with the manner in which the conceptual database gets represented in the computer as a stored database. It is hidden behind the conceptual schema and can usually be modified without affecting the application programs.

2. Subschema

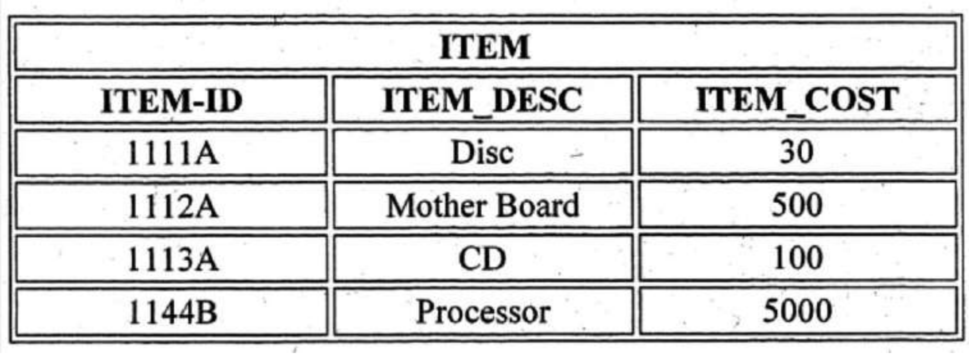
- A subschema is a subset of the schema having the same properties that a schema has.
- It identifies a subset of areas, sets, records, and data names defined in the database schema available to user sessions.
- It allows the user to view only that part of the database that is of interest to him.
- It defines the portion of the database as seen by the application programs and the application programs can have different view of data stored in the database.
- The different application programs can change their respective subschema without affecting other's subschema or view.

3. Instances

- The data in the database at a particular moment of time is called an instance or a database state.

- In a given instance, each schema construct has its own current set of instances. Many instances or database states can be constructed to correspond to a particular database schema.
- Every time we update (i.e., insert, delete or modify) the value of a data item in a record, one state of the database changes into another state.

The following figure shows an instance of the ITEM relation in a database schema.



ITEM		
ITEM-ID	**ITEM_DESC**	**ITEM_COST**
1111A	Disc	30
1112A	Mother Board	500
1113A	CD	100
1144B	Processor	5000

2.1.6 Mapping Between Different Views

Mapping: In three schema architecture, each user group refers only to its own external view. Whenever a user specifies a request to generate a new external view, the DBMS must transform the request specified at external level into a request at conceptual level, and then into a request at physical level. If the user requests for data retrieval, the data extracted from the database must be presented according to the need of the user. *This process of transforming the requests and results between various levels of DBMS architecture is known as mapping.*

The DBMS is responsible for mapping between the three types of schema. Two mapping are required in database systems which are as follows:

(a) **External/Conceptual Mapping:** Each external scheme is related to the conceptual schema by the external/conceptual mapping. The external/ conceptual mapping gives the correspondence among the records and the relationships of the external and conceptual views. A given external record could be derived from a number of conceptual records.

(b) **Conceptual/Internal Mapping:** Conceptual schema is related to the internal schema by the conceptual/internal mapping. The conceptual/ internal mapping specifies the method of deriving the conceptual record from the physical database.

Advantages of View Mapping

1. Each user is able to access the same data but have a different customized view of the data as per their own needs.
2. A user can change his/her view and this change does not affect other user views.
3. There user's interaction with the database is independent of physical data storage organization.
4. The database administrator is able to change the database storage structure without affecting the user's view.
5. The database administrator is able to change the conceptual structure of the database without affecting all users.
6. The database administrator can change existing storage devices with the new storage devices without affecting others user's.

2.2 EXAMPLE OF THREE LEVEL ARCHITECTURE

To understand the three-schema architecture, consider the three levels of the BOOK file in Online Book database as shown in Figure this figure, two views (view 1 and view 2) of the BOOK file have been defined at the external level. Different database users can see these views. The details of the data types are hidden from the users. At the conceptual level, the BOOK records are described by a type definition. The application programmers and the DBA generally work at this level of abstraction. At the internal level, the BOOK records are described as a block of consecutive storage locations such as words or bytes. The database users and the application programmers are not aware of these details; however, the DBA may be aware of certain details of the physical organization of the data.

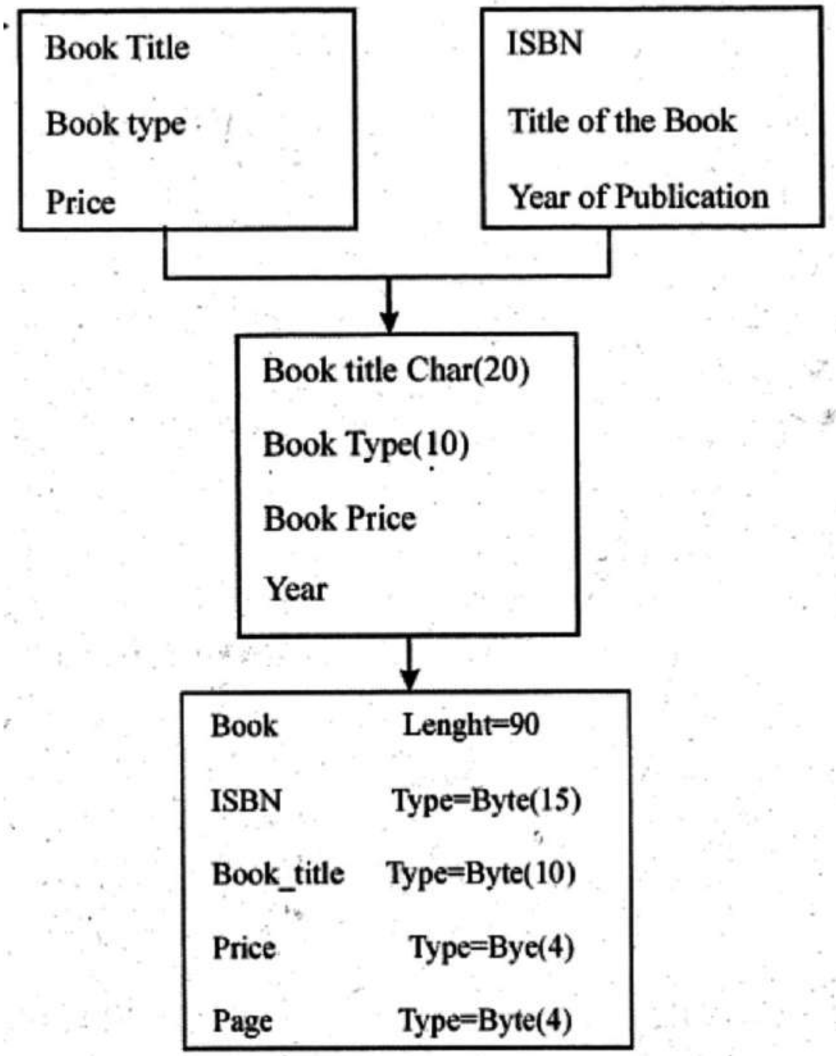


Fig. 2.4: Three Level Schema Architecture

2.3 DATA INDEPENDENCE

Data Independence: *The ability of a database management system to modify its Schema definition at one level without affecting a Schema definition at the next level is called data Independence. It provides flexibility to make changes at one Schema level without affecting the next level Schema.*

Key Points of Data Independence

- The main advantage of three-schema architecture is that it provides data independence.

- Data independence Is the ability to change the, schema at one level of the database system without having to change the schema at the other levels.
- The data independence deals with independence between the way the data is presented, structured and stored. It provides independence to make changes in one level without affecting other levels.
- Data independence means upper levels are unaffected by the changes in lower level.
- For example, DBMS may change the structure of the data without having to change application program. It is possible due to mapping between three levels which enable the user to make changes at one level without affecting the next level of architecture.
- Data Independence implies that the application programs should not need to know any of the following:
  - Ordering of data fields in a record
  - The size of the record
  - The size of the field
  - The format and type of each data item
  - The type of data structured used to store the data.
- The three level DBMS architecture provides two type of data independence. The first is called *logical data independence* and second is called physical data independence. The logical independence is enabled the user to change the conceptual view without affecting the external view. Whereas, Physical data independence is the idea to make changes into internal view without affecting the conceptual or external views. These two types of data independence are discussed in detail below:

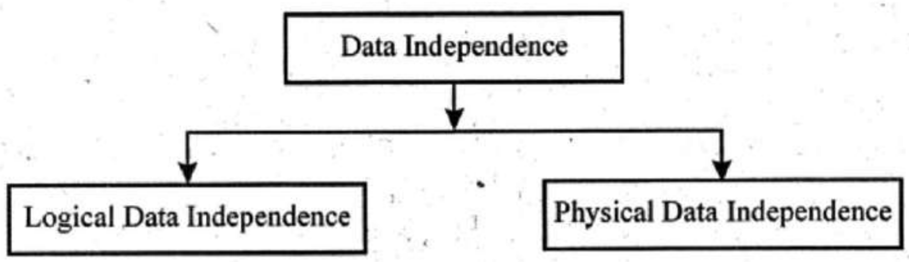


Fig. 2.5: Types of Data Independence Logical Data Independence*

- It is the ability to change the conceptual schema without affecting the external schemas or application programs.
- The conceptual schema may be changed due to change in constraints or addition of new data item or removal of existing data item, etc., from the database.
- The separation of the external level from the conceptual level enables the users to make changes at the conceptual level without affecting the external level or the application programs.
- For Example: The name field in conceptual view is stored as first hame, middle name and last name whereas in external view, it remains to be as a single name field.
- It indicates that the conceptual schema can be changes without affecting the existing external schema.
- It requires the flexibility-in the design of database.
- The programmer is required to make modification in the design as per the requirements.

(b) Physical data independence:

- It is the ability to change the internal schema without affecting the conceptual or external schema.
- An internal schema may be changed due to several reasons such as for creating additional access structure, changing the storage structure, etc.
- The separation of internal schema from the conceptual schema facilitates physical data independence.

- For Example: The location of the database, if changed from C drive to D drive will not affect the conceptual view or external view as the commands are independent of the location of the database.
- It indicates that the physical storage structure used for thedata could be change without affecting the conceptual schema.
- The storage structure and access methods used to retrieve of the data from physical storage medium are not concerned with conceptual schema.

Logical data independence is more difficult to achieve than the physical data independence because the application programs are always dependent on the logical structure of the database. Therefore, the change in the logical structure of the database may require change in the application programs.

2.4 DIFFERENCE BETWEEN LOGICAL DATA INDEPENDENCE AND PHYSICAL DATA INDEPENDENCE

Sr. No.	Logical Data Independence	Physical Data Independence
1.	Whenever, there is a change or modification at the conceptual level without t affecting the user level or external level, it is "known as logical data independence.	Whenever, the changes are made at the internal level without affecting the above layers, it is known as physical data independence.
2.	It is concerned with the structure of the data or changing the data definition	It is concerned with the storage of the data.
3.	It is concerned with the conceptual schema	It is concerned with the internal schema.
4.	Application program need not be change if new fields are added	Physical database is concerned with the change of the storage device
5.	It is very difficult to retrieve the data because data re heavily dependent on the	It is easy to retrieve the data.

	logical structure of data	

2.5 COMPONENTS OF A DBMS

The DBMS accepts the SQL commands generated from a variety of user interfaces, produces query evaluation these plans against the database, and returns the answers.

1. **Query processor:** The query processor transforms users queries into a series of low-level instructions directed to the run time database manager. It is used to interpret the online user's query and convert it into an efficient series of operations in a form capable of being sent to the run time data manager for execution. The query processor uses the data dictionary to find the structure of the relevant portion of the database and uses this information in modifying the query and preparing an optimal plan to access the database.

2. **Run time database manager:** Run time database manager is the central software component of the DBMS, which interfaces with user-submitted application programs and queries. It handles database access at run time. It converts operations in user's queries coming directly via the query processor or indirectly via an application program from the user's logical view to a physical file system. It-accepts queries and examines the external and conceptual schemas to determine what conceptual records are required to satisfy the users request. The run time data manager then places a call to the physical database to perform the request. It enforces constraints to maintain-the consistency and integrity of thewell as its security. It also performs backing and recovery operations. Run time database manager is sometimes referred to as the *database control system* and has the following components:

(i) **Authorization control:** The authorization control module checks that the user has necessary authorization to carry out the required operation.

(ii) **Command processor:** The command processor processes the queries passed by authorization control module.

(iii) **Integrity checker:** The integrity checker checks for necessary integrity constraints for all the requested operations that changes the database.

(iv) **Query optimizer:** The query optimizer determines an optimal strategy for the query execution. It uses information on how the data is stored to produce an efficient execution plan for evaluating query.

(v) **Transaction manager:** The transaction manager performs the required processing of operations it receives from transactions. It ensures that (a) transactions request and release locks according to a suitable locking protocol and (b) schedules the execution of transactions.

(vi) **Scheduler:** The scheduler is responsible for ensuring that concurrent operations on the database proceed without conflicting with one another. It controls the relative order in which transaction operations are executed.

(vii) **Data manager:** The data manager is responsible for the actual handling of data in the database. This module has the following components:

(a) ***Recovery manager:*** The recovery manager ensures that the database remains in a consistent state in the presence of failures. It is responsible for (a) transaction commit and abort operations, (b) maintaining a log, and (c) restoring the system to a consistent state after a crash.

(b) ***Buffer manager:*** The buffer manager is responsible for the transfer of data between the main memory and secondary storage (such as disk or tape). It brings in pages from the disk to the main memory as needed in response to read user requests. Buffer manager is sometimes referred as the *cache manager*.

3. **DML processor:** Using a DML compiler, the DML processor converts the DML statements embedded in an application program into standard function calls in the host language. The DML compiler converts the DML statements written in a host programming language into object code for database access. The DML processor must interact with the query processor to generate the appropriate code.

4. **DDL processor:** Using a DDL compiler, the DDL processor converts the -

DDL statements into a set of tables containing metadata. These tables contain the metadata concerning the database and are in a form that can be used by other components of the DBMS. These tables are then stored in the system catalog while control information is stored in data file headers. The DDL compiler processes schema definitions, specified in the DDL and stores description of the schema (metadata) in the DBMS system catalog. The system catalog includes information such as the names of data files, data items, storage details of each data file, mapping information amongst schemas, and constraints.

2.6 DATA DICTIONARY

1. Data dictionary is also known as *Meta data.* A metadata is the data about the data. It is the self-describing nature of the database that provides program-data independence. It is also called as the *System Catalog.*

2. ***Data Dictionary is a repository-of information about a database that documents data elements of a database. It stores information about the database, attribute names and definitions for each table in the database***

3. It holds the following information about each data element in the databases, it normally includes:

- Name
- Range of values
- Access authorization
- Type
- Source

4. Data dictionary is the integral part of the DBMS. Maintaining the data dictionary is the responsibility of DBA (Database Administrator).

5. The most general structure of data dictionary is shown in figure....

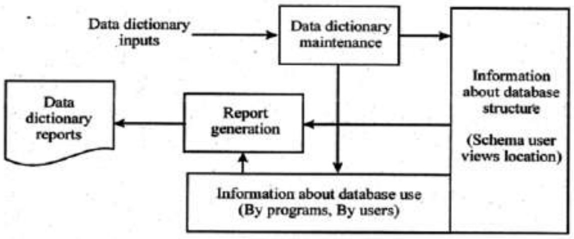

Data dictionary inputs
Data dictionary maintenance
Information about database structure
(Schema user views location)
Data dictionary reports
Report generation
Information about database use
(By programs, By users)

Fig. 2.10: Data Dictionary

6. Data dictionary is usually a part of the system catalog that is generated for each database. *A useful data dictionary system usually stores and manages the following types of information:*

- Descriptions of the schema of the database.
- Detailed information on physical database design, such as storage structures, access paths and file and record sizes.
- Description of the database users, their responsibilities and their access rights.
- High-level descriptions of the database transactions and applications and of the relationships of users to transactions.
- The relationship between database transactions and the data items referenced by them. This is useful in determining which transactions are affected when certain data definitions are changed.
- Usage statistics such as frequencies of queries and transactions and access counts to different portions of the database.
- Data dictionary provides the name of a data element, its description and data structure in which it may be found.
- Data dictionary provides great assistance in producing a report of where a data element is used in all programs that mention it.
- It is also possible to search for a data name, given keywords that describe the name. For example, one might want to determine the name of a variable that stands for net pay. Entering keywords would produce a list of possible identifiers and their definitions. Using keywords one can search the dictionary to locate the proper identifier to use in a program.

7. Data dictionary is used by developers to develop the programs, queries, controls-and procedures to manage and manipulate the data. It is available to database administrators (DBAs), designers and authorized user as on-line system documentation. This improves the control of database administrators (DBAs) over the information system

and the user's understanding and use of the system.

2.7 DBMS LANGUAGES

1. The main objective of a database management system is to allow its users to perform a number of operations on the database such as insert, delete, and retrieve data in abstract of data.
2. To provide the various facilities to different types of users, a DBMS normally provides one or more specialized programming languages called **Database** (or **DBMS**) **Languages.**
3. The DBMS mainly provides two database languages, namely, data definition language and data manipulation language to implement the databases.
4. Data definition language (DDL) is used for defining the database schema. The DBMS comprises DDL compiler that identifies and stores the schema description in the DBMS catalog.
5. Data manipulation language (DML) is used to manipulate the database.

The following are the DBMS languages:

1. **Data Definition Language:** DDL is used to specify the structure of table.

Sr.	Need And Usage	The SQL DDL Statement
1	Create schema objects	CREATE
2	Alter schema objects	ALTER
3	Delete schema objects	DROP
4	Rename schema objects	RENAME

We will discuss these statement in the chapter 10.

2. **Data Manipulation Language:** The DBMS provides data manipulation language (DML) that enables users to retrieve and manipulate the data. The statement which is used to retrieve the information is called a query. The part of the DML used to retrieve the information is called a query language.

S No.	**Need And Usage**	**The SQL DDL Statement**

1	Remove rows from tables or views	DELETE
2	Add new rows of data into table or view	INSERT
3	Retrieve data from one or more tables	SELECT
4	Change columns values in existing rows of a table or view	UPDATE

We will discuss these statement in the chapter 10.

3. **Data Control Language-(DCL):** DCL statements control access to data and the database using statements such as GRANT and REVOKE. A privilege can either be granted to a user with the help of GRANT statement. We can also revoke these statements by using REVOKES command.

S. NO.	**Need and Usage**	**The SQL DDL Statement**
1	Grant and take away privileges and roles	GRANT and REVOKE
2	Add a comment to the data dictionary	COMMENT

We will discuss these statement in the chapter 10.

Questions

1. Discuss the concept of data independence and explain its importance in a database environment.
2. What is logical data independence and why is it important?
3. What is the difference between physical data independence and logical data independence?
4. Explain the difference between external, conceptual and internal schemas. How are these different schema layers related to the concepts of physical and logical data independence?
5. Describe the structure of a DBMS.
6. Describe the main components of a DBMS with a neat sketch, explain the structure of DBMS.
7. What do you mean by a data model? Describe the different types of data models

used.

8. Explain the following with their advantages and disadvantages:
   (a) Hierarchical database model
   (b) Network database model E-R data models
   (c) Relational database model
   (d) E-R data models
   (e) Object-oriented data model.
9. Define the following terms:
   (a) Data independence
   (b) Query processor
   (c) DDL processor
   (d) DML processor.
   (e) Run time database manager.
10. What is meant by the term client/server architecture and what are the advantages and disadvantages of this approach?
11. Compare and contrast the features of hierarchical, network and relational data models. What business needs led to the development of each of them?
12. Differentiate between schema, subschema and instances.
13. Explain the advantages and disadvantages of a centralised DBMS.
14. Explain the advantages and disadvantages of a parallel DBMS.
15. Explain the advantages and disadvantages of a distributed DBMS.
16. Explain data dictionary in detail.

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 3 : DATA MODELS

INTRODUCTION

A Data Model defines the logical design of the data. It describes the relationships between different parts of the data. Data model tells how the logical structure of a *database is modeled. Data Models are fundamental entities to introduce abstraction in DBMS. Data models define how data is connected to each other and how it will be processed and stored inside the system.*

Evolution of Major Data Models

The historical literature reported drastic changes in the year 1970-1994. The Edgar F. Codd, in year 1970 disclose new concept of data representation. Mr. Codd suggested that all data in a database could be represented as a tabular structure (tables with columns and rows, which he called relations) and that these relations could be accessed using a high-level nonprocedural language. This research was result of several Relational DBMS like Oracle, Informix, Ingres and DB2. The following was high lights related to evolution database:

- **1980s:** The several vendors had developed OODBMSs like Object Design, Versant, O2 and Objectivity. The OODBMSs were no threat in the late 1980s to the now big commercial vendors developing and selling hierarchical, network or relational databases.
- **1990s:** In 1990s. The Object Database Management Group was founded, mainly &thanks to Rick Cattell of JavaSoft. The Green Team started the development of a new programming language which was loosely based on C++.

  The language was named Oak after the trees outside the office window of the language designer - James Gosling.
- **1993s to till date:** In 1993 several vendors of OODBMSs agreed upon an OODBMS standard called ODMG-93. The relational databases already had its standard-SQL-92, defined by its ANSI committee and ISO. The concept of internet, xml and other database management system was evolved in the time span. The detailed summery is represented in table below:

Generation	Time	Model	Examples	Comments
First	1960s- 1970s	File . system	VMS/VSAM	Used mainly on IBM mainframe system Managed records, not relationships

Second	1970s	Hierarchical and network IDS --II	IMS ADABAS	Early database systems. Navigational access
Third	Mid-1970s to present	Relational	DB2 Oracle MS SQL-Server	Conceptual simplicity Entity relationship (ER) modeling support for relational data modeling
Fourth	Mid-1980s to present	Object oriented Extended Relational	Versant VFS/Fast Objects Objectivity/DB	Support complex data Extended relational products support objects and data warehousing Web databases become common
Next Generation	Present to future	XML	dbXML Tamino DB2 UDB Oracle 10gMS SQL Server	Organization and management of unstructured data Relational and object modelsaddsupport for XML documents

Table 3.1: A brief summary of how the major data models where developed

3.1 RDBMS

1. RDBMS stands for "Relational Database Management System[9]',
2. ***'"RDBMS is a DBMS in which data is stored in the formthe form of tables and the relationship among the data is also stored in the form of tables."***
3. RDBMS also provide relational operators to manipulate the data stored into the database tables.
4. It is based on the relational model and was introduced by E.F. Codd.
5. E.F. Codd, the famous mathematician has introduced 12 rules (known as Codd's rules) toassist a database product to qualify a RDBMS. - .
6. RDBMSproduct has to satisfy at least 6 of the 12 rules, of Codd to be accepted as a full-fledged RDBMS.
7. Examples of RDBMS are: Oracle, Sybase, SQL- Server.
8. In short, all the information in RDBMS should be presented in tabular form and it follows Codd's rules.

3.2 E.F. CODD'S RULES

1. E.R Codd the famous mathematician has introduced 12 rules for the relational model for databases commonly known as Codd'srules.
2. These rules define what is required for a DBMS to be considered RDBMS.

3. The Codd's rules are as follows:

(a) **InformationRule:**Every information in RDBMS is represented in the form of tables.

(b) **Guaranteed Access Rule:**Every information in RDBMS is accessed by usingcombination of table name and primary key. A primary key helps to identify a rowname and column name.

(c) **Systematic Treatment of Null Values:** RDBMS supports null valuesfor representingmissingorInapplicable information.

(d) **The Description Rule:**The database description is represented at the logicallevel in the same way as ordinary data. The authorized users can apply the samerelational language for its manipulation as they apply to the regular data.

(e) **The Comprehensive Data Sublanguage Rule:** RDBMS supports many languages which allow users to define tables, query and update the data and set integrity constraints.

(f) **The View Updating Rule:**All the viewsthat are theoretically updatable must be updatable by the system.

(g) **High Level Insert, Update and Delete:** The system must support insert, update and delete operations.

(h) **Physical Data Independence:**It is the ability to change the internal schema (PhysicalLevel) without affecting the conceptual/logical or external schema.

(i) **Logical Data Independence:**Logical data independence is more difficult to achievethan the physical data independence. It is the ability to change the conceptual/logical schema without affecting the external schema.

(j) **Integrity Independence:**Integrity constraints should be specified separately fromapplication programs and stored in the catalog. Integrity constraints can be changedwithout affecting the application programs.

(k) **Distribution Independence:**User should not have to be aware of whether a databaseis distributedat different sites or not.

(l) **TheNon-SubversionRule:** If the RDBMS has a language that accesses the information of a record at a time, this language should not be used to bypass the integrityconstraints. .

3.3 COMPARISON BETWEEN DBMS AND RDBMS

DBMS	RDBMS
1. It stands for "Database Management System."	It stands for "Relational Database Management System."
2. It can store data in any format (graph, table, tree etc.)	It can store data only in tabular form.
3. It does not support client/server architecture.	It supports client/server architecture.
4. It does not satisfy Codd's rules.	It satisfyCodd's rules.
5. It requires low software and hardware requirements.	It requires high software and hardware requirements.
6. It can maintain only single user at a time. It supports single user.	It can maintain many users at a time. It supports multi-user.
7. It is designed for small organizations with small amount of data, where security of data is not a major issue.	It is designed for large organization with large amount of data where security of data is a major issue.
8. It does not support referential constraints.	It supports referential integrity constraints.
9. Examples of DBMS: Dbase, Foxpro	Examples of RDBMS: Oracle, Sybase, SQL-Server.

3.4 DATA MODEL

1. A model is a representation of reality, 'real world' objects and events,and their association.
2. A data model represents the organization itself.
3. Data model can be defined as an integrated collection of concepts for describing and manipulating data, relationships between data, and constraints on the data in an organization.
4. The purpose of a data model is to represent data and to make the data understandable.

Objectives of Data Model

- The main objective of database system is to highlight only the essential features and to hide the storage and data organization details from the user.
- A database model provides the necessary means to achieve data abstraction.
- A data, model is an abstract model that describes how the data is represented and used.
- A data model consists of a set of data structures and conceptual tools that is used todescribe the structure (data types, relationships, and constraints) of a database.
- A data model not only describes the structure of the data, it also defines a set of operations that can be performed on the data.
- A data model generally consists of data model theory, which is a formal description of howdata may be structured and used.
- The process of applying a data model theory to create a data model instance is known asdata modeling.

3.5 CLASSIFICATION OF DATA MODEL

Data Model is a collection of concepts to provide abstraction into DBMS so that superfluous details can be hide while highlighting important detail of data entities. It defines the logical design of data and establishes relationship between them. Data representation provides mechanisms to structure data for entities being modeled and allow a set of operations to be performed on them. A number of Models has been developed which are further categorized as below

1. Object based Logical Model
2. Record based Logical Model
3. Physical Data Model

Depending on the concept they use to model the structure of the database, the data models are categorized. The following logical tree display detailed classification:

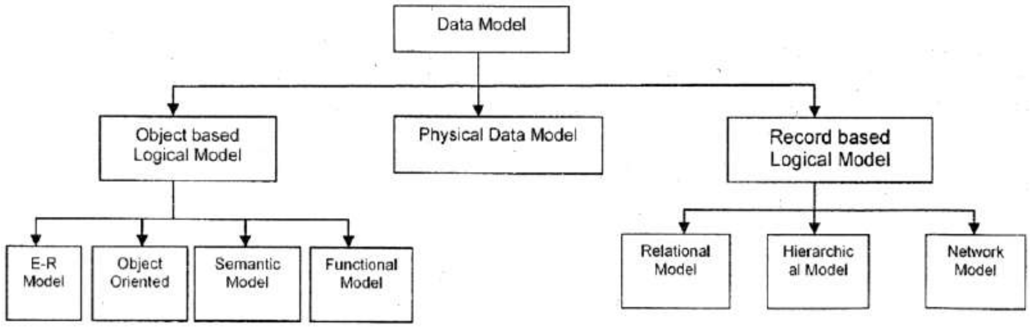


Fig. 3.1: Classification of Data Model

3.6 RECORD BASED MODELS

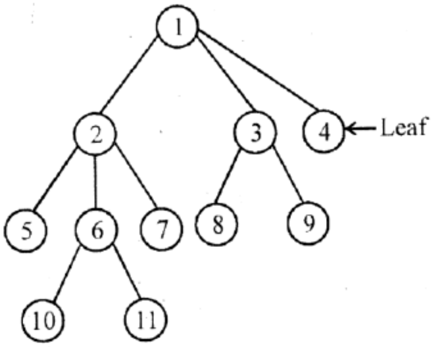
A record-based data models are used to specify the overall logical structures of the database. This model is used describing data at logical and view level. In the record based models, the database consists of a number of fixed-format records possibly of different types. Each record type defines a fixed number of fields, each typically of a fixed length. Data integrity constraints cannot be explicitly specified using record-based data models. There are three principle types of record-based data models:

3.6.1 Hierarchical Model

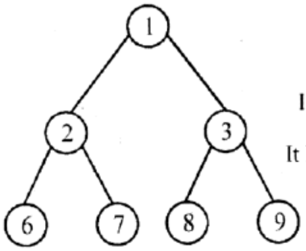
1. It was developed jointly by North American Rokwell Company and IBM.
2. It is the oldest model.
3. It follows tree as its basic structure.
4. Node at highest level is called Root.
5. A node may have any no. of children but each child node has only one parent.
6. Children of same parents are called siblings.

7. Anode that has no child is leaf node.

8. Representation of hierarchical model with suitable diagram

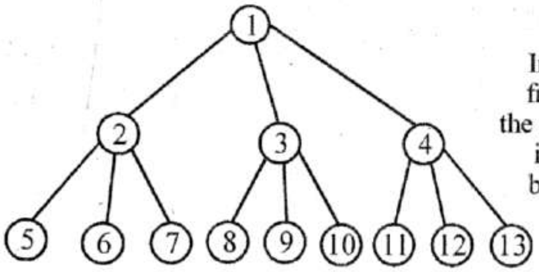


Leaf node is that node which has further no child. Here 5,10,11,7,8,9 & 4 are leaf nodes.



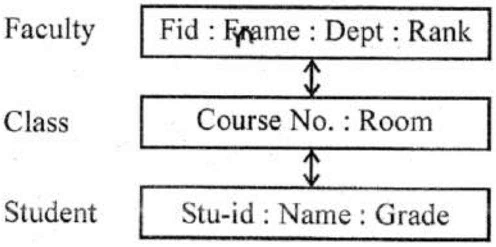
In binary tree, each node has not more than two child. It has exactly 2 child or 0 child.

Binary Tree

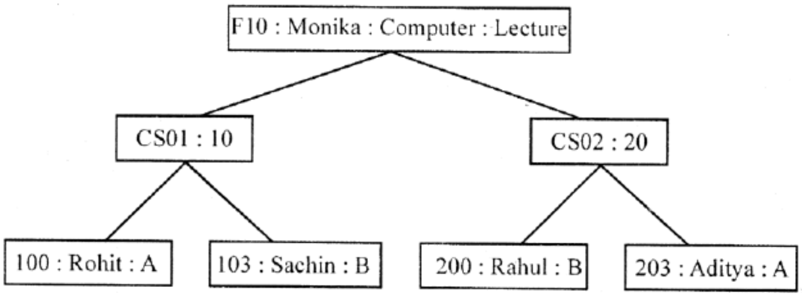


In balanced tree, every part from root node to a leaf has the same length. A balanced tree is always a binary tree but binary tree may or may not be a balanced tree.

Balanced Tree



- Faculty is on root node containing four attributes (Fid, Fname, Dept, Rank)
- Class is child of faculty node contains (Course no., room) attributes.
- There is one to many relationship between each faculty record and its class record.
- There is one to many relationship between each class record and its student record.



Operations on Hierarchical Model

(a) **Insertion:** A new class says CSO3 cannot be inserted unless some faculty is available at root level because without parent we can't insert any child node. This operation is used to insert a new record into the database. There are two possibilities:

(i) If the inserted, record is a root record then it creates new tree with the new record asthe root

(ii) If the inserted record is a child record, then we need to determine its parent first because no child record can exist without a parent record. So, insertion problem exists for the children who have no parents.

(b) **Deletion:**If we want to remove the class 100 then student Rohit will also have to be removed. This operation is used to delete a record from the database. To delete a record, we must first make it the current record of the database and then delete it. Here also, there are two possibilities:

(i) If the deleted record has no child node. It can be deleted easily.

(ii) If the deleted record has one or more child nodes, then the deletion process will deleteall the child nodes also. This may lead to loss of Information also.

(c) **Update:** If we want to updated the room 10, then we have to find all the recordsrelated to room 10 and have to modify. This operation is used to update a record. There are twopossibilities:

(i) If the record to be updated is a parent record, then updating it requires only one updation operation to be performed because there is only one occurrence of a parent record.

(ii) If the record to be updated is a child record, multiple updations may be required. If itnot happens, this may lead to inconsistency in the database.

(d) Record Retrieval: Recordretrieval methods for hierarchical model are complex and asymmetric. Retrieval means first searching the required record and then fetching it. Retrievalinvolves pointers from the parent node to the-child node in the tree and hence is complex andtime consuming.

Advantages of Hierarchical Model

1. **Simplicity:**The relationship between the various layers is logically simple. Thus the designof a hierarchical database is simple.
2. **Data Security:** Hierarchical model was the first database model that offered the data security that is provided and enforced by the DBMS.
3. **Data Integrity:** Hierarchical model is based on the parent/child there is always a link between the parent segment and the child segments under it. Thechild segments arealways automatically referred by its parents, so this model promotes data integrity.
4. **Efficiency:**The hierarchical database model is a very efficient one when the users require large number of transactions, using data whose relationships are fixed.

Disadvantages of Hierarchical Model

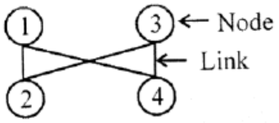
1. **Implementation Complexity:** Although the hierarchical databasemodel is conceptuallysimple and easy to design, it is quite complex to implement, the database designers shouldhave very good knowledge of the physical data storage characteristics.
2. **Lack ofStructural Independence:**Structural independence exists when the changes to the database structure does not affect the DBMS's ability to access data. Thus in a hierarchical database the benefits of data independence is limited by structural dependence.
3. **Programs Complexity:** Due to the structural dependence and the navigational structural, the application programs and the end users must know precisely how the data is distributed physically in the database in order to access data. This

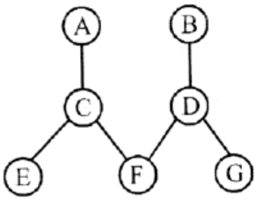
requires knowledge of complex painter systems, which is often beyond the grasp of ordinary users.

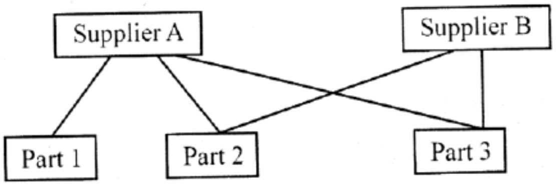
4. **Operational Limitations:** Hierarchical model suffers from the Insert anomalies, anomalies and deletion anomalies, also the retrieval operation is complex and asymmetric, thus hierarchical model is not suitable for all the cases.
5. **Implementation Limitations:** Many of the common relationships do not confirm to the 1'N format required by the hierarchical model. The many-to-many (N;N) relationships, which are more common in real life are very difficult to implement in a hierarchical model.

3.6.2 Network Model

1. Data in this model is represented by Links.
2. It looks like a tree structure containing nodes.
3. Every node may have one or more than one parent node.
4. Dependent node is called child node.

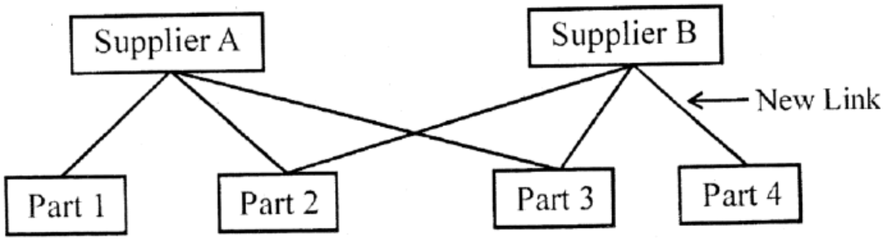




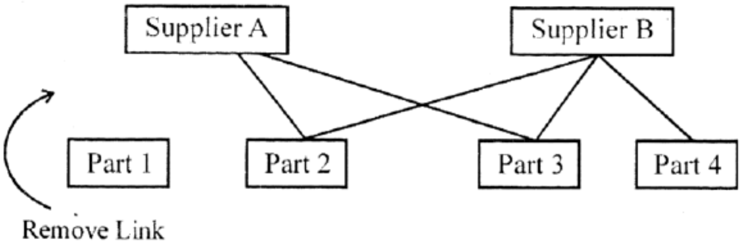


Operations on Network Model

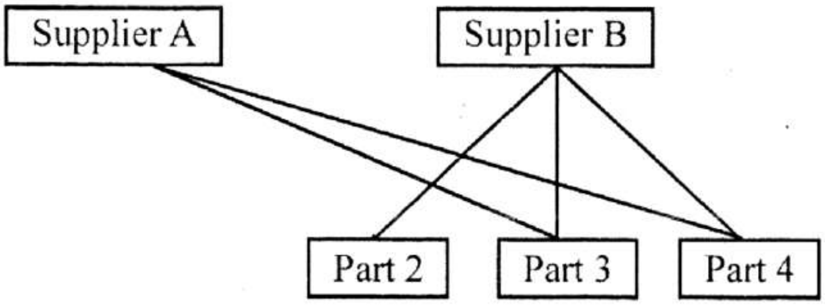
(a) **Insert Operation:** Insertion is easy i.e. supplier B supplies new part then we have to create a new link only. No other updation is required.



(b) **Delete Operation**: If we want to delete the information of any part, say supplier A doesn't want to supply part 1 now, so we have to remove only the link.



(c) **Updation Operation:** Updation is also easy. Suppose supplier A doesn't supply part 2 it supplies part 4 now.



(d) **Retrieval Operation:** Record retrieval method for network model aresymmetricbut complex.

Advantages of Network Model

1. **Conceptual Simplicity:** Like hierarchical model; the network model is also conceptually simple and easy to design.
2. **Capability to handle mass relationship types:** The network model can handle the one-to-many (1;N) and many to many (N:N) relationships, which is a real help in modeling the real life situations.

3. **Ease of data access:** The data access is easier than and flexible than the hierarchical model.
4. **Data Integrity:** The network model does not allow a member to exist without an owner. Thus a user must first define the owner record and then the member record. This ensures the data integrity.
5. **Data Independence:** The network model is better than the hierarchical model in isolating the programs from the complex physical storage details.
6. **Database Standard:** One of the major drawbacks of the hierarchical model was the nonavailability of universal standards for database design and modeling.

All the network database management systems conformed to these standards. These standards included a Data Definition Language [DDL] and the Data Manipulation Language [DML]$_9$ thus quality enhancing database administration and portability.

Disadvantages of Network Model

The network database model was significantly better than the hierarchical database model, it also had many drawbacks. These are

1. **System Complexity:** All the records are maintained using pointers and hence the whole database structure becomes very complex.
2. **Operational Anomalies:** Network model's insertion, deletion and updating operations of any record require large number of pointer adjustments, which makes its implementation very complex and complicated.
3. **Absence of Structural Independence:** If changes are mode to the database structure then all the application programs need to be modified before they can access data. Thus, every though the network database model succeeds in achieving data independence, it still fails to achieve structural independence.

Note: *We can conclude that network model does not suffers from the Insert anomalies, Update anomalies and Deletion anomalies. The retrieval operation is symmetric, as compared to hierarchical model, but the main disadvantage is complexity of the model.*

Hierarchical	Network

Each child node have only one parent.	Each child node may have more than one parent.
Hierarchical model records are organized as collection of trees.	Network model they are represented as arbitrary graphs.

3.6.3 Relational Model

1. It is primary data model for commercial data processing.The relational model was proposed by E.F.Codd of the IBM in 1972.
2. Relational model is a collection of tables. Tables are also known as relations.Therefore it is known as relational model.
3. Relational model represents the database as a collection of relations. Each relation (table)is a collection of row and columns.
4. Each table has a unique, name in database.
5. Columns are called attributes and rows are called tuples.
6. For each attribute there is a set of permitted values called domain.
7. Attribute name will be unique in a table.
8. Domain value can be NULL which shows that the value is unknown or does not exist.
9. The order of attribute has no significance. We can arrange attributes in any order.
10. We can insert record in any order.
11. Representation of data in Relational Model: A relational database consists of any number of relations. We can represent relation schemes by giving the name of the relation, followed by the attribute names in parenthesis.

Note: We will study Relational Model in detail in Unit-4

3.7 PHYSICAL MODEL

Physical model describes the in terms of a collection of files, indices, and other storage structures such as record formats, record ordering, and access paths. This model specifies how the database will be executed in a particular DBMS software such as Oracle, Sybase, etc., by taking into account the facilities and constraints of a given database management system. It also describes how the is stored on disk. Physical models are used for a higher-level description of storage structure and access mechanism.

They describe how data is stored in thecomputer, representinginformation such as record structures, record orderings and access paths. It is possible to implement the database at system level using physical data models. There are not as many physical data models so far. The most common physical data models are as follows:

- Unifying model
- Frame memory model.

3.8 OBJECT BASED MODELS

The object based models use the concepts of entities or objects and relationships among them. An entity is a distinct object (a person, place, concept, and event) in the organization that is to be represented In the database. An attribute is a property that describes some aspect of the object that we wish to recordand a relationship is an association between 'entitles. It providesflexible structuring capabilities and allows data constraints to be specified explicitly. The object based logical model are classified as follows:

1. E-RModel
2. Object Oriented Model
3. Semantic Model
4. Functional Model

3.8.1 E-R Model (Entity Relationship Model)

E-R model is an effective and standard method of communication amongst different designers, programmers and end-users who tend to view data and its use in different ways. It is a non- technical method, which is free from ambiguities and provides a standard and a logical way of visualizing the data. It gives precise understanding of the nature of the data and how it is used by the enterprise. It provides useful concepts that allow the database designers to move from an informal description of what users want from their database, to a more detailed and precise description that can be implemented in a database management system. Thus, E-R modeling is an important technique for any database designer to master. It has found wide acceptance in database design. A basic concept of E-R model has been introduced and few examples of E-R diagram of an enterprise database have been illustrated. The ER model is based on a concept of a real world entities and relationships among these entities. It can be used developed database design by allowing specification schema, which represents the overall logical structure of a database. It is very useful in mapping the meanings and interactions of real-world entities onto a conceptual schema.

1. E-R model is based on real world. It is a collection of basic objects, called entities and of relationships among these objects (Entity).
2. E-R model employs three basic features:
   - Entity
   - Attributes
   - Relationship
3. The overall logical structure of a database can be expressed graphically by an E-R diagram, which is built up by the following components:
   - Rectangle, which represent entity sets.
   - Ellipses, which represent attributes.
   - Diamonds, which represent relationships among entity sets.
   - Lines, which link attributes to entity sets and entity sets to relationships.

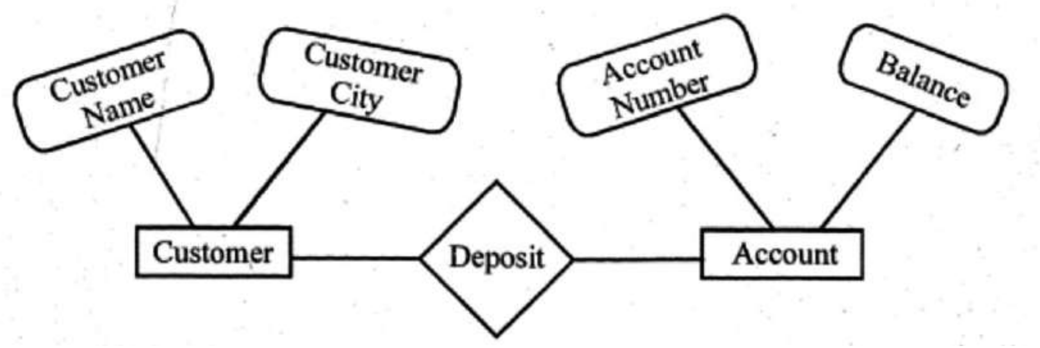


Fig. 3.2: E-R Model

Entity: Customer and Account are entity.

Attribute: Customer name, customer city are attributes of customer entity. Account number, balance are attributes of Account Entity.

Relation: Deposit is relationship among customer and Account.

I. ENTITY

- Entity is a thing which can be identified.
- Entity is a person, place thing event or concept which can be identified. We can say about which we want to store information i.e. employee, student, customer.
- □ Rectangle sign is used to represent the entity in E-R diagram.
- Entity can be of two types which are as follows:

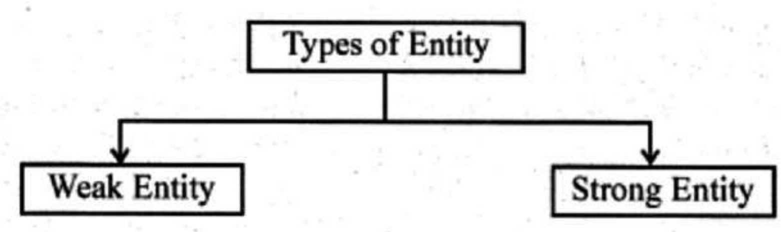
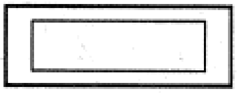
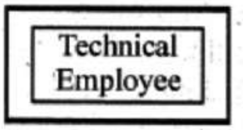


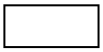
Fig. 3.3: Types of Entity

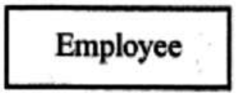
(a) Weak Entity:

- Weak entity depends on some other entity.
- It can't exist if other entity on which is depends does not exist.
- It is represented by 
- For example:



(b) Regular Entity/Strong Entity:

- Regular entity does not dependent on other entity.
- Its existence doesn't depend upon any other entity.
- It is represented by 
- For example:



(c) **Entity Set:** It is a set of entities of the same type that share the same properties (attributes) i.e. the set of all persons who are customer at same bank.

(d) **Entity Subtype and Super type:** Entity can be sub or super i.e. we have an Entity employee and employee can be programmer or operation.

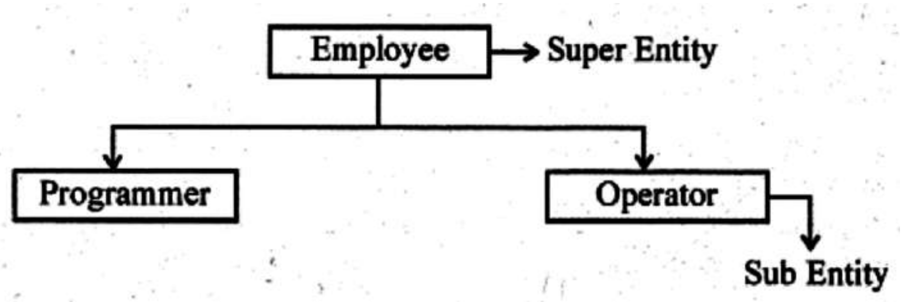


Fig. 3.4. Super Entity and Sub Entity

II. ATTRIBUTE

- Attributes are properties of entity. They are also called columns.
- Entity is about which we want to store information and attribute is what information we want to store.
- For example: If we want to store Name, City, and Salary information about employee. Then employee is our entity and name, city, salary are attributes.
- ⬭ Ellipse sign is used to represent attributes.
- Attribute can be of four types which are as follows:

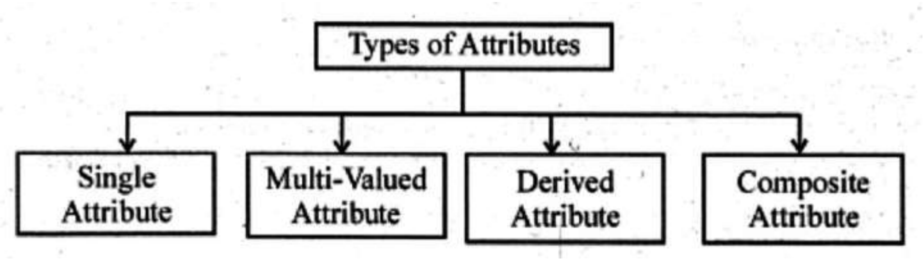
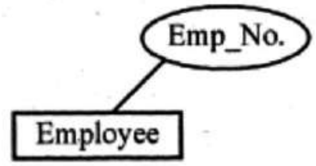
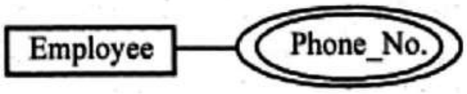


Fig. 3.5 :Types of Attribues

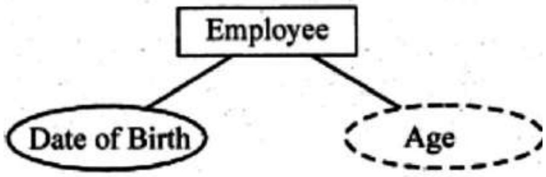
(a) **Single Attribute:** Single attributes are those attributes which can't be divided into sub parts i.e. Employee number is a simple attribute.



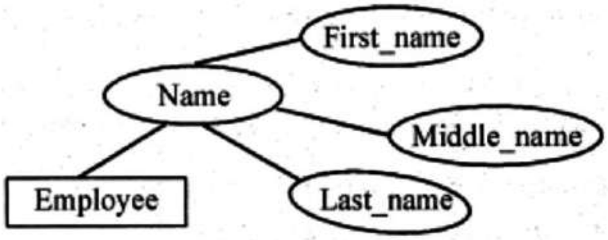
(b) **Multi Valued Attribute:** Multi valued attributes are those attribute which have more than one value i.e. Phone number is a multi-valued attribute. One employee may have more than one phone number.



(c) **Derived Attribute:** Derived attributes are those attributes whose value is derived from another attribute. For example: value of age attribute can be drive (calculate) from date of birth attribute and current date.



(d) **Composite Attribute:** Composite attributes are those attributes which can be divided into parts. For example: Name attribute can be divided into First Name, Middle Name and Last Name. Address attribute can be divided into street, city, state, zip-code.



III. RELATIONSHIP

- It is used to connect the entities.

- The entities involved in given relationship are called participants.
- The no. of participants in a given relationship is called degree of *relationship*.
- ◇ sign is used to represent relationship among entities.
- Deposit is a relationship among entity customer and entity account.
- Relationship can be of four types which are as follows:

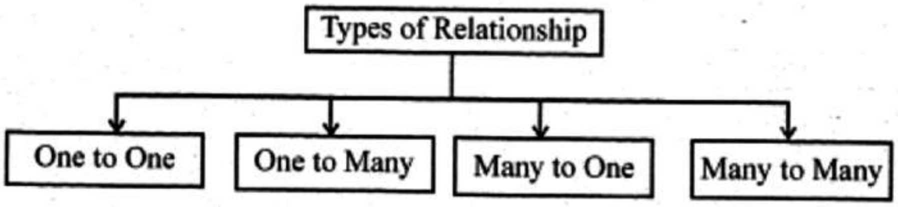
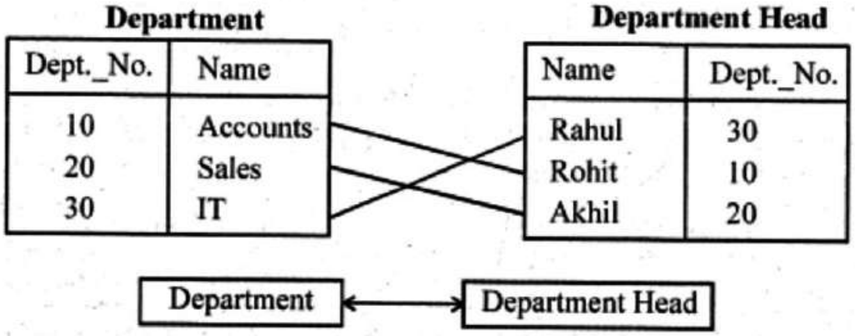


Fig. 4.5: Types of Relationship

(a) One to One Relationship: In one to one relationship for one record in entity A, there is exactly one record in entity B. For example: we have two entities department and department head. There is one to one relationship because one department will be under one head and one head will be appointed for one department.



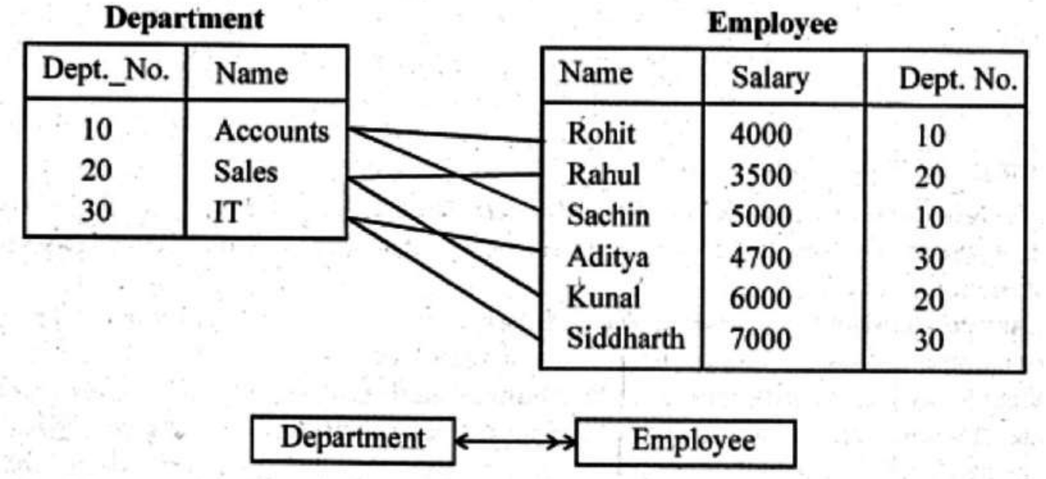
Department

Dept._No.	Name
10	Accounts
20	Sales
30	IT

Department Head

Name	Dept._No.
Rahul	30
Rohit	10
Akhil	20

(b) One to Many Relationships: In one to many relationships for one record in entity A, there is more than one record in entity B. For example: We have two entities department and employee. There is one to many relationships because there will be one department in a company and more than one employee will work in that particular department.



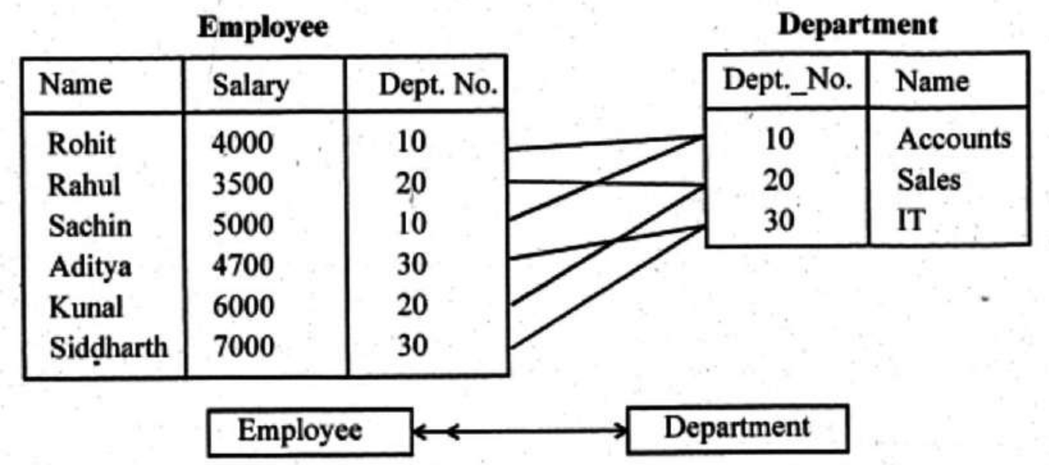
Department

Dept._No.	Name
10	Accounts
20	Sales
30	IT

Employee

Name	Salary	Dept. No.
Rohit	4000	10
Rahul	3500	20
Sachin	5000	10
Aditya	4700	30
Kunal	6000	20
Siddharth	7000	30

(c) Many to One Relationship: In many to one relationship, for many records in entity A, there is only one record in entity B. For example: We have two entities employee and department. There is many to one relationship because there will be many employees in a single



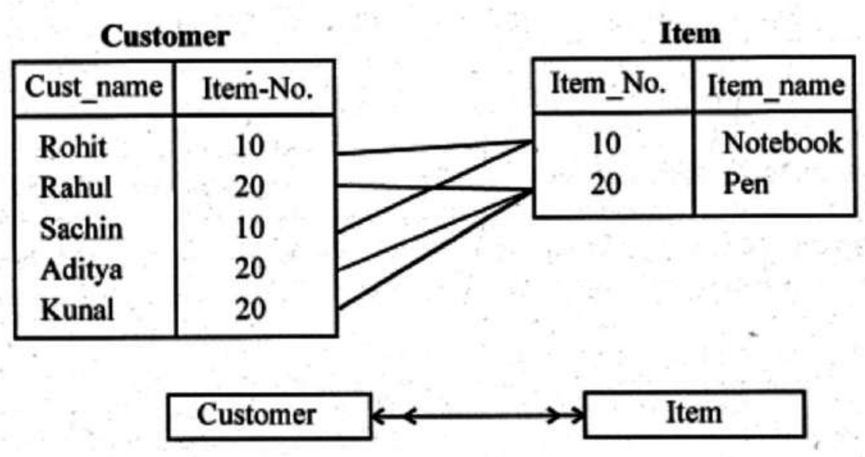
Employee

Name	Salary	Dept. No.
Rohit	4000	10
Rahul	3500	20
Sachin	5000	10
Aditya	4700	30
Kunal	6000	20
Siddharth	7000	30

Department

Dept._No.	Name
10	Accounts
20	Sales
30	IT

(d) Many to Many Relationships: In many to many relationships, for many record is an entity A, there will be many record in entity B. There is many to many relationship because there will be many customers for many items.



Customer

Cust_name	Item-No.
Rohit	10
Rahul	20
Sachin	10
Aditya	20
Kunal	20

Item

Item_No.	Item_name
10	Notebook
20	Pen

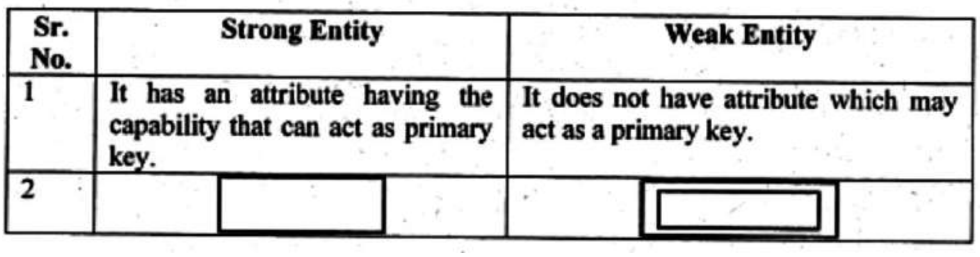
Advantages of E-R Model

1. **Straight forward relation representation:** Having designed an E-R diagram for a database application, the relational representation of database model becomes relatively straight forward.
2. **Easy conversion for E-R over Data Model:** Conversion from E-R diagram to network or hierarchical data model can easily be accomplished.
3. **Graphical Representation for better understanding:** An E-R model gives graphical and diagrammatical representation of various entities, its attributes and relationship between entities. This helps in understanding the data structure in easy way, minimize the redundancy and other problem.

Disadvantages of E-R Model

1. Popular for high-level design: It is especially popular for high level design.
2. No Industry standard of Notation.

Difference between Strong Entity and Weak Entity



Sr. No.	Strong Entity	Weak Entity
1	It has an attribute having the capability that can act as primary key.	It does not have attribute which may act as a primary key.
2	▭	⧈

3.8.2 Object Oriented Model

The object oriented data model is an. adaptation of the object oriented programming language paradigm to database systems. The model is based on the concept of encapsulating data and code that operates on that data in an object. On the other hand, the object-relational data model is an., extension of relational data model. It combines the features of both the relational data model and object-oriented data model.

Object oriented data models for databases "extend the above mentioned data modeling features of the object oriented paradigm The extensions include data integrity constraints, persistence of data which allows transient data to be distinguished from persistent data and support for collections.

Advantages of Object-Oriented Data Model

1. **Capable of handling a large variety of data types:** hierarchical, network or relational), the object-oriented database are capable of storingdifferent types of data, for example, pictures, voices, video, including text, numbers and soon.
2. **Combining object-oriented programming with database technology:**Object-orienteddata model is capable of combining object-oriented programming with database technologyand thus, providing an Integrated application development system.
3. **Improved productivity:** Object-oriented data models provide powerful features such asinheritance, polymorphism and dynamic binding that allow the users to compose objects and provide solutions without writing object-specific code. These features increase-theproductivity of the database application developers significantly.
4. **Improved data access:** Object-oriented data model represents relationships explicitly,supporting both navigationaland associative access to information. It further improves the data access performance over relational-value-based relationships.

Disadvantages of Object-Oriented Data Model

1. **No precise definition:** It is difficult to provide a precise definition of what constitutes an object-oriented DBMS because the name has been applied to a variety of products and prototypes, some of which differ considerably from one another.
2. **Difficult to maintain:** The definition of objects is required to be changed periodically and migration of existing databases to confirm to the new object definition with change in organisational information needs. It possess real challenge when changing object definitions and migrating databases.
3. **Not suited for all applications:** Object-oriented data models are used where there is a need to manage complex relationships among data objects. They are especially suited for specific applications such as engineering, e-commerce, medicines and so on, and not for all applications. Its performance degrades and requires high processing requirements when used for ordinary applications.

3.8.3 Semantic Model

This model is used to express greater interdependencies among entities of interest. These independencies enable the models to represent the semantics of the data in the database. The Semantic Data Model (SDM), like other data models, is a way of structuring data to represent it in a logical way. SDM differs from other data models in that it focuses on providing more meaning of the data itself, rather than only on the relationships and attributes of the data.

SDM provides a high-level understanding of the data by abstracting it further away from the physical aspects of data storage.

In SDM, an entity represents some aspect or item in the real world, such as a student An entity is similar to a record in a relational system or an object in an object-oriented system. These entities in SDM focus on types, which are more general, instead of sets of data. In SDM, an entity is a very basic notion of a real-world or conceptual object that is defined by a single attribute.

For instance, an SDM entity type might be *person* which would be a list of names of people that are to be represented by the data. The objects in this domain would then point to specific instances of a person that are represented by each person entity.

3.8.4 Functional Model

The functional data model describes those aspects of a system concerned with transformations of values-functions, mappings, constraints and functional dependencies. The functional data model describesthe computations within a system.

- It shows how output value is derived from input values without regard for the orderwhich the values are computed. It also includes constraints among values.
- It consists of multiple data flow diagrams.
- Data flow diagrams show the dependencies between values and computation of output values from input values and functions, without regard for when the functions are executed.
- Traditional computation concepts such as expression trees are examples of functional models.

3.9 COMPARISON OF DATA MODELS

Sr. No.	Hierarchical Model	Network Data Model	Relational Data Model
1.	Hierarchical data model represents data in a tree format where Parent and Child relationship is represented to show association.	Network model represents data in graphs where data is a record which is linked by pointers.	Relational data model logically represents data in Tabular form where data is placed in row and column.
2.	Many to many relationship cannot be expressed in hierarchical model	Many to many relationship can be expressed in hierarchical model.	Many to many relationship can be expressed in hierarchical model.
3.	It is good for expressing data in < parent child relationship	It is good for modelling of many to many relationship.	It is good for modelling real world entities.

4	Relationship are represented by pointer and relationship among records are physical in nature	Network model also represents relationship through pointers and nature of the relationship is physical.	Relational model is stored data in form of rows and column. There is no physical connection is established between different tables whereas connection is logical in nature and established through keys.
5	Searching of a particular record is a time consuming task as to reach a particular child we have to process through its parent record.	Searching of a particular record is easy since there are multiple access path available to reach a node in graph.	In case of relations tables we use concept of keys to identify the records and search a key through indexing is quite simple task
6.	Insertion is done in the form of parent node and child node relationship. We cannot insert child node in tree without parent node.	Network model insertion can be performed by inserting new node in the graph with ease and has no insertion anomaly	in Relation model , new record can be added any time and has no insertion anomaly
7.	Updation operation may results in inconsistency as there are multiple child records in a tree	Updation operation is free from any anomaly as there is only single occurrence of each record in a graph which may be connected with multiple records.	Updation operation is safe in a relational model as duplication of record is avoidable by applying normalisation and Primary keys relationships
8.	Hierarchical model is based on parent child relationship and deleting of child is easy as compare to parent, if we delete parent then child node will automatically deleted from the tree.	There is no deletion anomaly as deleting of one node does not affect other nodes due to many to many relationships.	The deleting of record from a relation is again a simple process and there is no anomaly related to deleting of records. Deletion of reference records is not allowed as it may linked to other records

3.10. OTHER TERMS USED IN E-R MODEL

CONSTRAINTS

Relationship types usually have certain constraints that limit the possible combinations of entities that may participate in the corresponding relationship set. The constraints should reflect the restrictions on the relationships as perceived in the 'real world'. For example, there could be a requirement that each department in the entity DEPT must have a person and each person in the PERSON entity must have a skill. The main types of constraints on relationships are multiplicity, cardinality, participation and so on.

1. **Multiplicity Constraints:** Multiplicity is the number (or range) of possible occurrences of an entity type that may relate to a single occurrence of an associated entity type through a particular relationship. It constrains the way that entities are related. It is a representation of the policies and business rules established by the enterprise or the user. It is important that all appropriate enterprise constraints are identified and represented while modeling an enterprise.

2. **Cardinality Constraints:** A cardinality constraint specifies the number of instances of one entity that can (or must) be associated with each instance of entity. There are two types of cardinality constraints namely minimum and maximum cardinality constraints. The minimum cardinality constraint of a relationship is the minimum number of instances of an entity that may be associated with each instance of another entity. The maximum cardinality constraint of a relationship is the maximum number of instances of one entity that may be associated with a single occurrence of another entity.

3. **Participation Constraints:** The participation constraint specifies whether the existence of an entity depends on its being related to another entity via the relationship type. There are two types of participation constraints namely total and partial participation constraints. Total participation constraints means that every entity in 'the total set' of an entity must be related to another entity via a relationship. Total participation is also called existence dependency. A partial participation constraint means that some or the 'part of the set of an entity are related to another entity via a relationship, but not necessarily all. The cardinality ratio and participation constraints are together known as the structural constraints of a relationship type.

4. **Exclusion and Uniqueness Constraints:** E-R modeling has also constraints such exclusion constraint and uniqueness constraint that results into poor semantic base and tries to make entity-attribute decisions early in the conceptual

modeling process. In exclusion constraint the normal or default treatment of multiple relationships is inclusive OR, which allows any or all of the entities to participate. In some situations, however, multiple relationships may be affected by the exclusive (disjoint or exclusive OR) constraint, which allows at most one entity instance among several entity types to participate in the relationship with a single root entity.

GENERALIZATION

1. A generalization hierarchy is a form of abstraction that specify the two or more entities that share the common attributes can be generalized into a higher-level entity type called a super type or generic entity.
2. The lower level of entities becomes the subtype. Subtypes are dependent entities.

SPECIALIZATION

1. Specialization is a process of taking subsets of higher level entity set to form lower level entity sets.
2. It is a process of defining a set of subclasses of an entity type which is called as super class of the specialization.

AGGREGATION

1. One limitation of the E-R model is that it cannot express relationship among relationships.
2. Aggregation is the process of compiling information on an object, thereby abstracting a higher level object.
3. Aggregation allows us to indicate that a relationship set participate in another relationship set.

Questions

1. What do you mean[1] by data models? Explain the answer.
2. How can we classify data models?
3. What do mean by relationships in a data model
4. What is an attribute in data modeling?

5. Explain the Relational Model? Write advantages and disadvantages.

6. Explain the Hierarchical Model?Write advantages and disadvantages.

7. Explain different operations that can be performed on Hierarchical

8. Compare different data models.

9 Define the following terms:

   (a) Entity Set

   (b) Attribute

   (c) Relationship Set

   (d) Simple attributes

   (e) Composite attributes

   (f) Multivalve attributes.

10 What are the different types of attributes? Explain using examples.

11 What are mapping constraints? What are its types?

12 What are weak entity sets? Why are they used?

13 What is generalization? Explain with a suitable example.

14 What is aggregation? Explain using a suitable example.

15 The E-R Diagram for an Employee Payroll System.

16 Discuss the advantages and disadvantages of ER model.

17 The E-R Diagram for Book Purchasing System..

18 Explain with diagrammatical illustrations about the different types of relationships

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 4 : RELATIONAL DATA MODEL

4.1 RELATIONAL MODEL

1. It is primary data model for commercial data processing.The relational model was proposed by E.F.Codd of the IBM in 1972.
2. Relational model is a collection of tables. Tables are also known as relations.Therefore it is known as relational model.
3. Relational model represents the database as a collection of relations. Each relation (table)is a collection of row and columns.
4. Each table has a unique, name in database.
5. Columns are called attributes and rows are called tuples.
6. For each attribute there is a set of permitted values called domain.
7. Attribute name will be unique in a table.
8. Domain value can be NULL which shows that the value is unknown or does not exist.
9. The order of attribute has no significance. We can arrange attributes in any order.
10. We can insert record in any order.
11. Representation of data in Relational Model: A relational database consists of any number of relations. We can represent relation schemes by giving the name of the relation, followed by the attribute names in parenthesis.

12. ***Components of Relational Model***

(1) Data Structure

(2) Data Integrity

(3) Integrity Constraints

1. **Data Structure**

(a) **Relation**

- All data is represented in table.
- Table contains rows and columns. Table isalso known as Relation.
- Columns are called attributed and rows are called tuples. Each cell of relation contains only single value.
- Relation contains information on one subject only.

Student

S. NO.	**Name**	**Class**	**City**	**Roll Number**
1.	Rohit	10th	Chennai	1010
2.	Akhil	12th	Mumbai	1229
3.	Aditya	8th	Bangalore	801
4.	Sachin	9th	Gurgaon	906
5.	Rahul	11th	Patiala	1112
6.	Sid	12th	Pune	1225
7.	Kunal	8th	Hyderabad	806

Student is the title of the relation. There are 5 column (S.No, Name, Class, City, Roll Number) and 7 rows.

(b) Attribute

- Columns-are called attributes.
- Attributes appear vertically in a relation.
- Attributes can appear In any order and provide specific information.
- Attributes.in a relation "Student" are S.No., Name, Class, City, Roll Number.

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10th	Chennai	1010
2.	Akhil	12th	Mumbai	1229
3.	Aditya	8th	Bangalore	801
4.	Sachin	9th	Gurgaon	906
5.	Rahul	11th	Patiala	1112
6.	Sid	12th	Pune	1225
7.	Kunal	8th	Hyderabad	806

(c) Tuple

- Rows are called tuples.
- Tuples appear horizontally in a relation.
- Tuples can appear in any order and provide complete information (full record).
- There are 7 tuples in a relation "Student".

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10th	Chennai	1010
2.	Akhil	12th	Mumbai	1229
3.	Aditya	8th	Bangalore	801
4.	Sachin	9th	Gurgaon	906
5.	Rahul	11th	Patiala	1112
6.	Sid	12th	Pune	1225
7.	Kunal	8th	Hyderabad	806

(d) Domain

- Domain is a set of all allowed or possible values for an attribute in a relation.

- Domain specifies the type of data used in an attribute.
- For example, in a relation student, students are from different cities. In this case, city is an attribute of relation student and Mumbai, Patiala, Gurgaon etc. are domain values of city attribute.

(e) Degree

- Degree is the number of attributes in a relation.
- If a relation have only one attribute, then its degree is one and known as called unary relation
- If a relation have two attributes, then its degree is two and known as called binary relation
- If a relation have three attributes, then its degree is three and known as called ternary and so on.
- Degree of relation "Student" is 5.

(f) Cardinality

- Cardinality is the number of tuples in a relation.
- Cardinality changes on the basis of insertions or deletions of records in a relation.
- Cardinality of relation "Student" is 7

2. Data Integrity: Data integrity ensures the accuracy of data. For this purpose, we should know about the keys.

"A key is a single attribute or a combination of two or more attributes of a relation. It is used to identify one or more instance of the set."

Types of Keys: There are six types of keys which are as follows:

(a) Candidate key:

- Candidate keys are those attributes which have unique values. But Null value is not allowed in a candidate key.
- If there is no attribute in a relation containing unique value then combination of two attributes of that relation can make candidate key.

- There can be any number of candidate keys in a relation (table).

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10th	Chennai	1010
2.	Akhil	12th	Mumbai	1229
3.	Aditya	8th	Bangalore	801
4.	Sachin	9th	Gurgaon	906
5.	Rahul	11th	Patiala	1112
6.	Sid	12th	Mumbai	1225
7.	Kunal	8th	Hyderabad	806
8.	Rohit	10th	Chennai	1011

In the above relation "Student", S. No. and Roll Number both attributes have unique value, therefore they are candidate keys.

(b) Primary Key

- The attribute which have unique value is known as primary key. But Null value is notallowed in a primary key.
- Primary key is used for query purposes.
- There will be only one primary keys in a relation (table).

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10th	Chennai	1010
2.	Akhil	12th	Mumbai	1229
3.	Aditya	8th	Bangalore	801
4.	Sachin	9th	Gurgaon	906
5.	Rahul	11th	Patiala	1112
6.	Sid	12th	Pune	1225
7.	Kunal	8th	Hyderabad	806
8.	Rohit	10th	Chennai	1011

In the above relation "Student", only one attribute Roll Number have unique value, therefore attribute Roll Number is a primary key.

(c) Alternate Key

- Alternate key also contains unique value.
- After identifying candidate keys, one key is known as primary key and another key (which is not selected as primary key) is known as alternate key.

Student

S. NO.	Name	Class	City	Roll Number
1.	Rohit	10th	Chennai	1010
2.	Akhil	12th	Mumbai	1229
3.	Aditya	8th	Bangalore	801
4.	Sachin	9th	Gurgaon	906
5.	Rahul	11th	Patiala	1112
6.	Sid	12th	Pune	1225
7.	Kunal	8th	Hyderabad	806
8.	Rohit	10th	Chennai	1011

In the above relation "Student", S. No. and Roll number both have unique values and called as candidate keys. S.No. is called alternate key and Roll number is known as primary key.

If Primary key: Roll Number

Then Alternate key: S. No.

(d) Composite Key

- Sometimes in a relation, there is no primary key. In that situation, more than one attributes are used to identify a unique entity.

- The combination of those attributes are known as composite key.

Name	Class	City	Age
Rohit	10th	Chennai	16
Akhil	12th	Mumbai	18
Aditya	8th	Bangalore	14
Sachin	9th	Gurgaon	15
Rahul	11th	Patiala	17
Sid	12th	Pune	18
Kunal	8th	Hyderabad	14
Rohit	10th	Chennai	17

In the above relation "Student”, Name and Age are used to identify an entity, therefore both are called composite key.

(e) Artificial Key: Sometimes in a relations, there is no primary key and there is no possibility to make primary key.In that situation, we can insert a key in a relation which has no meaning is known as an artificial key.

Student

Name	Class	City	Age
Rohit	10th	Chennai	16
Akhil	12th	Mumbai	18
Aditya	8th	Bangalore	14
Sachin	9th	Gurgaon	15
Rahul	11th	Patiala	17
Sid	12th	Pune	18
Kunal	8th	Hyderabad	14
Rohit	10th	Chennai	17

In the above relation "Student", there is no unique key. We can insert a new attribute S.No. into relation as a artificial key. This attribute S. No. has no meaning.

Student

S. No.	Name	Class	City	Age
1	Rohit	10th	Chennai	16
2	Akhil	12th	Mumbai	18
3	Aditya	8th	Bangalore	14
4	Sachin	9th	Gurgaon	15
5	Rahul	11th	Patiala	17
6	Sid	12th	Pune	18
7	Kunal	8th	Hyderabad	14
8	Rohit	10th	Chennai	17

(f) Foreign Key

- Foreign key is the attribute of a relation which acts as a primary key of another table.
- Foreign key .allows only those values which appears in primary key or may be null.
- Foreign key is used to make a relationship between two tables and to maintain thereferential integrity.

Class

Class	Class Incharge
8th	Mrs. Nidhi
9th	Ms. Aastha
10th	Mrs. Prathiba
11th	Mrs. Manmeet
12th	Mrs. Navreet

Student

S. No.	Name	Class	City	Roll Number
1	Rohit	10th	Chennai	1010
2	Akhil	12th	Mumbai	1229
3	Aditya	8th	Bangalore	801
4	Sachin	9th	Gurgaon	906
5	Rahul	11th	Patiala	1112
6	Sid	12th	Pune	1225
7	Kunal	8th	Hyderabad	806
				1011

In the above relation, class acts as an foreign key.

3. Relational Model Constraints/Integrity Constraints

- Integrity constraints ensure that changes made to the database by authorized users and any change do not lose the data.
- Integrity constraints also ensure the restrictions on the data and provide the security against the accidental damage to the database.

Type of Constraints

(a) Domain Constraint:

- It ensures that each attribute have a correct value.
- The data type associated with domains includes integer, character, string, data, and time.
- For example: A is not allowed in the attribute Roll Number because Roll Number is an integer attribute.

S. No.	Name	Class	City	Roll Number
1	Rohit	10th	Chennai	1010
2	Akhil	12th	Mumbai	1229
3	Aditya	8th	Bangalore	801
4	Sachin	9th	Gurgaon	906
5	Rahul	11th	Patiala	1112

6	Sid	12th	Pune	1225
7	Kunal	8th	Hyderabad	806
8	Rohit	10th	Chennai	1011

(b) Tuple Uniqueness Constraint

- Relation is a set of tuples (rows).
- All tuples in a relation must be different from each other. It means there must be unique value or attribute by which we can identify a tuple.

(c) Key Constraint

- Primary key must have unique value in the relation (table).
- If S.No. is considered as a primary key then there must be unique value in this attribute. We cannot insert duplicate value in the primary key.

S. No.	Name	Class	City	Roll Number
1	Rohit	10th	Chennai	1010
2	Akhil	12th	Mumbai	1229
3	Aditya	8th	Bangalore	801
4	Sachin	9th	Gurgaon	906
5	Rahul	11th	Patiala	1112
5/3	Sid	12th	Pune	1225
7	Kunal	8th	Hyderabad	806
7/4	Rohit	10th	Chennai	1011

(d) Entity Integrity: Entity integrity ensures that primary key cannot have NULL value.

S. No.	Name	Class	City	Roll Number
1	Rohit	10th	Chennai	1010
2	Akhil	12th	Mumbai	1229
3	Aditya	8th	Bangalore	801
4	Sachin	9th	Gurgaon	906
5	Rahul	11th	Patiala	1112

	Sid	12th	Pune	1225
7	Kunal	8th	Hyderabad	806
	Rohit	10th	Chennai	1011

(e) **Referential Integrity:** Referential integrity ensures that if a foreign key of a table I refers to the primary key of table II, then every value of the foreign key in table I must be null or be available in table II.

I	
Class	**Class Incharge**
8th	Mrs. Nidhi
9th	Ms. Aastha
10th	Mrs. Prathiba
11th	Mrs. Manmeet
12th	Mrs. Navreet

II				
S. No.	**Name**	**Class**	**City**	**Roll Number**
1	Rohit	10th	Chennai	1010
2	Akhil	12th	Mumbai	1229
3	Aditya	8th	Bangalore	801
4	Sachin	9th	Gurgaon	906
5	Rahul	11th	Patiala	1112
6	Sid	12th	Pune	1225
7	Kunal	8th	Hyderabad	806
8	Rohit	10th	Chennai	1011

Operations of Relational Model

1. **Insert Operation:** Relational model does not suffer from any insert anomaly.

2. **Update Operation:** Relational model does not suffer from any update anomaly.

3. **Delete Operation:**Relational model does not suffer from any delete anomaly.

4. **Retrieve Operation:** Retrieve operation for relational data model is simple and symmetric.

Advantages of Relational Data Model

1. **Simplicity:** A relational data model is even simpler than hierarchical and network models. It frees the designers from the actual physical data storage details, thereby allowing them to concentrate on the logical view of the database.

2. **Structural Independence:** Unlike hierarchical and network models, the relational data model does not depend on the navigational data access system. Changes in the database structure do not affect the data access. Ease of design, implementation, maintenance and uses: The relational model provides bothstructural independence and data independence. Therefore, itmakes the database design, implementation, maintenance and usage much easier.

3. **Flexible and Powerful Query Capability:** Its structured query capability makesad hoc queries a reality.The relational database model provides very powerful, flexible, and easy-to-use query facilities. Information in a table can be easily modified.

4. **Easy to Use:** To collect the information in table consisting columns and rows is very easy.

5. **Security:** In relational model, security control and authorization can be implemented.

Disadvantages of Relational Data Model

1. **Hardware overheads:** The relational data models need more powerful computing hardware and data storage devices to perform complex tasks. Consequently, they tend to be slower than the other database systems. However, with rapid advancement in computing technology and development of much more efficient operating systems, the disadvantage of being slow is getting faded.

2. **Easy-to-design capability leading to bad design:** Easy-to-use feature of relational database results into untrained people generating queries and reports

without much understanding and giving much thought to the need of proper database design. With the growth of database, the poor design results into slower system, degraded performance and data corruption.

4.2 COMPARISON OF DATA MODELS

Sr. No.	Hierarchical Model	Network Data Model	Relational Data Model
1.	Hierarchical data model represents data in a tree format where Parent and Child relationship is represented to show association.	Network model represents data in graphs where data is a record which is linked by pointers.	Relational data model logically represents data in Tabular form where data is placed in row and column.
2.	Many to many relationship cannot be expressed in hierarchical model	Many to many relationship can be expressed in hierarchical model.	Many to many relationship can be expressed in hierarchical model.
3.	It is good for expressing data in < parent child relationship	It is good for modeling of many to many relationships.	It is good for modeling real world entities.
4	Relationship are represented by pointer and relationship among records are physical in nature	Network model also represents relationship through pointers and nature of the relationship is physical.	Relational model is stored data in form of rows and column. There is no physical connection is established between different tables whereas connection is logical in nature and established through keys.
5	Searching of a particular record is a time consuming task as to reach a particular child we have to process through its parent record.	Searching of a particular record is easy since there are multiple access path available to reach a node in graph.	In case of relations tables we use concept of keys to identify the records and search a key through indexing is quite simple task
6.	Insertion is done in the form of parent node and child node relationship. We cannot insert child node in tree without parent node.	Network model insertion can be performed by inserting new node in the graph with ease and has no insertion anomaly	in Relation model , new record can be added any time and has no insertion anomaly

7.	Updation operation may results in inconsistency as there are multiple child records in a tree	Updation operation is free from any anomaly as there is only single occurrence of each record in a graph which may be connected with multiple records.	Updation operation is safe in a relational model as duplication of record is avoidable by applying normalisation and Primary keys relationships
8.	Hierarchical model is based on parent child relationship and deleting of child is easy as compare to parent, if we delete parent then child node will automatically deleted from the tree.	There is no deletion anomaly as deleting of one node does not affect other nodes due to many to many relationships.	The deleting of record from a relation is again a simple process and there is no anomaly related to deleting of records. Deletion of reference records is not allowed as it may linked to other records

4.3 RELATIONAL ALGEBRA AND RELATIONAL CALCULUS

The relational model uses the concept of a mathematical relation in the form of table of values which acts as building block. The table is a logical representation of data in the form of rows and columns. The relational algebra is a formal query language applied on relational model. It is a procedural language which specifies the operations to be performed on relations. The operations are performed in form of sequence of algebra operations which results in a new relation/table. The relational algebra operations can be classified into two types.

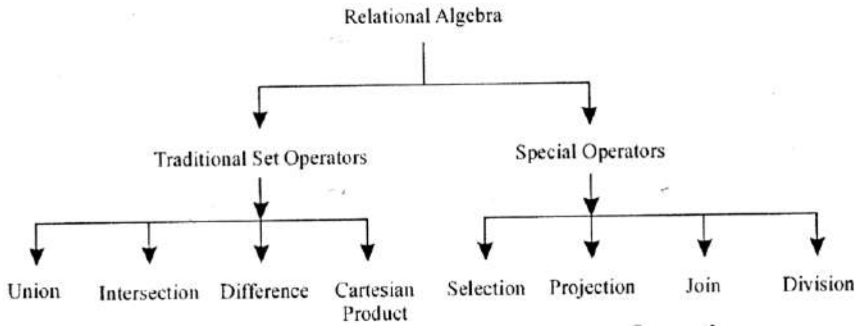


Figure 3.1: Classification of Relational Algebra Operations

Relational calculus is a non-procedural query language. Here, no procedures are provided to generate result based on 'query'. In relational calculus, query is expressed as variables and formulas on these variables. There are two types of relational calculus: tuple Relational Calculus and Domain Relational Calculus:

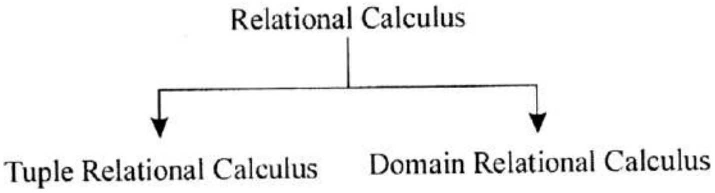


Figure 3.2: Classification of Relational Calculus

4.4 RELATIONAL ALGEBRA

1. Relational algebra is a procedural query language.
2. It consists of set of operators that take one or two relations as input and produce a new relation as output.
3. It uses relational operators.
4. It is of mainly two types which are as follows:

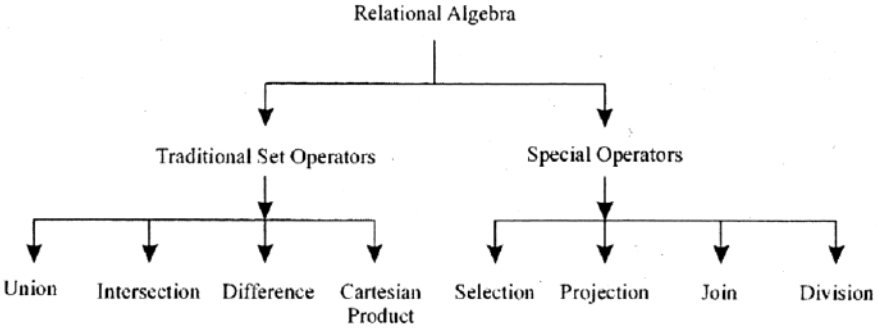


Figure 3.3: Classification of Relational Algebra

I. Traditional Set Operators

(a)Union Operator

(b)Intersection Operator

(c)Difference Operator

(d)Cartesian Product Operator

(a) Union Operator:

- Union of two relations is the set of all elements belonging to both relations.
- Result must not contain duplicate elements.
- It is denoted by U.

- For example: We want to list all the names and roll numbers which are present in both tables:'A' and 'B'.

AB

Name	Roll Number
Akhil	211
Monika	129

Name	Roll Number
Aastha	112
Akhil	211

Formula:πName, Roll Number (A) U πName, Roll Number (B).

AUB

Name	**Roll Number**
Akhil	211
Monika	129
Aastha	112

(b) Intersection Operator:

- Intersection of two relations produces a relation which contains all elements that are common to both relations.
- It is denoted by $\cap$.
- For example: We want to list only those names and roll numbers which are common inboth tables 'A' and 'B'.

A

Name	Roll Number
Akhil	211
Monika	129

B

Name	Roll Number
Aastha	112
Akhil	211

Formula:πName, Roll Number (A) $\cap$$\pi$Name, Roll Number (B)

A$\cap$B

Name	**Roll Number**
Akhil	211

(c) Difference Operator

- Difference operator is used to find those tuples which are present in one relation but not in another relation.
- It is denoted by (-) sign.
- For example: We want to list those names and roll numbers which are present in table 'A^{9} only, not in table'B'.

AB

Name	Roll Number
Akhil	211
Monika	129

Name	Roll Number
Aastha	112
Akhil	211

Formula:πName, Roll Number (A) –πName, Roll Number (B)

A-BB-A

Name	Roll Number
Monika	129

Name	Roll Number
Aastha	112

(d) Cartesian Product

- Cartesian product operator is used to combine information from any two relations.
- It is denoted by (X) symbol.
- For example: We want to list the names of employees with all departments of tables 'A'and'B'.

AB

Name	**Emp_No**	**Dept_Id**
Akhil	101	11
Monika	102	12
Aastha	101	11

Dept_Name	**Dept_Id**
Production	11
Accounts	12

Formula: πName (A) X π Dept_Name (B)

AXB

Name	Dept_Name
Akhil	Production
Akhil	Accounts
Monika	Accounts
Monika	Production
Aastha	Production
Aastha	Accounts

II. Special Operators

- **(a) Selection Operator**
- **(b) Projection Operator**
- **(c) Join Operator**
- **(d) Division Operator**

(a) Selection Operator

- Selection operator selects tuples (rows) that satisfy a given condition.
- It is denoted by lower Greek letter sigma (a).
- We can also use folio wing symbols: = >,<>>=,<= #
- For example: We want to list the tuples (employees) who live in city 'chd'.

Formula: σcity = "chd" (employee)

(b) Projection Operator

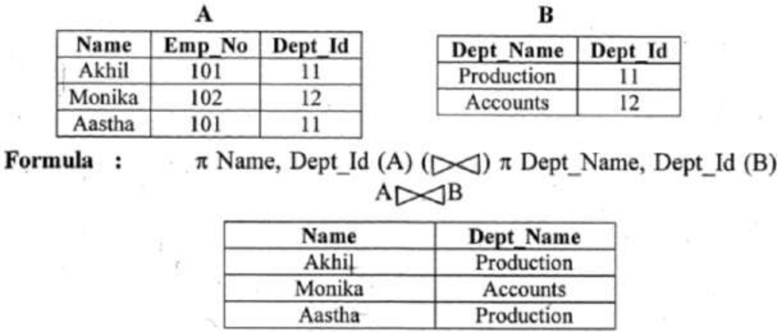
- Projection operator returns a new relation as output with certain attributes.
- It is denoted by Greek letter pie (π).
- For example: We want to list all the emp_no and name of employee.

Formula: πemp_no, name (employee)

(c) Join Operator

- Join operator is also known as natural join operator.
- It is denoted by the symbol ([><]).

- Cartesian product operator is used to combine two tables, but the output of Cartesianproduct is not correct
- Join operator is used to combine the two tables instead of Cartesian product operator.
- For example: We want to combine the two tables 'A'and 'B'.



A

Name	Emp_No	Dept_Id
Akhil	101	11
Monika	102	12
Aastha	101	11

B

Dept_Name	Dept_Id
Production	11
Accounts	12

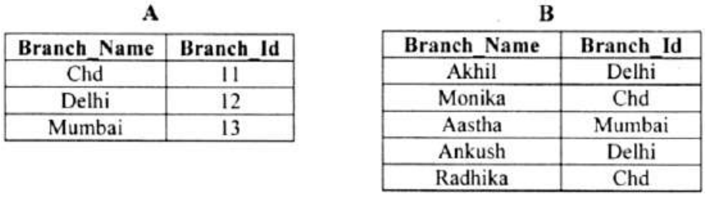
Formula : π Name, Dept_Id (A) ($\bowtie$) π Dept_Name, Dept_Id (B)

A$\bowtie$B

Name	Dept_Name
Akhil	Production
Monika	Accounts
Aastha	Production

(d) Division Operator

- Division operator will work on two relations (tables).
- It make another relation consisting of values of an attribute of one relation that match all the values in the another relation.
- It is denoted by the (÷) symbol.



A

Branch_Name	Branch_Id
Chd	11
Delhi	12
Mumbai	13

B

Branch_Name	Branch_Id
Akhil	Delhi
Monika	Chd
Aastha	Mumbai
Ankush	Delhi
Radhika	Chd

Formula: πName (A ÷B)

(A ÷B)

Name
Akhil

4.5 RELATIONAL CALCULUS

1. It was first proposed by E.F.Codd.
2. It is a formal language used to symbolize logical arguments in mathematics.

3. In relational calculus, query is expressed as formula containing number of variables and expression.
4. User will only tell the requirement without knowing the methods of retrieval.
5. User is not concerned with the procedure to obtain the results.
6. It is the responsibility of DBMS to transform these queries and give the result to the user.
7. Relational calculus is of mainly two types which are as follows:

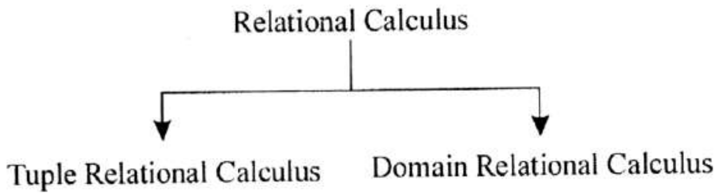


Figure 3.4: Classification of Relation Calculus

I. Tuple Oriented Relational Calculus

- It is based on specifying a number of tuples variables.
- The query of tuple relational calculus is

  {t/COND(t)}

  t-> is tuple variable

  COND (t)->is conditional expression.
- The result of such query is a relation that contains all the types (rows) that satisfy condition COND (t).

  Query of relational calculus is:

  {t. title, t. author/Book(t) and t. PRICE > 100}

  It will give us title, author of all the books whose price is greater than 100.

  Expression of tuple relational calculus is:

  {t1. A1, t2.A2, t3.t3,…tn.An/COND (t1, t2, t3, …tn)}

  t1, t2 …. are tuple variables.

  A1, A2 … are the attributes of relations.

  COND is condition.

II. Domain oriented relational calculus

- Domain calculus is different from tuple calculus in the type of variables used in formula.
- In domain oriented relational calculus, variable range will be single value rather than multiple values.
- Expression of domain oriented relational calculus is:

  {X1, X2, …Xn | COND (X1, X2, … Xn)}

  X1, X2, …Xn are domain variables.

  COND is condition or formula of domain relation calculus.

  i.e. Get employee no. of for job clerk

  EX where EMP (emp no: EX, job = 'clerk')

  Get employee name that belongs to dept no. 10 and having salary > 2000.

  Ex where EMP (ename: EX, deptno = 10, sal> 2000)

4.6 DIFFERENCE BETWEEN RELATIONAL ALGEBRA AND RELATIONAL CALCULUS

Sr. No.	Relational Algebra	Relational Calculus
1	It is a procedural method of solving the queries.	It is a non-procedural method of solving the queries.
2	It is used as a vehicle for implementation of relational calculus.	The queries of relational calculus are transformed into equivalent relational algebra format and then implemented with the help of relational algebra operators.
3	The solution to the database access problem using a relational algebra is obtained by stating what is required? And what are the steps to obtain that / information?	The solution to the database access problem using a relational calculus is obtained by stating what is required? And system will find the answer?

Questions

1. What do you mean[1] by data models? Explain the answer.
2. How can we classify data models?
3. What do mean by relationships in a data model
4. What is an attribute in data modelling?
5. Explain the Relational Model? Write advantages and disadvantages.
6. Explain the Hierarchical Model?Write advantages and disadvantages.
7. Explain different operations that can be performed on Hierarchical
8. Compare different data models.
9. List the various relational operators available in a relational model.
10. What is the difference between select and project operators?
11. Explain the various set operators available in relational algebra.
12. What is the Cartesian product operation? Why is it rarely used without a select operation?
13. What is the significance of the join operator? Explain the different types of join.
14. Explain relational calculus in detail.

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 5: NORMALIZATION

5.1 NORMALIZATION

"Normalization is the process of efficiently organizing data to minimize redundancy in a database and makes database more flexible."

1. E.F. Codd introduced the concept of normalization.
2. Normalization technique is used in designing relational model.
3. It improves database design and removes anomalies for database activities.
4. Its objective is to reduce the redundancy (duplicity) and eliminates the insertion, updation, and deletion anomalies from the database.
5. To achieve its objective, it breaks the database into smaller tables and establishes the relationships between those tables.
6. It makes data consistent throughout the database.
7. Normalization follows some rules. Each rule is known as normal form.
8. E.F. Codd introduced the first normal form (INF) in 1970.
9. He introduced the second normal form (2NF), third normal form (3NF) in 1971 and boyce codd normal form (BCNF) in 1974.
10. For many applications, third normal form (3NF) is necessary.
11. Fourth normal form (4NF) was introduced by Ronald Fagin in 1977.
12. Normal forms are numbered from lowest (INF) to highest (5NF).
13. ***The following are the disadvantages of normalizations:***

- It is a difficult and time consuming process.
- Sometimes, the performance of database degrades from lowest (INF) to highest (5NF).

14. Un-normalized Form (UNF) is one in which a table contains non atomic values at each row. Non atomic values need further decomposition for simplification. For the simplification, un-normalized form goes into first normal form.
15. ***The levels/steps of normalization are as follows:***

Un-normalized Form (UNF)

↓

First Normal Form (INF)

↓

Second Normal Form (2NF)

↓

Third Normal Form (3NF)

↓

Boyce Cod'd Normal Form (BCNF)

↓

Fourth Normal Form (4NF)

↓

Fifth Normal Form (5NF)

Fig. 5.1: Steps of Normalization

5.2 FUNCTIONAL DEPENDENCY

1. Functional dependency is an association between two attributes (columns) of the same relation (table).
2. It is basically a constraint between two sets attributes from the same relation in a database.
3. One attribute is called determinant and other is called determined.
4. For each value of determinant, there is only one value of determined.
5. For example: A→B
    - "B is functionally dependent on A" because for each value of attribute 'A', there is exactly one value of attribute 'B'.
    - If A is determinant and B is determined then we can say that
      "A functionally determines B" OR "B is functionally dependent on A".

Supplier

Sr. No,	**Name**	**Status**	**City**
S1	Akhil	10	Delhi
S2	Monika	2.0	Patiala
S3	Aastha	30	Delhi

In above table "Supplier", attribute 'Name' is functionally dependent (FD) on attribute 'Sr. No.' because 'Name' has only one value for given 'Sr. No.'.

We can say Sr. No. → Name

'Sr. No' is determinant and 'Name' is determined.

But attribute 'City' is not functionally dependent (FD) on attribute ' Sr. No.' because 'City' has more than one value for given 'Sr. No.'.

5.3 FULLY FUNCTIONAL DEPENDENCY

1. Fully functional dependency is a functional dependency in which all the non-key attributes are dependent on the key attributes.
2. For example: A → B
   - "B is fully functionally dependent on A" because 'B' is functionally dependent on 'A' but not on any proper subset of 'A'.
   - "B is fully functionally dependent on A" means we cannot identify the value of 'B' only from 'A', we can identify the value of 'B' from 'A' and another attribute from the same relation. Another attribute will help the 'A' to find the value of 'B'.
   - If we delete any attribute from the relation, then it will violate the concept of functional dependency.
3. In the below table, Qty. is F.F.D. on 'Sr. No.' and 'Pr. No.' because we can get the value of Qty. only by the combination of both 'Sf. No.' and 'Pr. No.'.

Sr. No.	**Pr. No.**	**Qty.**
S1	P1	270
S1	P2	300
S1	P3	700
S2	P1	270
S2	P2	700
S3	P2	300

$$\underbrace{\text{Sr. No., Pr. No.}}_{A} \longrightarrow \underbrace{\text{Qty.}}_{B}$$

5.4 PARTIAL FUNCTIONAL DEPENDENCY

1. Partial functional dependency occurs, when some non-key attribute depends on primary key attribute.

2. For example: A → B

The attribute 'B' is partial functional dependent on attribute 'A', if there is some attribute that can be removed from 'A' and yet the dependency holds.

5.5 TRANSITIVE FUNCTIONAL DEPENDENCY

1. Transitive functional dependency occurs, when some non-key attribute depends upon other non-key attributes.

2. For example: There are three attributes 'AVB' and 'C'.

- $A \rightarrow B$
- $B \rightarrow C$
- $\Rightarrow A \rightarrow C$

It means 'C' is transitively dependent on 'A'.

5.6 MULTI VALUED DEPENDENCY

1. Multivalued dependency is a full constraint between two sets of attributes in a relation.

2. It plays a role in fourth normal form (4NF) of normalization.

3. For example: If there are three attributes 'A', 'B' and 'C' in a relation.

- 'B' and 'C' are independent from each other.
- 'B' and 'C' are multi valued fact about A.

  Then $A \rightarrow \rightarrow B$

  $A \rightarrow \rightarrow C$

- Then we can say that "A multi determines B" OR "B is multi dependent on A".

Course_Student_Book

Course	Student	Book
Chemistry	Akhil	B1
Chemistry	Akhil	C1
Physics	Moaika	A1
Physics	Monika	D1
Chemistry	Aastha	Bl
Chemistry	Aastha	Cl
English	Rohit	A1
English	Rohit	D1

Course $\rightarrow\rightarrow$ Student

Course $\rightarrow\rightarrow$ Book

5.7 FIRST NORMAL FORM (1NF)

1. E.F. Cold introduced the first normal form (1NF) in 1970.
2. First normal form (1NF) eliminates the repeating columns from an un-normalized table.
3. In 1NF, there is no repeating column (group).
4. We convert un-normalized table into normalized for.
5. Primary key is required in each table to identify a record.
6. The purpose of primary key is to uniquely identify a record.
7. First normal from depends on the functional dependency.
8. Formula : f(x)=y

   For every value of x, there is only one value for y.
9. For example: The following table "Student" having columns (Name, Course, Roll Number) is an un-normalized table. We have to convert this un-normalized table into normalized table.

Student

Name	Course	Roll Number
Akhil	Science	211, 128
Monika	Computer	129
Aastha	Management	112

The above table "Student" is un-normalized because it contains more than one value for the column 'Roll Number'. 'Akhil' has two values (211, 128) for the column 'roll number' which is not possible. For normalization, there should be only one value in one column.

The following are two methods to convert un-normalized table into normalized table:

- **Method 1:** To convert the un-normalized table "Student" into normalized form, we decompose (divide) this un-normalized table into two tables.

Student 1

Name	Course
Akhil	Science
Monika	Computer
A'astha	Management

Student 2

Name	Roll Number
Akhil	211
Akhil	128
Monika	129
Aastha	112

- **Method 2:** To convert the un-normalized table "Student" into normalized form, we convert this this un-normalized table into flat table.

Student

Name	Course	Roll Number
Akhil	Science	211
Akhil	Science	128
Monika	Computer	129
Aastha	Management	112

Anomalies in First Normal Form (INF)

1. **Insert Anomaly: We** cannot insert any information of new student in table "Student" until he join any course. Similarly, we cannot insert any information about the course until there is any student. This phenomenon is known as insert anomaly.

Student

Name	Roll Number	Course
Akhil	211	Science
Monika	129	Computer
Aastha	112	Management

Rohit	**111**

The details of new student 'Rohit' cannot insert into the table "Student" until he join any course. It is called insert anomaly.

2. **Update Anomaly :In** the update anomaly, if we want to change (update) the course of any student, then we have to change (update) the multiple records. If we change the course of the student but forget to change the details of that student from all the locations where it occures, then data become inconsistent/This phenomenon is known as update anomaly.

3. **Delete Anomaly**: If we delete any course from table "Student", then all the related information to that course automatically deletes.

For Example: if we delete the course 'management' from the table "Student", then it automatically ceases the name 'Aastha' and roll number '112'.

Student

Name	Roll Number	Course
Akhil	211	Science
Monika	129	Computer
Aastha	112	Management

After deletion, table "Student will be look like:

Student

Name	Roll Number	Course
Akhil	211	Science
Monika	129	Computer

5.8 SECOND NORMAL FORM (2NF)

1. E.F. Codd introduced the second normal form (2NF) in 1971.
2. A relation is in 2NF if it fulfills the following conditions
   - Relation should be in INF and
   - Every non-key attribute (non-prime attribute) is fully functionally dependent on Primary key.
3. For example-.The following table "Products" having columns (Item, Price, Quantity, Order Number, and Order Date) is in INF.

Products

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015

Watch	800	18	13	3-7-2015
Wallet	600	12	14	4-7-2015

- The table "Products" has two primary key columns (Item and Order Number).
- Price (non-primary key column) is fully functionally dependent on Item (prime key column).
- Order Date (non-primary key column) is fully functionally dependent on Order Number (prime key column).
- The table "Products" can be converted into second normal form (2NF) by decomposing it into sub tables such as:

Item	**Price**
Mobile	2000
Sunglasses	1000
Watch	800
Wallet	600

Order Number	**Order Date**
11	1-7-2015
12	2-7-2015
13	3-7-2015
14	4-7-2015

Item	**Quantity**	**Number**
Mobile	20	11
Sunglasses	15	12
Watch	18	13
Wallet	12	14

Anomalies in Second Normal Form (2NF):

1. **Insert Anomaly:** Second form (2NF) also Suffers from the inset anomaly same like the first normal form (1NF). We cannot insert any information of 'Price' in table "Products" until is associates with any 'item'. Similarly, we cannot insert any information about the 'item' in the table "Products" until its price is fixed. This phenomenon is known as insert anomaly.

Products

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015
Watch	800	18	13	3-7-2015
Wallet	600	12	14	4-7-2015
	1200	16	15	5-5-2015

The details of new price '1200' cannot insert into the table "Products" until it associates with any 'item'. We cannot left blank the value of any column. It is called insert anomaly.

2. **Update Anomaly:** In the update anomaly, if we want to change (update) the 'price' of any 'item', them we has to change (update) the multiple records. It we change the 'price' of any 'item' but forget to change the details of that 'item' from all the locations where is occurs, then data become inconsistent. This phenomenon is known as update anomaly.

3. **Delete Anomaly:** Like 1NF, 2NF also suffers with delete anomaly. If we delete any 'item' from table "Products", then all the related information to that 'item' automatically deletes. For example : if we delete the item 'watch' from the table "products", then it automatically deletes it's all related information (price, quantity, order number, order date).

Products

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015

Watch	800	18	13	3-7-2015
Wallet	600	12	14	4-7-2015

After deletion, table "products" will be look:

Product

Item	Price	Quantity	Order Number	Order Date
Mobile	2000	20	11	1-7-2015
Sunglasses	1000	15	12	2-7-2015
Wallet	600	12	14	4-7-2015

5.9 THIRD NORMAL FORM (3NF)

1. E.F. Codd introduced the third normal form (3NF) in 1971.
2. It means a relation (table) is in 3NF if it is in 2NF and there is no transitive dependency.
3. The objective to 3NF is to remove all transitive dependencies.
4. A relation is in 3NF if it fulfills the following conditions:
    - Relation should be in 2NF and
    - Every non-key attribute (non-prime attribute) is transitively dependent on Primary key only.
5. It removes the anomalies of 2NF.
6. For many applications, third-normal form (3NF) is necessary.
7. For example: The following table "Record" having columns (Name, Roll Number, System, Number, Hours_Rate) is in 2NF.

Record

Name	Roll Number	System Number	Hours Rate
Aastha	112	S1	20

Akhil	211	S2	18
Monika	129	S3	17
Rohit	219	S2	15
Aditya	285	S3	16
Kunal	712	S4	12
Sachin	125	S1	23
Rahul	231	S4	25
Siddharth	123	S5	13

- 'Name' is a primary key and the entire non-key attributes (Roll Number, System Number, Homrs_Rate) are dependent on it.
- To convert the table "Record" into 3NF, we decompose it into two tables (Student 'Record, Charge Record).

Student Record

Name	Roll Number	System Number
Aastha	112	SI
Akhil	211	S2
Monika	129	S3
Rohit	219	S2
Aditya	285	S3
Kunal	712	S4
Sachin	125	S1
Rahul	231	S4
Siddharth	123	S5

Charge Record

System Number	Hours Rate

SI	43
S2	33
S3	35
S4	37
S5	13

- Table "Student Record" provides the detail of student like Name, Roll Number and System Number used by him/her.
- Table "Charge Record" provides the details of system like System Number, Charges for using System.

Anomalies in Third Normal Form (3NF)

1. **Insert Anomaly:** Third normal form (3NF) is also suffers from insert anomaly but upto some extent. It is possible to insert in advance, the rate to be charged from student for a system.
2. **Update Anomaly:** If Hours_Rate for a system in table "System Record" changed (updated), then we need only to change a single record in table "Charge Record".
3. **Delete Anomaly:** It we delete the record of a student who is only student working on a particular system, then we will not lose the information of the system and hours_rate of that system.

5.10 BOYCE CODD NORMAL FORM (BCNF)

1. E.F. Codd introduced the Boyce Codd Normal Form (BCNF) in 1 974.
2. A relation is in BCNF, if it is in 3NF and every determinant (attribute) is a candidate key.
3. It means BCNF have multiple candidate keys (more than one primary key).

5.11 FOURTH NORMAL FORM (4NF)

1. Fourth normal form (4NF) was introduced by Ronald Fagin in 1977.

2. 2NF, 3NF and BCNF are concerned with functional dependencies whereas 4NF concerned with multivalued dependencies.
3. A relation is in 4NF if it is in 3NF or BCNF and contains no multi valued dependencies.
4. For example: The following table "Course_Student_Book" is in 3NF.

Course Student Book

Course	**Student**	**Book**
Chemistry	Akhil	Organic Chemistry
Chemistry	Akhil	Physical Chemistry
Physics	Monika	Optics
Physics	Monika	Mechanics
Chemistry	Aastha	Organic Chemistry
Chemistry	Aastha	Physical Chemistry
English	Rohit	English Literature
English	Rohit	English Grammar

- Attributes 'Student' and 'Book' are multivalued facts about the attribute 'Course'. There are many students for one course and many books for one course.
- The condition of 4NF is that there should be no multi valued attribute in a table.
- To convert the table "Course_Student_Book" into 4NF, we decompose it into two tables (Course Student, Course Book).
- Table "Course-Student" tells us which student is studying which course.
- Table "Course_Book" tells us which book is available for which course.

Course Student

Course	**Student**
Chemistry	Akhil
Physics	Monika
Chemistry	Aastha

Course Book

Course	**Student**
Chemistry	Organic Chemistry
Chemistry	Physical Chemistry
Physics	Optics

English	Rohit

Physics	Mechanics
English	English Literature
English	English Grammar

Note: If anew student 'Rahul' wants to join a course 'English' and use books of 'English' and 'Chemistry', then we have to insert new information of student 'Rahul'. We will insert the name 'Rahul' twice. First entry for 'English' and second entry for 'Chemistry'.

5.12 FIFTH NORNAL FORM (5NF)

1. A relation is in 5NF if it is in 4NF and based on join dependency.
2. Join dependency means when a table is decompose/divide into three or more tables, and then the resulting tables (divided tables) can be rejoined to form the original table.
3. The following are three sub tables (Course_Student, Course_Book and Student_Book) of original table "Course_Student_Book". *The table "Course_Student_Book" is used in the fourth normal form (4NF).*

Course_Student

Course	**Student**
Chemistiy	Akliil
Physics	Monika
Chemistry	Aastha
English	Rohit

Course	**Book**
Chemistry	Organic Chemistry
Chemistry	Physical Chemistry
Physics	Optics
Physics	Mechanics
English	English Literature
English	English Grammar

Student_Book

Student	Book
Akhil	Organic Chemistry
Akhil	Physical Chemistry
Monika	Optics
Monika	Mechanics
Aastha	Organic Chemistry
Aastha	Physical Chemistry '
Rohit	English Literature
Rohit	English Grammar

4. When we will join these three tables (Course_Student? Course_Book and Student_Book), then we will get the original table "Course_Student_Book".

Questions

1. What is Normalization? State and explain its types.
2. What is the need of Normalization of data? What are the various techniques for normalization in relational database model?
3. What is Functional dependency? Explain in detail Give an example also.
4. What do you mean by redundancy? Explain the ways to remove it from the database?
5. What do you mean by Normal forms? Explain the various types of it along with the suitable example.
6. What is the difference between First and second Normal Forms?
7. What is INF? Give example to demonstrate how INF improves a table.
8. Discuss 2NF. Discuss the problems that can be encountered in a table, which is in INF, How 2NF solve them?
9. Define 3NF? Discuss its need.
10. Explain Boyce Codd Normal Form.
11. Explain multivalued dependency. Give an example.

12. Explain Join dependency. Give an example.

13. Explain 4NF along with example.

14. What do you mean by FDs? Explain the Closure of a Set of FDs.

15. Explain 5NF along with example.

16. What is fully functional dependency? Give an example.

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 6: TRANSACTION MANAGEMENT AND CONCURRENCY CONTROL

6.1 TRANSACTION

1. One or more operations collectivelya single unit of database work is known as database transaction.
2. A transaction is a group of actions such as select, insert, update and delete performed on the database to change the state of a database.
3. A transaction is an action orof actions carried out by the user or the application.

4. It is an atomic operation by the use in reality and goes through number of states during its-lifetime.
5. Once a transaction starts, it ends with success or failure.
6. A successful transaction commits andreaches a new consistent state.
7. A failed transaction and 'rolled back' or 'undone'. In the failed transaction; database restores the previous consistent state.
8. A database transaction must be ACID ([Atomicity-Consistency, Isolation, and Durability]. We will discuss ACID of transaction in *section 7.1.2.*
9. Transactions are supported by SQL [Structured Query Language]. We will discuss SQL in *chapter 10.*
10. A transaction is required to manage of oversees the sequence of events (transactions).

6.1.1 States of Transaction

The database transaction can be in one of the following four states:

1. Active State

- A transaction is in active state while its statements start to be executed.
- Once astate, it starts executing its statements and ends with the commit state.
- Sometimes, with partially committed state. At this phase, the database has its but it is still, possible for the transaction to be aborted because the output is residing temporarily in main memory due to hardware failure.

2. Failed State

- After the active state, sometimes transaction enters in the failed state because its execution can no longer proceed (due to program error or hardware error).

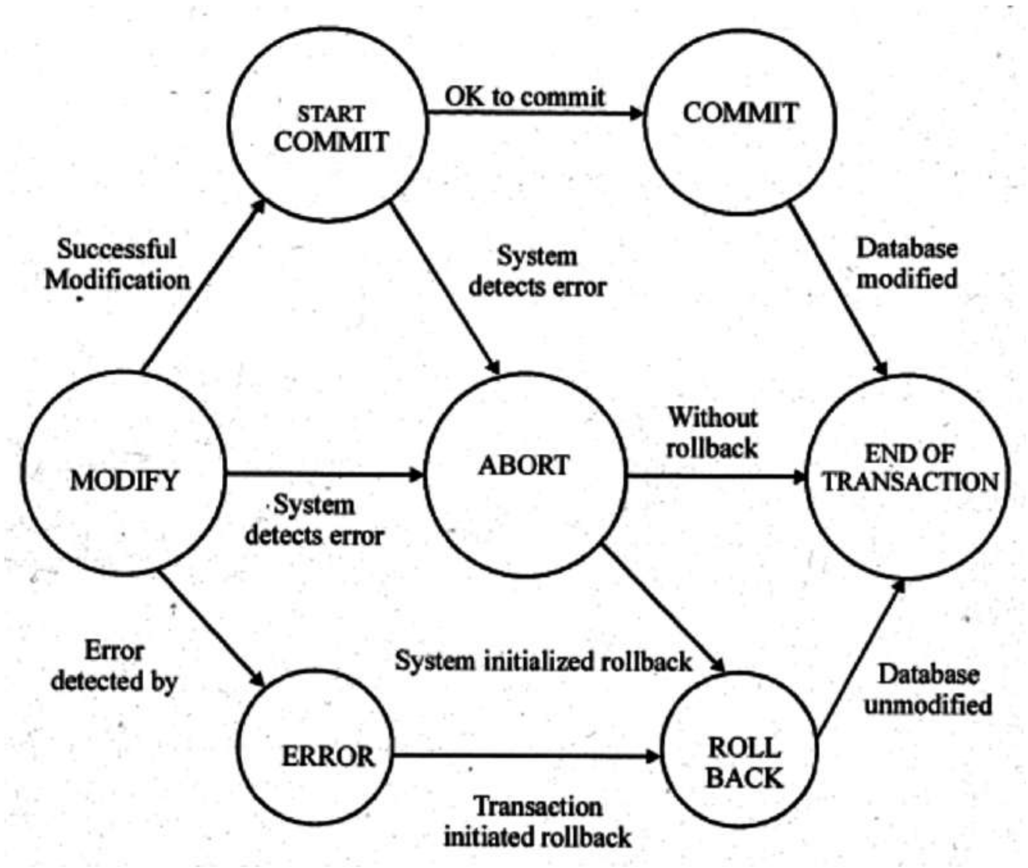


Fig. 6.1: States of Transaction

3. Aborted State

- Aborted state comes, when transaction end with failure.
- A failed transaction enters in the aborted state and 'rolled back' or 'undone'.
- An aborted transaction has no effect on database and can retain its consistent state.
- In the failed transaction, database restores the previous consistent state.

4. Committed State

- Committed state comes, when transaction end with success.
- A successful transaction commits and database reaches in a new consistent state.
- A successful transaction cannot 'rolled back' or 'undone'

6.1.2 Acid Properties of Transaction

To ensure the integrity of data we require that the database system maintains the properties of transactions abbreviated as ACID. The ACID properties of transaction are as follows:

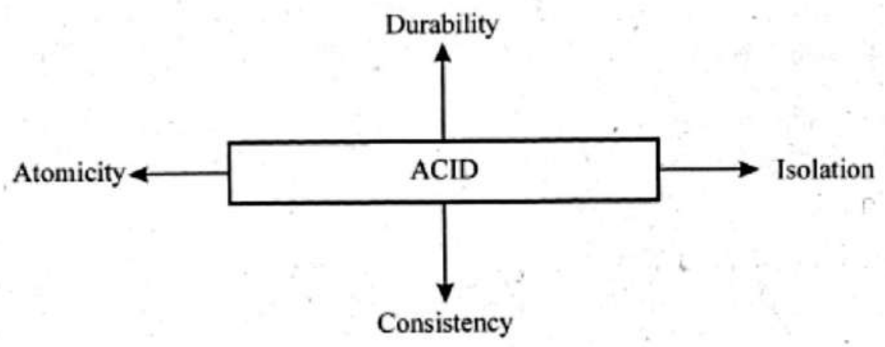


Fig. 6.2: Acid Properties Atomicity

1. **Atomicity**

   - Atomicity means either 100% modification on 0% modification.
   - According to atomicity, we cannot perform half operations of any transaction.
   - It means we should perform all the operations of the transaction or we should not perform any operation.
   - It ensures that either a transaction ends with committed state or rolled back state.
   - Committed state comes, when transaction end with success and database reaches in a new consistent state.
   - Aborted state comes, when transaction end with failure and database restores the previous consistent state. A failed transaction enters in the aborted state and 'rolled back' or 'undone'.
   - Ensuring atomicity is the responsibility of the database system itself. It is handled by a component called the *transaction management component.*

2. **Consistency**

- According to the -consistency property of the transaction, database remains in a valid state (consistent state) before and after the transaction is committed.
- It means transaction cannot violate the rules (integrity constraints) of the database.
- Ensuring consistency for an individual transaction is the responsibility of the *application manager* who codes the transaction.

3. Isolation

- It means that the execution of one transaction in not affected the other concurrent transactions.
- According to isolation property, data used during the execution of one transaction is not used by another transaction until the execution is not completed.
- It means that the actions performed by a transaction will be isolated or hidden from outside the transaction until the transaction terminates.
- Ensuring the isolation property is the responsibility of a component of a database system called the concurrency control component.

4. Durability

- According to the durability, once a transaction completes successfully, all the updates that it carried out on the database persists even if there is a system failure after the transaction completes execution.
- After the transaction has been successfully completed, all the modifications of a transaction will permanent (cannot 'roll back' or 'undone') even if the system failure occurs.

6.1.3 Scheduling of Transaction

- A schedule is a list of actions (Reading, Writing, Aborting or Committing) from a set of transactions.

- A schedule is a sequence of the operations by a set of concurrent transactions that preserves the order of the operations in each of the individual transactions.
- For example: ^{4}M' is a schedule. Schedule 'M' is a set of three transactions Tl, T2 and T3. Transaction Tl reads and writes to object A.

  Transaction T2 reads and writes to object B.

  Transaction T3 reads and writes to object C.

Schedule 'M'

Tl	T2	T3
Read (A)		
Write (A)		
Commit		
	Read (B)	
	Write (B)	
	Commit	
		Read (C)
		Write (C)
		Commit

Example of Scheduling of Three Transactions

6.1.3.1 Types of Schedules

There are two types of schedules:

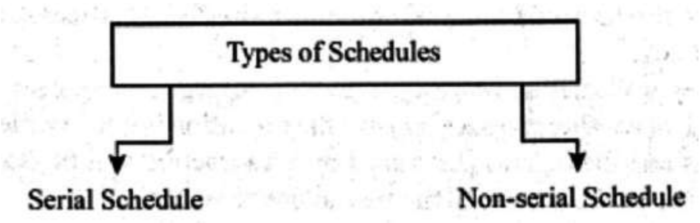


Fig. 6.3: Types of Schedules

1. Serial Schedule

- In a serial schedule, the transactions are performed in serial order. T there is no interference between transactions and only one transaction is executing at a given time.
- In a serial schedule, all the steps (operations) of each transaction executed consecutively without overlapping.
- For example: Schedule 'M' is a set of three transactions Tl, T2 and T3. Schedule 'M' is a serial schedule because the actions of the 3 transactions (Tl, T2 and T3) are not interleaved and executed consecutively.

  Transaction Tl reads and writes to object A.

  Transaction T2 reads and writes to object B.

  Transaction T3 reads and writes to object C.

Schedule 'M'

Tl	T2	T3
Read (A)		
Write (A)		
Commit		
	Read (B)	
	Write (B)	
	Commit	
		Read(C)
		Write (C)
		Commit

Example of Serial Schedule

2. Non-Serial Schedule

- A schedule where the operations from a set of concurrent transactions are interleaved.

- In a non-serial schedule, if the operations of the transactions are not properly interleaved, then they result in problems such as lost update, dirty read and inconsistency analysis.
- For example: Schedule 'N' is a set of three transactions T1, T2 and T3. Schedule 'N' is a non-serial schedule because the actions of the 3 transactions (T 1, T2 and T3) are interleaved.

  Transaction Tl reads and writes to object A.

  Transaction T2 reads and writes to object B.

  Transaction T3 reads and writes to object C.

Schedule 'N'

Tl	T2	T3
Read (A)		
	Read (B)	
		Read (C)
Write (A)		
	Write (B)	
		Write (C)
Commit	Commit	Commit

Example of Non-Serial Schedule

6.1.4 **Serializability**

- The objective of serializability is to find out the non-serial schedules and allow the all transactions to execute concurrently without creating any problem.
- In a non-serial schedule, if the operations of the transactions are not properly interleaved, then they result in problems such as lost update, dirty read and inconsistency analysis.

Schedule 'N'

Tl	T2	T3
Read (A)		
	Read (B)	
		Read (C)
Write (A)		
	Write (B)	
		Write (C)
Commit	Commit	Commit

Example of Non-Serial Schedule

- Serializability is used to find out or prevent the inconsistency (problems) occurs during the non-serial schedule.

6.2 CONCURRENCY CONTROL

1. ***"Concurrency control is the activity of coordinating concurrent accesses to a database in a multi-user database management system.lt is used to coordinate simultaneous transactions while preserving data integrity".***

2. When many transactions execute concurrently, then the concurrency control scheme is used to make isolation.
3. The concurrency control is required when there are multiple accesses to same data by multiple users.
4. Concurrency control in DBMS ensures that transactions are performed concurrently without the concurrency violating the data integrity of a databse.
5. Executed transaction should follow the ACID rules [Atomicity, Consistency, Isolation, and Durability]. We discussed ACID rules in the section 7.1.2.
6. The DBMS must guarantee that only realizable.
7. It also guarantees that no effect of committed transactions is lost, and no effect of aborted (rolled back) transactions remains in the related database.

8. For example: Two travelers who go to electronic ticket booking center at the same time to purchase a train ticket to the same destination on the same train. There's only one seat left in the coach, but without concurrency control, it's possible that both travelers will end up purchasing a ticket for that one seat, However, with concurrency control, the database wouldn't allow this to happen. Both travelers would still be able to access the train seating database, but concurrency control would preserve data accuracy and allow only one traveler to purchase the seat.

9. Similarly, the concurrency control protocol is used to schedule transactions in such a way as to avoid any interference between them. It allows only one transaction to execute at a time; one transaction is committed before the next transaction is allowed to begin.

6.2.1 Need of Concurrency Control

Several problems can occur when concurrent transactions execute in an uncontrolled manner. The some of the problems are as follows:

1. The Lost Update Problem: This problem occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database item incorrect. Successfully completed update is overridden by another user.

2. The Temporary Update Problem: This problem occurs when one transaction updates a database item and then the transaction fails for some reason. The updated item is accessed by another transaction before it is changed back to its original value. Occurs when one transaction can see intermediate results of another transaction before it has committed.

3. The Incorrect Summary Problem: If one transaction is calculating an aggregate summary function on a number of records while other transactions are updating some of these records, the aggregate function may calculate some values before they are updated and others after they are updated.

6.3 CONCURRENCY CONTROL METHODS/SCHEMES

There are many concurrency control methods/schemes to prevent the conflicts between the transactions. The following are some concurrency control methods:

6.3.1 Locks

1. A lock is a variable associated with the data item to describe its status.
2. Locks are used in concurrent transactions to ensure serializability.
3. It prevents undesired or inconsistent operations on shared resources by other current transactions.
4. They are used to make the isolation property of transaction in the concurrent environment.
5. They describe the status of the data item whether it has been modified or not.
6. A lock on any database object needs to be acquired by the transaction before accessing it.
7. If transaction 'A' acquires a lock on a database object and another transaction 'B' needs to access that database object, then the existing type of lock is checked.
8. According to the locking scheme, if the existing type of lock (transaction 'A') is matched with another transaction's lock (transaction ^{C}B'), then transaction ^{4}B' can use that object.
9. But, if the existing type of lock (transaction 'A') is not matched with another transaction's lock (transaction 'B'), then transaction attempting access is aborted or blocked.
10. There are many types of locks but only one lock is used for each item in database.

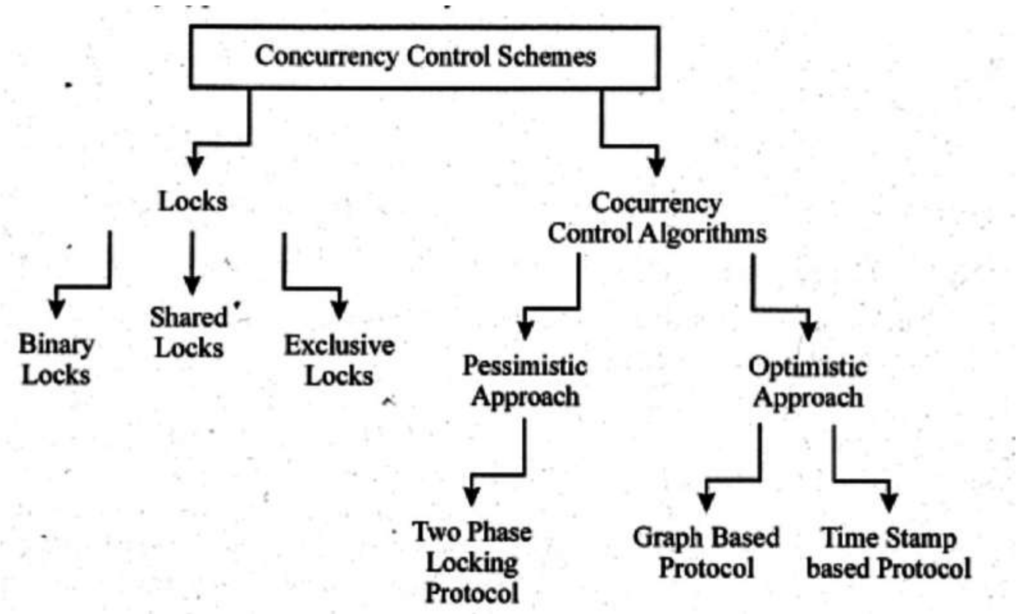

Concurrency Control Schemes
Locks
Cocurrency Control Algorithms
Binary Locks
Shared Locks
Exclusive Locks
Pessimistic Approach
Optimistic Approach
Two Phase Locking Protocol
Graph Based Protocol
Time Stamp based Protocol

Fig. 6.4: Concurrency Control Methods/Schemes

6.3.1.1 Types of Locks

The following are the types of locks:

I. Binary Locks

1. Binary lock has two states i.e. locked or unlocked.
2. When we use binary lock, it may assign 0 or 1 to the data items.
3. Locked state is represented by 1. It means item cannot be accessed.
4. Unlocked state is represented by 0. It means item can be accessed.
5. If a transaction 'A' wants to access a data item, then it must request for lock.
6. If the data item is already used by another transaction ^{4}B, then transaction 'B' got lock [1] on that date item. It means transaction 'A' cannot access that data item and get zero [0] state.
7. It means transaction 'A' has to wait to access that data item until transaction 'B' finished.
8. Binary lock follow some rules which are as follows:
   - A transaction must get a lock on data item on which it wants to perform read or write operation.

- After the read or write operation, transaction must unlock the data item.
- If any data item holds a lock, then no other transaction can make a lock on that particular data item.
- No two transactions can get the lock on the same data item. It means only one transaction can get the lock on a particular data item.

9. They are not used in practice because only one transaction can hold a lock on a given data item at a particular time which is very impractical.

II. Shared Locks

1. In a binary lock, only one transaction can get the lock on a particular data item. But in shared lock, more than one transaction can use shared fock at a particular time.
2. It is denoted by 'S'.
3. Shared lock is used only for reading purpose. It means, if a transaction want to read data then it will use shared lock on it.
4. Read lock is a shared lock. It means multiple transactions can have read lock on the same item in order to read it.
5. If a transaction 'A' has a shared lock on data item 'M', then other transaction 'B' can only read that data item 'M' not write.
6. For example:

   Lock_S (M): → It is used to request a shared lock on data item 'M'.

   Unlock (M): → It is used to unlock data item 'M'.

III. Exclusive Locks

1. In a binary lock, only one transaction can get the lock on a particular data item. But in exclusive look, more than one transaction can use exclusive lock at a particular time.
2. It is denoted by 'X'
3. Exclusive lock is used only for writing purpose. It means, if a transaction want to write data then it will use exclusive lock on it.

4. Write tock is an exclusive lock. It means multiple transactions can have write lock on the same item in order to write it.

5. If a transaction 'T1' has obtains an exclusive lock on a data item then another transaction 'T2' cannot perform read but performs write operation.

6. If a transaction 'A' has a exclusive lock on data item 'M', then other transaction 'B' can only write that data item 'M' not read.

7. For example:

   Lock_X (M): → It is used to request an exclusive lock on data item 'M'.

   Unlock (M): →It is used to unlock data item 'M'.

6.3.1.2 Compatibility of Locks

Compatibility of Locks	**Shared**	**Exclusive**
Shared	True	False
Exclusive	False	False

1. **Shared lock is compatible with shared lock:** According to this, more than one transaction can read a data item. It means multiple transactions can have read lock on the same item in order to read it.

2. **Shared lock is not compatible with exclusive lock:** According to this, if a data item has exclusive lock, then no other transaction can make shared lock on that particular data item.

3. **Exclusive lock is not Compatible with exclusive** lock: According to this, if a data item has exclusive lock, then no other transaction can make exclusive lock on that particular data item. No two transactions can make exclusive lock simultaneously.

6.3.2 Concurrency Control Algorithms

To control the concurrency problems, there are two algorithms which as follows:

6.3.2.1 Pessimistic Approach

1. In this approach, if the transactions conflict with each other, then there should be some delay in the transactions.

2. **Two phase locking protocol based on pessimistic approach as follows:**
   - It is a common locking protocol which guarantees the serializability.
   - It does not ensure the freedom from deadlock.
   - It has two phases i.e. growing phase and shrinking phase.

   (a) ***Growing Phase:*** In the growing phase, number of locks increases. In this phase, all locks are requested and no one is released. When a transaction begins, it is in a growing phase and required lock is provided.

   (b) ***Shrinking Phase:*** In the shrinking phase, number of locks decreases. In this phase, all locks are released and no one is requested. When a transaction ends, it is in shrinking phase. It releases the lock and cannot get any more lock.

6.3.2.2 Optimistic Approach

1. Optimistic approach is also known as validation or certification method.
2. In this approach, there is an assumption that conflicts in database operations are very rare.
3. There is no checking process during the execution of a transaction. Transaction runs unsynchronized and conflicts are checked only at the end.
4. According to this approach, first let the transaction run to the completion, then check the conflicts before the transaction commits.
5. **Advantages of Optimistic Method**
   - This technique is very efficient when conflicts are rare.
   - The rollback involves only the local copy of data.
6. **Disadvantages of Optimistic Method**
   - Conflicts are expensive to deal.
   - Longer transactions are more likely to have conflicts and may be repeatedly rolled back.

7. Non-Two phase locking protocol based on optimistic approach as follows:

(a) Graph Based Protocol

1. It is also known as tree protocol.
2. In this protocol, data items are arrange in a tree.
3. We must have prior knowledge about the order in which the database items will be accessed.
4. (X, Y) are shows that X is parent of Y.
5. If there is directed path from X to Y then X is called ancestor of Y.
6. It ensures serializability.
7. **The following are some rules of graph based protocol:**

- No data can be accessed unless transaction locks it.
- Transaction can unlock data any time.
- Transaction after unlocking data cannot relock it again.

8. **Advantages of Graph Based Protocol**

- Unlocking may occur earlier which may lead to shorter waiting time.
- It is deadlock free, No rollback is required.

9. **Disadvantages of Graph Based Protocol**

- In most of the cases, it is not known prior what data will need to be lock.

(b) Time Stamp Based Protocol

1. Time stamp based protocol is used in relational databases to safely handle transactions
2. In this protocol, we must have the prior knowledge of order of transactions and data items.
3. The time stamp is assigned by the database system before the transaction starts.
4. A unique fixed time stamp [TS(Ti)] is associated with each transaction.
5. It follows the serializability order.
6. **There are two methods for assigning the time stamp to each transaction as follows:**

- Use a separate counter to assign the time stamp.
- Use a system clock to assign the time stamp.

7. We can implement time stamp on data item 'M' as follows:
   - W-timestamp (M) :-> It is a time stamp for writing on data item 'M'.
   - R-timestamp (M):-> It is a time stamp for reading on data item 'M'.

6.4 DEADLOCK

1. Deadlock is basically a mutual blocking between transactions.
2. A system is in deadlock state if there are set of transaction and every transaction is waiting for other transaction to release lock.
3. Example of deadlock:
   - There are two transactions ^{C}TP and 'T2'.
   - Transaction 'T1' has exclusive lock and transaction 'T2' has shared lock
   - Transactions 'Tl' and 'T2' execute concurrently.
   - Transaction 'Tl' make exclusive lock (X) on data item 'A'.
   - Transaction 'T2' make shared lock (S) on data item 'B'.

Tl	**T2**
Lock-X(A)	
Read (A, a)	
a = a-50	
Write (A, a)	
	Lock-S(B)
	Read(B, b)
	Unlock (B)
	Lock-S(A)
	Wait.......
Lock -X(B)	
Wait…………	

- Transaction 'T2' make shared lock (S) on data item 'B' and want to make shared lock on data item 'A'. But transaction 'T2' has to wait till the transaction 'Tl' release an exclusive lock (X) on data item 'A'.
- On the other hand, transaction 'T' make exclusive lock (X) on data item 'A' and want to make exclusive lock on data item 'B'. But the transaction 'TP' has to wait till the transaction 'T2' release the shared lock (S) on data item 'B'.
- Both transactions 'Tl' and 'T2' cannot release the lock because shared and exclusive locks are incompatible with each other.
- This is the situation where neither of the transactions can proceed. Both transactions wait for each other to release the lock but both are incompatible and cannot release the lock. This situation is called deadlock.

4. A deadlock can be resolved by aborting a transactions and breaking the cycle.
5. When deadlock occur, system must rollback one of transaction.
6. One transaction rollback the data that were locked by that transaction are unlocked and is available to other transaction.

6.4.1 Reasons for the Occurence of Deadlock

A set of processes is in a deadlock state if every process in the set is waiting for an event to release that can only be caused by some other process in the same set. The following are some reasons for the occurrence of deadlock:

1. Mutual Exclusive: When a single process is used by two or more processes, means a single resource if used for performing the two or more activities as a shared based. But this is will also create a problem because when a second user request for the system resource which is being used by the user.

2 Hold and Wait: A single process may need two or more system resources. And suppose if a process have a single resource, and is waiting the second resource. Then process can't leave the first resource and waiting for the second resource. So that there will also be the condition of Deadlock.

3. No Preemption: If there is no rule to use the system resources. It means if all the system resources are not allocated in the manner of scheduling. Then this will also create a problem for a Deadlock because there is no surety that a process will release the system resources after the completion.

4. Circular Wait: When two or more requests are waiting for a long period of time and no one can access the resource from the system resources, then this is called as Circular Wait. For example if two or more users request for a Printer, at a same tifrie, they request to print a page. Then they will be on the Circular Wait means System will display a busy sign.

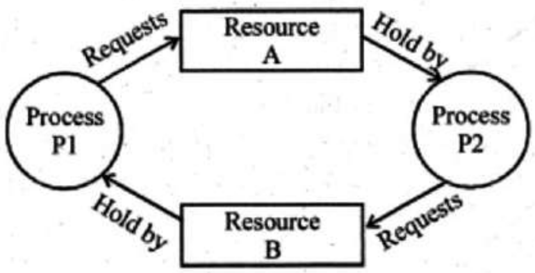


Fig. 6.5: Circular Wait

Example of occurrence of deadlock: There exists a set of waiting transactions {Processl,.,.,.,., Process 4} such that Process1 is waiting for data item that is held by Process2, Process! is waiting for a data item that is held by Process3, so on. None of the transactions can make progress in such a situation.

For avoiding a Dead Lock first of all we have to detect Dead-Lock means firstly we have to detect why and how a Deadlock has occurred and then avoid or solve the problems those are occurred due to occurrence of Deadlock.

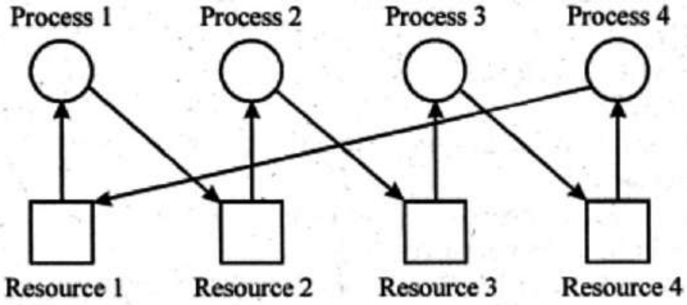


Fig. 6.6: Deadlock condition

6.4.2 Deadlock Prevention

To prevent the system from the deadlock state. There are two methods as follows:

1. **According to first method:** Transaction manager should not allow a transaction which goes in a waiting state for a data item. The following are some rules/protocols which help to prevent deadlock state:

    - Transaction manager should avoid the waiting cycles. He should use deadlock prevention protocol so that system never enters in a deadlock state.
    - One protocol to ensure that hold and wait condition never occurs. Each process must request and get all of its resources before it begins execution.
    - Each process can request resources only when it does not occupy any resources. If a process holding some resources, requests another resource (new resource) which is not allocated to it, then a process must release all the allocated resources so that it can access the new resource.
    - If there are multiple requests for a particular resource, then each process can access that resource in increasing order of priority.

2. **According to the second method**

    - If the system enters in a deadlock state, then we should try to recover a transaction from deadlock state by using deadlock detection and deadlock recovery techniques.

6.4.3 Deadlock Detection

1. If a system has no deadlock prevention and no deadlock avoidance scheme, then it needs a deadlock detection scheme with recovery from deadlock capability.
2. In the deadlock detection scheme, an algorithm is used to determine whether the system entered in a deadlock state or not
3. The deadlock detection algorithm should be invoked periodically.
4. The deadlock detection algorithm is as follows:

    Data Structure is as: Available [m]

Allocation [n, m] as in Banker's Algorithm.

Request [n, m] indicates the current requests of each process.

Let work and finish be vectors of length m and n, as in the safety algorithm.

The algorithm is as follows:

1. Initialize Work = Available

   For i= 1 ton do

   If Allocation (i) = 0 then Finish[i] = true else Finish[i] = false

2. Search an i such that

   Finish[i] = false and Request (i)≤ Work

   If no such i can be found, go to step 4.

3. For that i found in step 2 do: Work = Work + Alloeation(i)

   Finish[i] = true

   Go to step 2.

4. If Finish[i] ?true for some i then the system is in deadlock state else the system is safe.

6.5 DATABASE SECURITY AND INTEGRITY

The information stored in database is very valuable for an organization and it must be protected from unauthorized access and unwanted damage. Database security is a method to protect database from unwanted damages due to various reasons like unwanted access, physical damage, technical or mechanical damage, accidental loss, corruption of data etc. The database security allows or disallows users from performing actions on the objects contained within organizational database.

The database security is concerned with various policies which are framed by DBA to protect data. The DBA is responsible for the overall security of the database system. The DBA design overall policies, procedures and appropriate controls to protect and safe the data in database.

6.6 DATABASE SECURITY

1. Database security is the protection of database from internal and external threats.

2. Security is important because database is very valuable. All decisions of organization depend on data.
3. Data should be protected from unauthorized access.
4. Any corruption of data would affect the day-to-day operation.

6.6.1 Issues in Security

Sr. No.	**Issue**	**Description**
1	**Data Tampering**	Data cannot be modified; or viewed during transit. In case of distributed database where data moves between different sites, data can be modified during transit. So data security is important.
2	**Data Theft**	Data can be stolen from within the organization or from Internet so data security is important.
3	**Unauthorized Access**	Data should be protected from unauthorized users.
4	**Password Related Threats**	Usually users use their name, date of birth as password which can easily be traced there for security is important to protect data in the case of password threats.

6.6.2 Need/Requirement of Security

Sr. No.	**Issue**	**Description**
1	**Confidentiality**	Data should be confidential and user should be able to see the data he is supposed to see
2	**Authentication**	This is a process of verify the user's identity on database. Authenticity can be check after asking user name and password.
3	**Secure Storage**	After confidential data has been entry it should be store or protected in secure database

4	**Privacy of Communications**	DBMS should be able to secure the private data of use like health, employment and credit card number etc.
5	**Availability**	It is the duty of DBMS that data should be available to the user when it is required. It is possible only if it is secure and in authorized hands.
6	**Authorization**	After authenticity authorization get information about the operation that user may perform and the database user may access that may be • Read Authorization • Insert Authorization • Update Authorization • Delete Authorization • Drop Authorization • Alteration Authorization
7	**Integrity**	Integrity ensures that data is protected from deletion and corruption while it is store in database and while it is being transmitted over network.

6.6.3 Levels of Security

If database security has to maintain then it should be maintain at all levels. The security levels are:

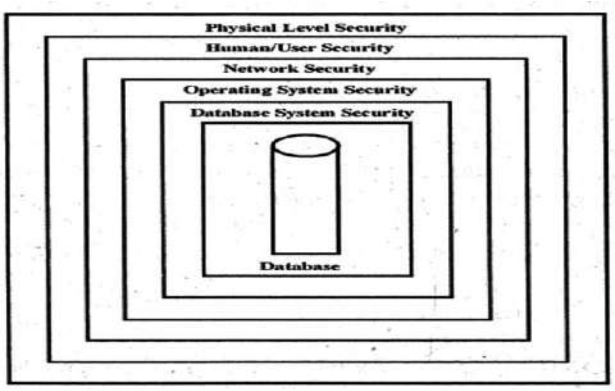


Fig. 6.7: Levels of Security

Sr. No.	Level	Description
1	**Physical Level**	Database must be secure from armed and weapons.
2	**Human**	Database should be secure from unauthorized users.
3	**Network**	Network security is important if database allow to access data , remotely. Network security features must be strong.
4	**Operating System**	If operating system is weak then no strong security features of a database can protect it from unauthorized access.
5	**Database**	The database should be secured from outside world (unauthorized people). Various database security methods are discussed in *section 9.1.4.*

6.6.4 Different Methods of Database Security

There are different methods for protecting data. These methods are as follows:

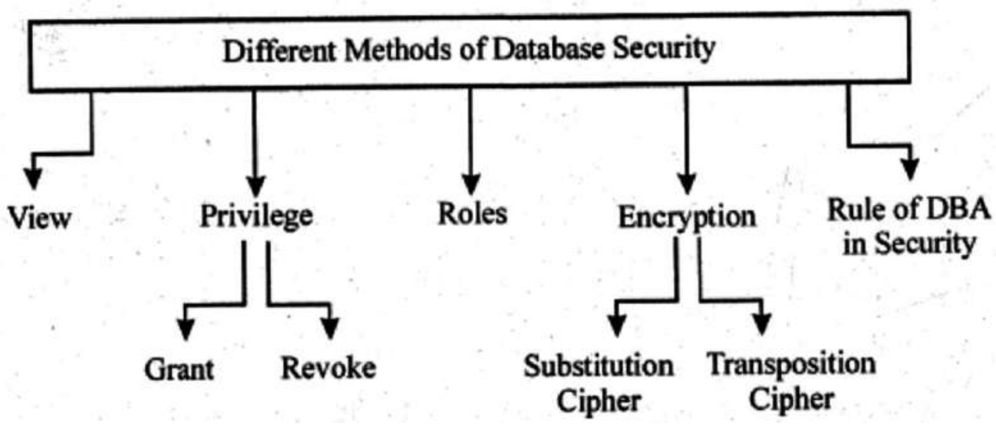


Fig. 6.8: Different Methods of Database Security

6.6.4.1 View

1. View is logical table based on one or more tables.
2. Table on which view is made is called base table.
3. It is just like a table but does not store the data.
4. Using views, we can restrict the set of rows and columns of table.
5. User can only see the provided row and columns.

6. All operations performed on view effect the base table.

7. **Syntax**

   Create view VI as

   Select Emp_No, Name, Dept from EMP

   Where City ='Chc';

   VI is a view and all DML commands can be used on view VI.

6.6.4.2 Privilege

1. Privilege is a permission given to the user to access the database objects.
2. After getting the privilege, user can use any SQL command.
3. With the help of privileges, we can perform the following tasks:
   - Create a table
   - Select rows from table
   - Insert new record in table
   - Update data of table
   - Delete data from table.
4. DBA give privileges to the users.
5. There are two commands used to give and withdrawal the privileges which are as follows:

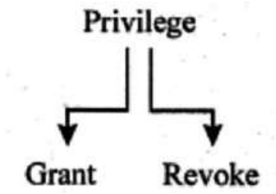


Fig. 6.9: Two Commands in Privilege

(a) Grant

- Granting a privilege to user means giving permission to user for some specific task.
- Granting of privileges to users is done by Grand Command.
- On clause is used to specify the object name.
- To clause is used to specify user name.

- ***Syntax:***

  Grant <privilege> on <object name>to <user name>;

- ***Examples of Grant Command :***

  (a) Grant to user 1 for select the records on employee table, then the query will be:

  Grant Select on Emp to Userl;

  (b) Grant to all users for select the records on employee table, then the query will be:

  Grant Select on Emp to Public;

  (c) Grant all privilege to all the users on employee table, then the query will be:

  Grant all on Emp to Public;

(b) Revoke

- Revoke withdraws granted privileges.
- Revoke takes back all the privileges given to the users.
- ***Syntax:***

  Revoke <privilege> on <table name> from <user name>;

- ***Example of Revoke Command:***

  Withdraw the select grant permission on employee table from user 1, then the query will be:

  Revoke Select on Emp from Userl;

6.6.4.3 Roles

1. Role is a mechanism that Is used to provide authorization.
2. It is a group of privileges.
3. A single person or group of persons can be granted a role.
4. Using roles, DBA can manage access privilege more easily.
5. Suppose there are two roles in college database.,

   (a) **Role 1:** Which have created, after, drop, insert, select, update, and delete privileges?

   (b) **Role 2:** Which have select privilege only?

6. Role 1 will be provided to all the users related to staff. Role 2 will be provided to all the users related to students.

7. When a new staff user will joint it will be provided Role 1 and get all the privileges related to Role1. When a new student user will joint it will be provided Role 2 and get all the privileges related to Role 2.

6.6.4.4 Encryption

1. Encryption is a technique by which we can convert the data in coding form.
2. This process of converting is called Encryption.
3. Encrypted key Is needed to convert the plain text to cipher text
4. Plaintext: The message or data which Is to be converted.
5. Cipher text: The converted data.
6. Cipher text is then transmitted to the network.
7. Decryption key is needed to decode the cipher text back to plain text.
8. The process of converting cipher text (coded data) into plain text (original data) is called decryption.

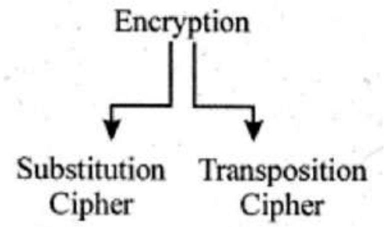


Fig. 6.10: Techniques Used in Encryption

Techniques Used in Encryption

I. Substitution Ciphers

- In this technique, each letter Is replaced by another letter.
- For example: A Is replaced with D, B is replaced with E, C with F and so on such as Attack becomes OWWDFN.
- This technique is not secure because It can be guess easily.

II. Transposition'Ciphers

- In this technique, letters are re-ordered not replaced.

- We arrange the letters In different order but not replace these letters with new letters.
- For example: theft can be .coded as eftth.

6.6.4.5 Role of DBA *in* Security

- Database Administrator plays an important role to provide security.
- DBA is responsible to provide the overall security to the database.
- DBA has a special account which is called system account.
- DBA create new user and provide them user_id and password.
- He creates roles and assigns privileges to the role.
- DBA can assign or change the role of a user.
- DBA create views and assign to the user.
- DBA can also check the log file which describes the detail of a user.
- DBA can detect the violation area.

6.7 BASIC CONCEPTS OF SECURITY

1. Security. Policy

The purpose of a security policy is to elaborate the three general security objectives of secrecy, integrity and availability, in the context of a particular system.

In general, security policy is largely determined within an organization rather than imposed by mandate from outside. This is particularly so in the integrity and availability areas. There are three main objectives to consider while designing a secure database application:

Secrecy: It is concerned with improper disclosure of information. Information should not be disclosed to unauthorized users. For example, a student should not be allowed to examine other students' grades.

Integrity: It is concerned with improper modification of information or processes. Only authorized users should be, allowed to modify data. For example, students may be allowed to see their grades, yet not allowed to modify them.

Availability: It is concerned with improper denial of access to information. The term denial of service is also used as a synonym for availability. Authorized users should not be denied access. For example, an instructor who wishes to change a grade should be allowed to do so.

2. Prevention.

Prevention ensures that security breaches cannot occur. The basic technique is that the system examines every action and checks its conformance with the security policy before allowing it to occur. This technique is called access control.

3. Detection

Detection ensures that sufficient history of the activity in the system is recorded in an audit trail, so that a security breach can be detected after the fact. This technique is called auditing.

4. Assurance

Security mechanisms, whether preventive or detective in nature, can be implemented with various degrees of assurance. Assurance is directly related to the effort required to threaten the mechanism. Low assurance mechanisms are easy to implement but also relatively easy to disrupt. Subtle bugs in system/application software have led to numerous security breaches. On the other hand, high assurance mechanisms are notoriously difficult to implement. They also tend to suffer from degraded performance.

Note: *Prevention is the more fundamental technique. An effective detection mechanism requires a mechanism to prevent improper modification of the audit trail. Moreover, detection is ultimately useful only to the extent that it prevents improper activity by threatening punitive action.*

6.8 DATABASE INTEGRITY

1. It concerned with correctness and consistency of data.
2. This is a main task in multi-user database.
3. Integrity violation may arise from many different sources like:
4. Typing error by data entry clerks.j

5. Logical errors in application.

6. Error in system software.

7. Result of all these violations in data corruption.

8. Database integrity is responsible for monitoring and detecting integrity violations.

9. When integrity violation occur, system then take following actions:
   - Rejection the operation
   - Reporting violation
   - Returning the database to consistent state.

10. ***The following are the database integrity rules:***

(a) **Domain Integrity Rules:** It is used to maintain the correct value of attributes, i.e. for age attribute integrity rules should be in integer and should be positive and it should be possible to specify upper and lower bounds for values of age.

(b) **Entity Integrity Rules**

- It is used to preserve the key uniqueness. (Primary key)
- It specifies that all entries are unique.
- There is no NULL entry in primary key.

(c) **Referential Integrity Rules**

- These rules are concerned with maintaining the correctness and consistency of relationship among relations.
- It specifies that foreign key must have either a NULL value or match with primary key value.
- It ensures not to enter invalid value.
- Referential integrity rule make possible not to delete a row in one table whose matching foreign key value is existing.

6.9 RECOVERY

1. Recovery is the process of restoring the database after the failure.

2. Failure may be the result of system crash due to hardware or software.

3. Recovery is the responsibility of DBA (Database Administrator).

4. Various procedures and strategies (backup and recovery)are used in recovery to protect the database.
5. Whenever a transaction is submitted to a DBMS for execution, the system is responsible for making sure that either
   - All the operations in the transaction are completed successfully and their effect is recorded permanently in the database or.
   - The transaction has no effect on the database or on any other transactions.
6. Before understand the concept of recovery, it is important to understand the cause of failures.

6.9.1 Cause/Reason of Failure

Failures are generally classified as transaction, system, and media failures. There are several possible reasons transaction to fail in the middle of execution:

I. System Crash/Computer Failure

1. In system crash, system hangs up and needed to reboot.
2. A hardware, software, or network error occurs in the computer system during transaction execution.
3. In this case, the data which is in main memory is lost and transaction rolls back.
4. The permanent data which is in permanent storage devices are not affected by system crash.
5. The reasons of this failure are hardware, database software and operating system.

II. User Error

1. User drops a full table.
2. User deletes a record.
3. User purchases the hardware of poor quality.

III. Statement Failure

1. A transaction which has multiple statements and one statement might fall.
2. Result will be error message by database software or operating system.

3. The recovery in this case will be automatic because the transaction will rollback and user can re-execute the statement again,

4. Reasons may be
    - Selecting rows from- a table which doesn't exists.
    - Inserting records by there is not enough space.

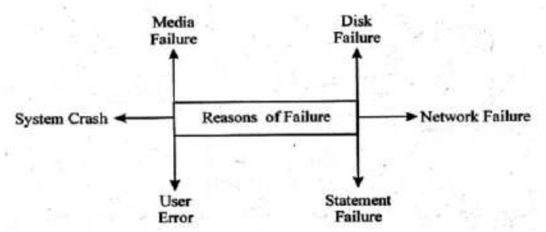


Fig. 6.11:-Reasons of Failure

IV. Network Failure

1. Network failure effect distributed database where data is .coming from different sites.

2. Reason may be
    - Client server configuration
    - Communication software failure

V. Disk Failure

1. This may happen during a read or a write operation of the transaction.

2. Some disk blocks may lose their data because *of* a read or write mal function or because of a disk read/write head crash.

VI. Media Failure (Disasters)

1. This is most dangerous failure.

2. It is very difficult to recover the data effect from these failures.

3. It has only one solution is to take regular backup, i ;,

4. It is because of fires, floods and earthquakes. We will discuss disaster and its management techniques in *section 9.3.*

6.9.2 Terms Used in Recovery Process

1. To understand the concept of recovery, we must understand the concept of main memory (RAM) and secondary memory (Hard disk).
2. Each disk is partitioned into blocks.
3. Block in main memory is called buffer block or disk buffer.
4. A block in disk is called physical block.
5. When a transaction starts, data is transfer form physical block to buffer block.
6. When a transaction ends, data is transfer back from buffer block to physical block.
7. ***There will be two main operations which are as follows:***

   (a) ***Input Operation:*** Transfer data from physical block to buffer block.

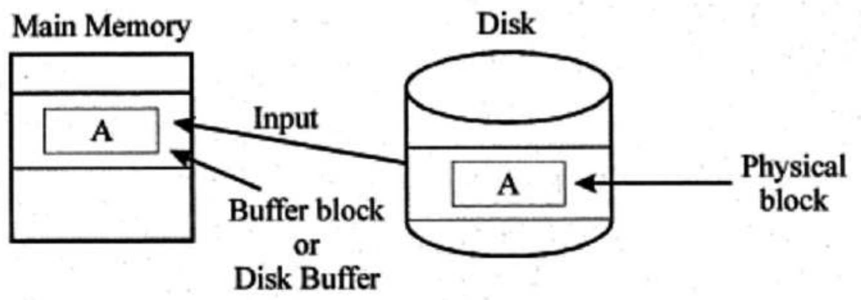


Fig. 6.12: Input Operation

   (b) ***Output Operation:*** Transfer data from buffer block to physical block. .Main MemoryDisk

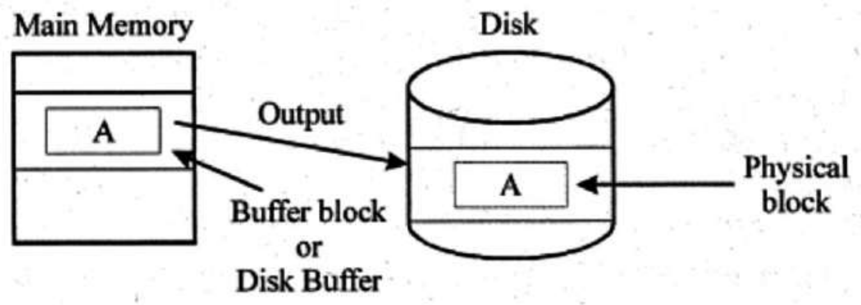


Fig. 6.13: Output Operation

8. Each transaction has a private working area where transaction executes. This area is created and removed according to the transaction.

   For example: we want to perform the following transaction.

Read (A, .a)

a = a-100

Write (A, a)

Three operations will be performed to complete the above transaction as follows:

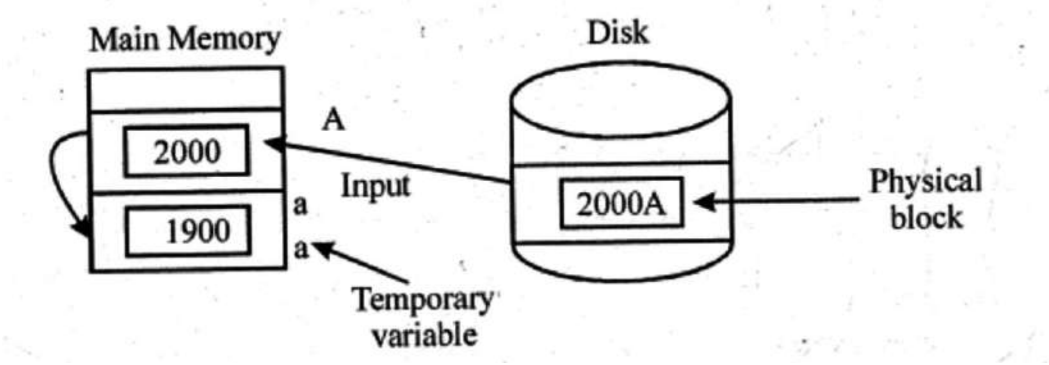
(a) Read (A, a)

(b) Write (A, a)

(c) Output (X)

(a) Read (A, a)

- Find the database item 'A' in block X of disk.
- Then, transfer database item 'A^5 in block X from disk to main memory.
- In main memory, database item 'A' will be copy in temporary variable "a^5.
- The arithmetic operation will be performed as



Fig, 6.14: Read Operation

(b) Write (A, **a)**

- The data item 'A' belongs to ^{C}X' block.
- If 'X^5 is in main memory, then data of temporary variable ^{c}a' is copy to data item 'A'.
- If 'X' is not in main memory then input operation performed again.

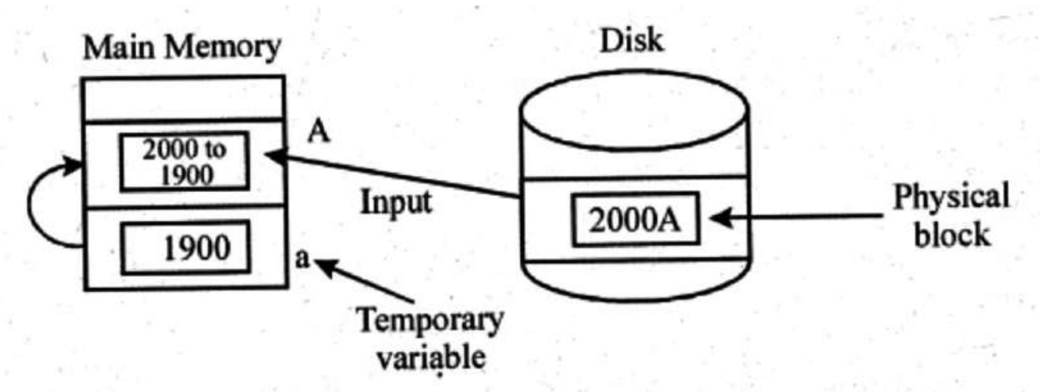


Fig. 6.15: Write Operation

(c) Output X

- After performing the write operation, block 'X' is now written in disk.
- Now, output operation is performed.
- Database item 'A' is copy to disk permanently.

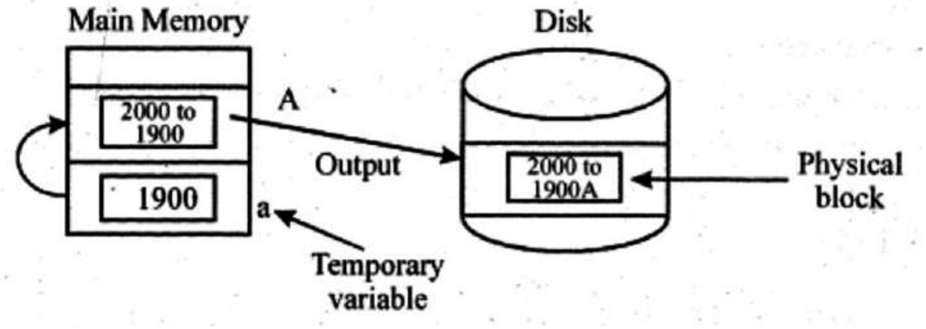


Fig. 6.16: Output Operation

6.10 ATOMICITY OF TRANSACTION

1. Atomicity means either all operations of the transaction are reflected properly in database or none.
2. It ensures that either a transaction ends with committed state or rolled back state.
3. Committed state comes, when transaction end with success and database reaches in a new consistent state.
4. Aborted state comes, when transaction end with failure and database restores the previous consistent state. A failed transaction enters in the aborted state and 'rolled back' or 'undone'.

5. According to atomicity, we cannot perform half operations of any transaction.
6. It means we should perform all the operations of the transaction or we should not perform any operation.
7. In short, atomicity means either 100% modification on 0% modification.
8. Ensuring atomicity is the responsibility of the database system itself. It is handled by a component called the *transaction management component,*
9. *For example:*

   There are two accounts 'A' and 'B'. Account 'A' contains Rs.2000. Account 'B' contains Rs.1000.

   The transaction Tl transfers Rs.100 from account "A" to account 'B'.

Transaction Tl

A is account	Read (A, a)	A = 2000, a = 2000
a is temp variable	a=a-100	a -1900
B is account	write (A, a)	A =1900
b is temp variable	output (A, X)	
AX, BX buffer block -	Read (B,b)	B = 1000, b= 1000
	b = b + 100	b =1.100.
	Write (B,'b)	B = 1100
	Output (BX)	

Example 6.1: Example of Atomicity

In the transaction Tl, if a system crash occurs after the output (AX) but before output (BX), then the output will be:

A is account	Read (A, a)	A = 2000, a = 2000
a is temp variable	a = a- 100	a= 1900
B is account	write (A, a)	A =1900
	output (A, X)	

After the reduction of Rs.100 from account 'A', system crashes. Now, system is in inconsistent state and there is a loss of Rs.100.

Account A has Rs. 1900

Accounts has Rs. 1000

To recover this loss, if we execute transaction Tl again, then database will again in inconsistent state because account 'A^{9} has Rs.1900 and account 'B^{9} has Rs.1000. We cannot recover this Rs.100 again if we re-execute the transaction Tl after the modification. This example gives the concept of atomicity which means either 100% modification on 0% modification.

10. Atomicity can be achieved by transferring the output in storage without modifying database. There are two ways to achieve atomicity as follows:

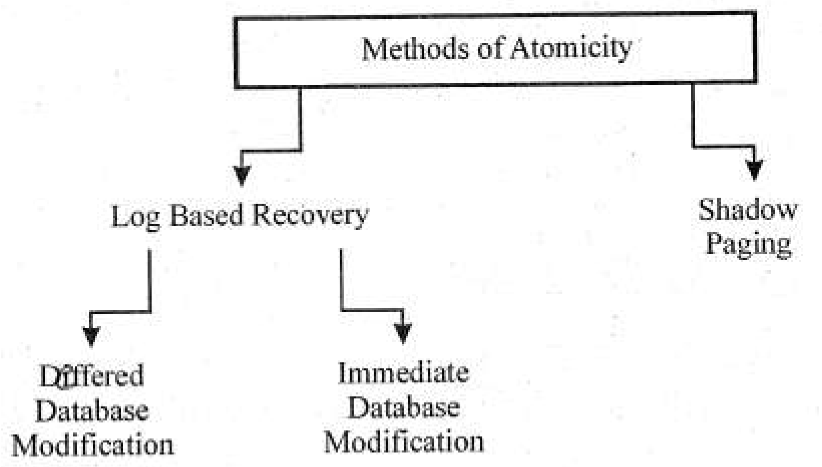

Methods of Atomicity
Log Based Recovery
Shadow Paging
Differed Database Modification
Immediate Database Modification

Fig. 6.17: Methods of Atomicity

6.10.1 Log Based Recovery

1. A log file is maintained in Log based recovery.
2. A log file is used to maintain the record of all the operations of database.
3. Basically, it is a sequence of log record.
4. The following are the different types of log records:

1	**<Start> Log Record**	It contains the information about the start of each transaction and used to differentiate the state of

		different transactions. <Ti start>
2	**<Update> Log Record**	It updates the data item under the transaction. <Ti, Xj, VI, V2> Ti is transaction Xj is data item VI is old value V2 is new value
3	**<Commit > Log Record**	It tells that the transaction is completed. When transaction Ti successfully completed, then log record is store in the log file.
4	**<Abort> Log Record**	It tells that the transaction is not completed When transaction Ti is not successful completed then it will be abort and log record is stored in log file.

5. After the log record, database can be modified.
6. We can recover the data from the log record. In the previous example discussed in atomicity (example 9.1), when a system crashes after the output (AX) but before output (BX), there is a loss of Rs.100 after the execution of transaction Ti.
7. We can recover that loss by copying the old value of ' A' Rs.2000 from log record to database. It is called undo operation.
8. The following are the two techniques for log based recovery:
   - Differed Database Modification
   - Immediate Database Modification

6.10.1.1 Deffered Database Modification

1. During write operation, the modified value of local variable is stored in log record not in database.
2. After the successful execution of transaction, the modified value is copied in the database.

3. If the transaction fails to complete, then the modified value and log record in ignored. In this case, the value of data item maintains its old value.
4. For example:

There are two accounts 'A' and 'B'. Account 'A' contains Rs.2000. Account 'B' contains Rs. 1000. Account 'C' contains Rs.3000.

The transaction TI transfers Rs.100 from account 'A' to account 'B'. The transaction T2 withdraws Rs.100 from account 'C'.

Transaction Tl

Read (A, a)
= a-100
write (A, a)
Read (B, b)
= b+100
Write (B, b)

Transaction T2

Read (C, c)
= C-100
Write (C,c)

The transactions Tl and T2 will execute in sequence.

In the deferred database modification, data will be modified in the database if log contain both <Ti, start> and <Ti, commit> otherwise transaction will rollback and updation will be cancelled.

<table>
<tr><th colspan="3">During Transaction Tl</th></tr>
<tr><th>Tl</th><th>Log</th><th>Database</th></tr>
<tr><td>Read (A, a)</td><td><T1, start></td><td>(Buffer)</td></tr>
<tr><td>a = a- 100</td><td>^TUA, 1900></td><td></td></tr>
<tr><td>Write (A, a)</td><td><T1,B, 1100></td><td></td></tr>
<tr><td>Read (B, b)</td><td><T1, commit></td><td>A = 1900</td></tr>
<tr><td>B = b+100</td><td></td><td>B = 1100</td></tr>
<tr><td>Write (B,b)</td><td></td><td></td></tr>
<tr><th colspan="3">During Transaction T2</th></tr>
</table>

T2	Log	Database
Read (C, c)	<T2, Start>	
c = c-200	<T2, C, 3800>	C = 28'00
Write (C, c)	<T2, commit>	

Example 6.2: Example of Deferred Database Modification

6.10.1.2 Immediate Database Modification

1. This technique allows database modification while transaction is still in active state.
2. If the system crashes, then the transaction abort and log record is used to restore value.
3. The log record can restore the value with the help of undo operation. <Ti,Xj,Vold,Vnew>

   Ti = Transaction id

   Xj = Data item

   Void = Old value

   Vnew = New value
4. For example:

   There are two accounts 'A' and 'B'. Account 'A' contains Rs.2000. Account 'B' contains Rs.1000.Account 'C' contains Rs.3000.

   The transaction Tl transfers Rs. 100 from account 'A' to account ^{C}B'. The transaction T2 withdraws Rs. 100 from account 'C'.

During Tl		
Tl,	**Log**	**Database**
Read (A, a)		
a = a- 100	<Tl,Start>	
Write (A, a)	<T1,A,2000$_?$ 1900>	.A = 1900
Read (B, b)		

B = b + 100	<T1,B, 1000, 1100>	B = llOO
Write (B, b)		
	<T1, Commit>	
During Transaction T2		
T2		
Read(C, e)	<T2,Start>	
c = c - 200	<T2, C? 4000, 3800>	C = 3800
* Write (C, c)		
	<T2?Commit>	

Example 6.3: Example of Immediate Database Modification

6.10.2 Shadow Paging

1. Database is divided into blocks.
2. Blocks are of fixed lengths.
3. We use a page table. The page table has entries for each database table.
4. Shadow paging technique use two page tables during transaction execution.
   - Current page table
   - Shadow page table
5. At beginning of transaction both pages are identical.
6. Each page table entry contains a pointer to a page on disk.
7. Current page table may change during write operation.
8. Shadow page table never change during transaction.
9. There are two cases of recovery in the shadow paging technique which are as follows:
   - ***Case 1:*** If system crashes before the successful completion of transaction, then current page table will removed. This is just like undo operation.lt means if system fail before commit transaction then it will get the previous state of data from shadow table.

- ***Case* 2.** If system crashes after the successful completion of transaction, then current page table will become the shadow page table. It means if system fails after commit transaction then it will recover the data from shadow table,

6.11 DISASTER MANAGEMENT

1. *"A disaster can be defined as an occurrence either nature or manmade that causes human suffering and creates human needs that victims cannot alleviate without assistance".*
2. ***There are four main types of disaster as follows:***
   - **Natural disasters:** Natural disasters include floods, hurricanes, earthquakes and volcano eruptions that can have immediate impacts on human health.
   - **Environmental emergencies:** Environmental emergencies include technological or industrial accidents, usually involving hazardous material, and occur where these materials are produced, used or transported.
   - **Complex emergencies:** Complex emergencies include a break-down of authority, looting and attacks on strategic installations. Complex emergencies include conflict situations and war.
   - **Pandemic emergencies:** Pandemic emergencies include a sudden onset of a contagious disease that affects health but also disrupts services and businesses, bringing economic and social costs.

Disaster Management

1. Disaster management is also known as emergency management.
2. It avoids both natural and man-made disasters.
3. *"Disaster management can be defined as the organization and management of resources and responsibilities for dealing with all humanitarian aspects of emergencies, in particular preparedness, response and recovery in order to lessen the impact of disasters."*

4. It involves preparedness before disaster, rebuilding and supporting society after natural disasters such as, earthquakes, drought, tsunami etc.

5. The following are the techniques to manage disaster:

   - **Disaster Management Teams:** Worldwide, governments, business and non-business organization are setting up disaster or crisis management teams in order to manage the disaster. The disaster management teams are broadly divided into three parts namely: the policy team, the management team and the liaison team.
   - **Systematic Planning:** Disaster management involves systematic planning to avoid a disaster. If disaster occurs, then systematic planning is required to overcome the crisis arising out of disaster. It indicates, what to do, when to do, how to do and who is to do certain activities to manage and overcome the problems of disaster.
   - **Training to Manpower:** There is a need to provide proper training to the disaster management personnel to manage a disaster effectively. The training will help to develop and improve the disaster management skills in the people.
   - **Suitability:** Disaster management is required before and after a disaster. It is suitable before a disaster in order to avoid a disaster, or to caution the people. It is also very much required after a disaster takes place, in order to undertake rescue, relief and rehabilitation measures at the time of floods, earthquakes.
   - **Stability:** Normally, disaster management teams lack stability. They are formed just prior to a disaster in order to avert it, whenever possible. But there should be some permanent disaster management teams.

Questions

1. What is Concurrency Control? Why is it required?
2. Explain the Concurrency Control schema based on timestamp protocol.
3. What are AICD properties of transaction? Explain the uses of each.
4. What are the different locking techniques for Concurrency Control?
5. What is a lock? Differentiate between exclusive and shared lock. Give suitable examples also.
6. What is time stamping?
7. Explain Concurrency Control without locking.
8. What is deadlock? Give an appropriate example.
9. Explain the following:
   (a) Deadlock prevention
   (b) Deadlock detection
10. What are the necessary conditions for deadlock?
11. Define Database Security. What are the various issues addressed by it?
12. What are the different security mechanisms?

13 What are the different access rights that may be given to the users of a database?

14 What do you mean by data encryption? How is it achieved?

15 Explain the following with suitable example:
   (a) Entity Integrity Constraint.
   (b) Domain integrity constraint

16 What is the role of DBA in security?

17. What is the concept of recovery in database? What are the techniques used for it.

18 What is deferred update and immediate update?

19 What is shadow paging?

20 Explain transaction rollback.

21 Explain the recovery and atomicity of transaction in detail with suitable examples.

22 What is log based recovery?

23 Discuss the disaster management in detail.

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 7: SQL

7.2 INTRODUCTION TO SQL

- SQL stands for "Structured Query Language".
- It is a widely used database language.
- It has been adopted as the standard relational database language.
- It can be pronounced as "SQL" or "SEQUEL".

- It was first introduced as a commercial database system in 1979 by Oracle Cooperation.
- SQL is different from other programming languages like C, C++, Java, Visual Basic etc.
- Unlike other languages, there is no need to specify the sequence of steps to perform any particular task, SQL statements directly provides the desired result.
- It is a non-procedural, pure English language rather than coding language but it has fixed syntax (structure).
- It processes set of records rather than one record at a time.
- It is made up of various commands and used to define, access and manipulate data in RDBMS.
- It is just not for query the database but it can do much more.
- All the programs written in SQL are portable. They can be moved from one database to another with little modification.\
- All the major relational database management system support SQL. SQL has proved to be very effective for heavy databases.

Functions/Tasks of SQL

- SQL can define the structure of database.
- It can execute the queries against a database.
- It can create new databases.
- It can create new tables in a database.
- It can insert records in a database.
- It can update records in a database.
- It can delete records in a database.
- It can retrieve data from a database.
- It can create views and stored procedures in a database.
- It can set permissions for users so that they can use the database.
- It can specify the security constraints in the database.
- It can create, replace and alter the objects.
- It guarantees the database consistency and language.

7.2 INTRODUCTION TO SQL*PLUS

- SQL*PLUS is command line tool.
- It allows user to type SQL statements to be executed directly against an Oracle database.

- With the help of SQL*PLUS, a user can perform the following tasks:
- User cap access Oracle databases with command procedures.
- User can interactively use the SQL commands (Enter, Edit, Store and Retrieve).
- User can produce reports.
- Access and copy data between SQL databases.
- Send messages and accept responses from an end user.
- List columns of any table.

7.3 DIFFERENCE BETWEEN SQL AND SQL*PLUS

Sr. No.	**SQL**	**SQL*PLUS**
1	It is a standard language to access RDBMS (Relational database management system)	It is a tool to implement SQL. It has a command line interface.
2	SQL commands are terminated by semicolon (;).	SQL*PLUS commands do not need semicolon (;) for termination. It only needs continuation character (-).
3	Data manipulation is possible in SQL; Data manipulation language/ command (DML) is used to modify the data.	Data manipulation is not possible in SQL*PLUS.
4	Commands are stored in buffer.	Commands are not stored in the buffer.
5	Commands are entered in one line or more than one line.	Commands are entered in one line at a time.
6	For formatting purpose, SQL use various functions.	For formatting purpose, SQL*PLUS use various formatting commands.
7	SQL commands are to be executed again and again.	SQL*PLUS commands remain in effect until any new command overwrite it.

7.4 STARTING SQL*PLUS

We can start the SQL*PLUS with either of the two methods. The following steps should be considered to start the SQL*PLUS:

Method 1

Step 1: Locate the **SQL PLUS** (Oracle Shortcut) on the desktop.

Step 2: Press **<Enter>** or **'double clock'** on the 'Oracle Shortcut'.

Method 2

If the SQL PLUS (Oracle Shortcut) is not available on the desktop, then follow the following steps:

Step 1: Click on **'Start'** □ **‘All Programs’** □ **‘Oracle-OraDb10g_home3'□ 'Application Development'□ 'SQL PLUS'.**

Step 2: Enter the **'User Name'** in the dialog box. The **'User Name'** should be 'Scott'.

Step 3: Enter the **'Password'** in the dialog box. The **'Password'** should be **'Tiger'**.

Step 4: Enter the **'Host String'** in the dialog box.

Step 5: Press the **"OK"** button to complete the Oracle log in process, then the SQL*PLUS prompt (SQL>) will appear.

7.5 DATATYPES

- Oracle uses the tables for storing and maintaining the information. Table consists of rows and columns.
- Each column contains only one type of data which we must define.
- A data type is an attribute that specifies the type of data.
- In short, a data type is a classification of a particular type of data (information).
- When we create a table, we must specify a data type for each of its columns.
- The following table show a list of different data types commonly used in Oracle:

Sr. No.	Data Type	Description
1	Char (n)	A fixed length character string with user specified length. The characters can be used in it. n = number of characters
2	Varchar (n)	A variable length character string with user specified length. n = number of characters
3	Int	A numeric value can be specified. Character cannot insert.

4	Numeric (P,D)	A floating number can be inserted. P indicate total digits. Whereas D indicate number of digits after decimal. For example: (3,1) allows 44.5 to be inserted but not 444.5 or 0.32.
5	Float (n)	A floating number with decimal value up to n number.
6	Date	It can contain date with year, month and day of month.
7	Time	It can contain time in hours, minutes and seconds.

7.6 PARTS OF SQL

- SQL consists of various commands. SQL commands are instructions used to perform the various operations like create, delete and manipulate the data in database.
- SQL commands are helpful in searching the data, drop the table, add data to table, set permissions for users etc.
- SQL is an easy, understandable and unified language.
- SQL languages/commands/statements are categorized into six major parts based on their functionality which are as follows:

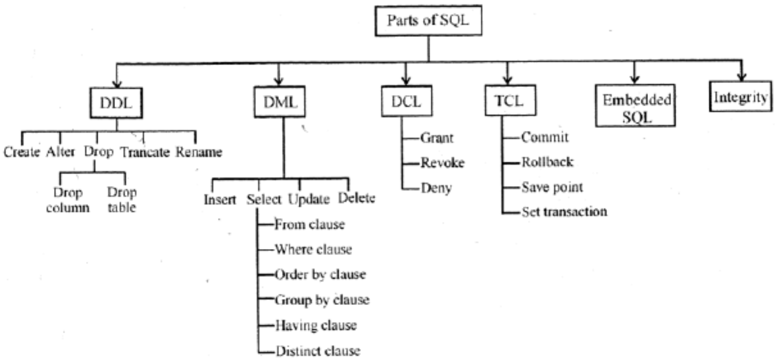


Fig. 10.1: Parts of SQL

7.6.1 DDL (Data Definition Language)

It is used for defining data structures. These SQL commands are used for creating, modifying and dropping the structure of database objects (relations).

These commands basically create, modify and drop the relations (tables) used in the database.

The following are the various DDL commands:

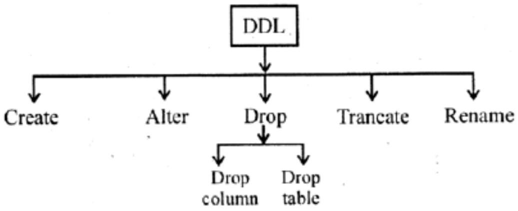


Fig. 10.2: Parts of DDL

1. Create: The create table command is used to create a new table. It creates the relation (table) in a database. It includes its name, names and attributes of its columns. One can create any number of columns with this command. If we want to add or remove the columns after creating the table then we use alter table.

Syntax of Create New Table:->

```
SQL>CRE^VTE TABLE table_name
(
column_name1 data type,
column_name2 data type,
…….
column_nameN datatype
);
```

Note: We can also create a table from existing table by copying the existing table's column.

Syntax of Create Table from Existing Table:->

```
SQL> CREATE TABLE new_table
        As (SELECT * from old_table);
```

Examples of Create Command:-

1. We want to create a table 'STUD' in SQL.

   Then the query will be:

   ```
   SQL> CREATE TABLE STUD
   (
   NAME char (40),
   CLASS char (5),
   ROLL NUMBER (8)
   );
   Table created
   ```

2. We want to create a table 'BMP' in SQL. ***(Mostly queries of this book are based on this table 'EMP')***

```
SQL> CREATE TABLE EMP
(
ENAME char (15),
DEPTNO int,
JOB char (10),
EMPNO int,
SAL int,
HIREDATE int,
MGR int,
CITY char (10),
COMM int
);
Table created
```

2. **Alter:** It alters the structure of table from database. It alters the table along with the columns. One can add one more than one column in a particular table with alter command. With this command, filed type can be changed or a new field can be added. It is used to enable or disable the integrity constraint. It is used to modify the column values and constraints.

Syntax of Alter Command:

```
SQL> ALTER TABLE table_name
ADD/MODIFY/DROP column_name datatype;
```

Examples of Alter **Command:**

1. To add a column (DOB) in an existing table 'BMP'. Then the query will be:

```
SQL> ALTER TABLE EMP
ADD DOB date;
Table altered
```

2. To add multiples columns (DOB and MOBNO) to an existing table 'EMP'. Then the query will be:

```
SQL>ALTER TABLE EMP
ADD (DOB date, MOBNO (11));
```

Table altered

3. Drop: With the drop command, we can drop the columns from table or we can remove the table. It drops the column or constraints from the table. It deletes the string of a table. It cannot be recovered. It use with caution. Drop operation is used with the alter table command. It removes single column or multiple columns.

(a) Dropping Column: If we want to remove column, then we use drop operation with alter table command.

Syntax of Dropping the Column:

```
SQL>ALTER TABLE table_name
DROP COLUMN column_name;
```

Examples of Dropping the Column:

1. To drop a column 'City' in an existing table 'EMP'. Then the query will be:

```
SQL>ALTER TABLE EMP
DROP COLUMN CITY;
Table altered.
```

2. To drop multiple columns (Hiredate and City) in an existing table 'EMP'. Then the query will be:

```
SQL>ALTER TABLE EMP
DROP COLUMN (HIREDATE, CITY);
Table altered.
```

(b) Dropping Table: If we want to remove the table, then there is no need to use it with alter table command. We can directly remove one or more columns with drop table command.

- This command removes one or more table definitions and all data, indexes, triggers, constraints and permission specifications.
- If we drop a table with drop table command, it deletes all rows from that particular table. The table structure is also removed from the database and it cannot get back.

Syntax of Dropping the Table

```
SQL> DROP TABLE table_name;
```

4. Truncate: It removes all the records from a table and memory. It releases the memory occupied by the records of the table. Data cannot be recovered after using the truncate command.

Truncate command removes all the rows from a table.

Syntax of Truncate Command:

SQL> TRUNCATE TABLE table_name;

Example of Truncate Command:

We want to delete all rows from the table 'EMP'. Then the query will be:

SQL> TRUNCATE TABLE EMP;

5. Rename: It is used to rename the old table with a new name. The data will remain same, only name of table will be change with 'Rename Command'.

Syntax of Rename Command:

SQL> RENAME <Old Table_Name>to<New Table_Name>;

Example of Rename Command:

If we want to change the name of table 'EMP' to new name 'EMPLOYEE'. Then the query will be:

SQL>RENAME EMP TO EMPLOYEE;

Note: We use drop command for tables and delete command for records.

7.6.2 DML (Data Manipulation Language)

- These commands are used for inserting, retrieving, deleting and modifying the data in a relation or a table.
- It includes the query language based on both relational algebra and tuple relation.
- These commands do not implicitly commit the current transaction.
- The folio wing are the various DML commands:

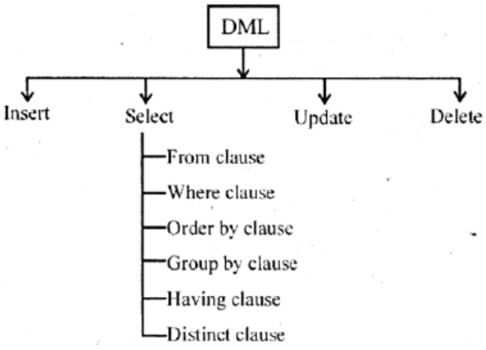


Fig. 10.3: Parts of DML

1. Insert

- When a new table is created, there is no data in the table.
- Insert command is used to insert the records in the new table.

- Insert command is used to add records to an existing table.
- 'Values clause' is used with inset command. This command will insert value in all the columns of a table in sequence.

Syntax of Insert Command:

```
SQL> INSERT INTO table_name
        VALUES (value1, valueZ, valueS,.....);
```

OR

```
SQL> INSERT INTO table_name (column1, column2, column3,......)
        VALUES (value1, value2, value3,....);
```

Examples of Insert Command:

1. Insert record in different order. Then the query will be:

```
SQL> INSERT INTO EMP (name, city, salary, emp_no)
VALUES ('Mona','Nba', 4500, 4);
```

2. Insert the Null value in record. Then the query will be:

```
SQL> INSERT INTO EMP
VALUES (3,'Mona', Null, 4000);
```

3. Insert the records in selected columns. Then the query will be:

```
SQL> INSERT INTO EMP (name, city)
VALUES ('Mona', 5000);
```

4. Insert the values in the table 'EMP'. Then the query will be:

```
SQL> INSERT INTO EMP VALUES ('Nidhi',20,'Clerk',6258,900,9-5-83, 6801,'Chd');
SQL> INSERT INTO EMP VALUES ('Aastha',30,'SaIesman',6388,1500,1-12- 89, 6587, 'Delhi', 300);
SQL> INSERT INTO EMP VALUES ('Sachin',30,'Salesman',6410,1350,25-1-92,6587,'Pta',500);
SQL> INSERT INTO EMP VALUES ('Rohit',20,'Manager',6455,2875,27-12-91,6728,'Nba');
SQL> INSERT INTO EMP VALUES ('Rahul',30,'Salesman',6543,1350,28-5-87,6587,'Nba',1400);
SQL> INSERT INTO EMP VALUES ('Aditya',30,'Manager',6587,2750,17-8-
86,6728,'Pta');
SQL> INSERT INTO EMP VALUES ('Siddharth',10,'Manager',6671, 2550,29-9-
80,6728,'Chd',Null);
SQL> INSERT INTO EMP VALUES ('Kunar,20,'Analyst',6677,3000,8-12-82,
6455,'Delhi',Null);
SQL> INSERT INTO EMP VALUES ('AkhiP,10,'President',6728,5000,2-11-
85,Null,'DeIhi',NulI);
SQL> INSERT INTO EMP VALUES ('Prathiba',30,'Salesman',6733,1600,4-6-
85,6587,'Pta',0);
SQL> INSERT INTO EMP VALUES ('Manmeet',20,'Clerk',6765,1050,11-1-
84,6677;'Ldh',Null);
SQL> INSERT INTO EMP VALUES ('Navreet',30,'Clerk',6800,950,25-3-
84,6587,'Pta',Null);
SQL> INSERT INTO EMP VALUES ('Saira',20,'Analyst',6801,3000,15-4-
80,6455,'Chd',Null);
SQL> INSERT INTO EMP VALUES ('Amit',10,'Clerk',6823,1400,25-8-
85,6671,'Ldh',Null);
```

After inserting, values, the table 'EMP' will look like:

EMP

ENAME	**DEPTNO**	**JOB**	**EMPNO**	**SAL**	**HIREDATE**	**MGR**	**CITY**	**COMM**
Nidhi	20	Clerk	6258	900	9-5-83	6801	Chd	
Aastha	30	Salesman	6388	1500	1-12-89	6587	Delhi	300
Sachin	30	Salesman	6410	1350	25-1-92	6587	Pta	500
Rohit	20	Manager	6455	2875	27-12-91	6728	Nba	
Rahul	30	Salesman .	6543	1350	28-5-87	6587	Nba	1400

Aditya	30	Manager	6587	2750	17-8-86	6728	Pta	
Siddharth	10	Manager	6671	2550	29-9-80	6728	Chd	
Kunal	20	Analyst	6677	3000	8-12-82	6455	Delhi	
Akhil	10	President	6728	5000	2-11-85		Delhi,	
Prathiba	30	Salesman	6733	1600	4-6-85	6587	Pta	0
Manmeet	20	Clerk	6765	1050	11-1-84	6677	Ldh	
Navrget	30	Clerk	6800	950	25-3-84	6587	Pta	
Saira	20	Analyst	6801	3000	15-4-80	6455	Chd	
Amit	10	Clerk	6823	1400	25-8-85	6671	Ldh	

NOTE: ***(Mostly queries of this book are based on this table** 'EMP')*

2. **Select:** Once data in inserted into a table, the next step is to view the data contained in the table.

- In order to view the data contained in the table, the select statement is used.
- Select statement is a powerful tool and a most commonly used command.
- It is used to retrieve the data from a table in a database.
- We can also use arithmetic operators in select statement (see example 4, 5 and 6 of select statement).
- With the help of select command, one can retrieve information from one column or more than one column.
- The basic select statement has 6 clauses which are as follows:

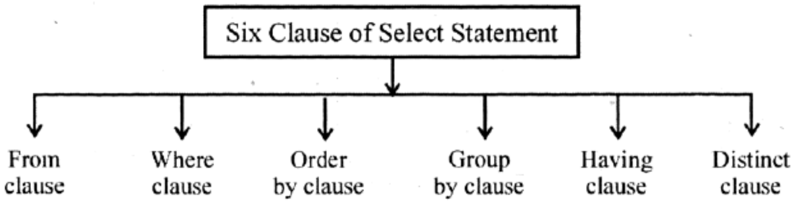


Fig. 10.4: Six Clauses of Select Statement

(a) **Select:** The select clause specifies the table columns that are retrieved. It always use with 'From Clause'.

Syntax of Select Command:

```
SQL> SELECT * FROM table_name;
                                  OR
SQL> SELECT column_list FROM table_name
         [WHERE Clause]
         [GROUP BY Clause]
         [HAVING Clause]
         [ORDER BY Clause];
```

(b) **From:** From clause specifies the table accessed. It is mandatory. It always use with 'Select Command'.

Syntax of From Clause:

```
SQL> SELECT.* FROM table_name;
         OR
SQL> SELECT column_list FROM table_name
         [Where Clause]
         [Group By Clause]
         [Having Clause]
         [Order By Clause];
```

(c) **Where:** Where clause is used when we want to retrieve the specific information from a relation excluding other irrelevant data.

Syntax of Where Clause:

```
SQL> SELECT column_list FROM table_name
         [WHERE Clause];
```

Examples of'-Select Command', 'From Clause' and 'Where Clause':

1. Display all the information of all the employees from relation 'EMP'. Then the query will be:

   SQL> SELECT * FROM EMP;

Result:

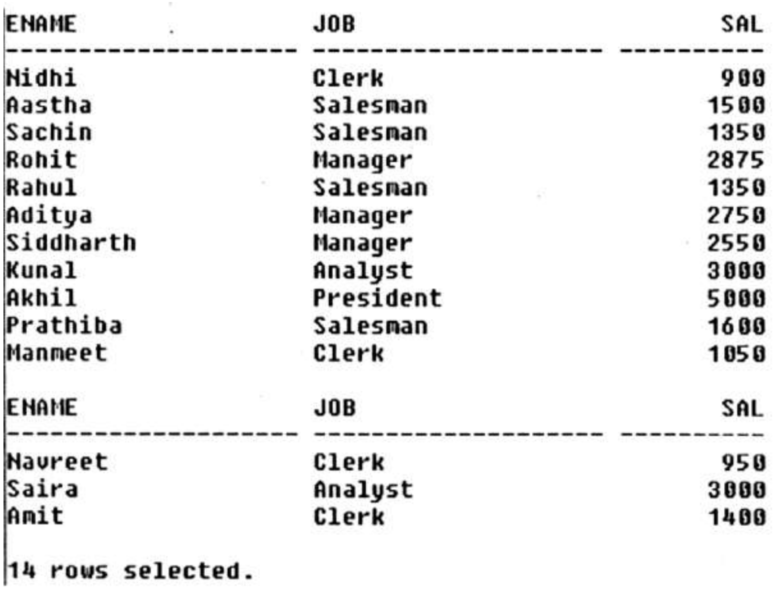
BMP

ENAME	**DEPTNO**	**JOB**	**EMPNO**	**SAL**	**HIREDATE**	**MGR**	**CITY**	**COMM**
Nidhi	20	Clerk	6258	100	9-5-83	6801	Chd	
Aastha	30	Salesman	6388	1500	1-12-89	6587	Delhi	300
Sachin	30	Salesman	6410	1350	25-1-92	6587	Pta	500
Rohit	20	Manager	6455	2875	27-12-91	6728	Nba	
Rahul	30	Salesman	6543	1350	28-5-87	6587	Nba	1400
Aditya	30	Manager	6587	2750	17-8-86	6728	Pta	
Siddharth	10	Manager	6671	2550	29-9-80	6728	Chd	
Kunal	20	Analyst	6677	3000	8-12-82	6455	Delhi	
Akhil	10	President	6728	5000	2-11-85		Delhi	
Prathiba	30	Salesman	6733	1600	4-6-85	6587	Pta	0
Manmeet	20	Clerk	6765	1050	11-1-84	6677	Ldh	
Navreet	30	Clerk	6800	950	25-3-84	6587	Pta	
Saira	20	Analyst	6801	3000	15-4-80	6455	Chd	
Amit	10	Clerk	6823	1400	25-8-85	6671	Ldh	

2. Display only the name, job and salary of all the employees from table "EMP^{55}. Then the query will be;

```
SQL> Select ENAME, JOB, SAL
        From EMP;
```

Result:



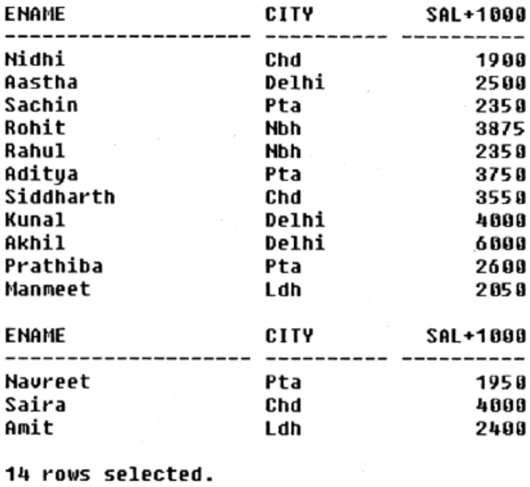
```
ENAME                JOB                         SAL
-------------------- -------------------- ----------
Nidhi                Clerk                       900
Aastha               Salesman                   1500
Sachin               Salesman                   1350
Rohit                Manager                    2875
Rahul                Salesman                   1350
Aditya               Manager                    2750
Siddharth            Manager                    2550
Kunal                Analyst                    3000
Akhil                President                  5000
Prathiba             Salesman                   1600
Manmeet              Clerk                      1050

ENAME                JOB                         SAL
-------------------- -------------------- ----------
Navreet              Clerk                       950
Saira                Analyst                    3000
Amit                 Clerk                      1400

14 rows selected.
```

3. Display name, city and salary of employees from relation 'EMP' where salary of each employee is increased by 1000. Then the query will be:

SQL> SELECT ENAME, CITY, SAL + 1000

FROM EMP;

Result:



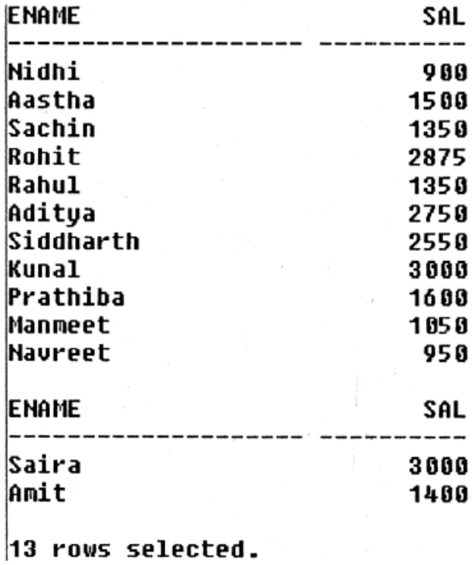
```
ENAME                CITY         SAL+1000
-------------------- ---------- ----------
Nidhi                Chd              1900
Aastha               Delhi            2500
Sachin               Pta              2350
Rohit                Nbh              3875
Rahul                Nbh              2350
Aditya               Pta              3750
Siddharth            Chd              3550
Kunal                Delhi            4000
Akhil                Delhi            6000
Prathiba             Pta              2600
Manmeet              Ldh              2050

ENAME                CITY         SAL+1000
-------------------- ---------- ----------
Navreet              Pta              1950
Saira                Chd              4000
Amit                 Ldh              2400

14 rows selected.
```

4. Display the name and salary of employees whose salary is less than 5000. Then the query will be:

SQL> SELECT ENAME, SAL from EMP

WHERE SAL <5000;

Result:



```
ENAME                       SAL
-------------------- ----------
Nidhi                       900
Aastha                     1500
Sachin                     1350
Rohit                      2875
Rahul                      1350
Aditya                     2750
Siddharth                  2550
Kunal                      3000
Prathiba                   1600
Manmeet                    1050
Navreet                     950

ENAME                       SAL
-------------------- ----------
Saira                      3000
Amit                       1400

13 rows selected.
```

5. Display the names of all the employees belonging to the department number 10 from the relation 'BMP'. Then the query will be:

SQL>SELECT ENAME FROM EMP

WHERE DEPTNO = 10;

Result:

```
ENAME
------------------
Siddharth
Akhil
Amit
```

(d) Order By: The 'Order By Clause' is used with 'Select Statement' to sort the results either in ascending or descending order. By default, it provides results in ascending order. We use column values to sort the table. We can use more than one column to sort the results.

Syntax of Order By Clause:

SQL> SELECT column_list FROM table_name

[ORDER BY Clause];

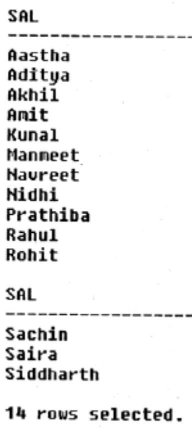
Examples of Order By Clause:

1. Sort the table 'EMP' by the salary of employees. Then the query will be:

SQL>SELECT ENAME SAL FROM EMP

ORDER BY SAL;

Result:



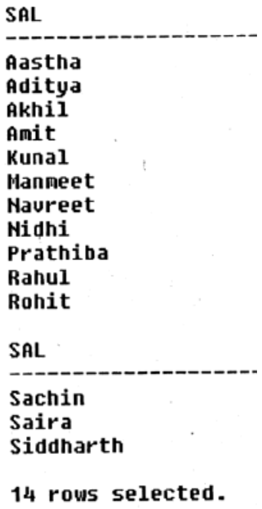
```
SAL
--------------------
Aastha
Aditya
Akhil
Amit
Kunal
Manmeet
Navreet
Nidhi
Prathiba
Rahul
Rohit

SAL
--------------------
Sachin
Saira
Siddharth

14 rows selected.
```

2. Sort the table 'BMP', by the name and salary of employees. Then the query will be:

SQL>SELECT ENAME SAL FROM EMP

ORDER BY ENAME, SAL;

Result:



```
SAL
--------------------
Aastha
Aditya
Akhil
Amit
Kunal
Manmeet
Navreet
Nidhi
Prathiba
Rahul
Rohit

SAL
--------------------
Sachin
Saira
Siddharth

14 rows selected.
```

(e) Group By: It is used to divide the rows into smaller groups. The 'Group By Clause' is used with 'Select Statement' to combine a group of rows based on the values of a particular column or expression. It groups the result after it retrieves the rows from a table. 'Group functions' can be used with 'Having Clause' and cannot be used with 'Where Clause'.

Syntax of Group By Clause:

SQL> SELECT column_list FROM table__name

[GROUP BY Clause];

Example of Group By Clause:

To find the total amount of salary spent on each department from the table 'EMP'. Then the query will be:

```
SQL>SEI.ECT DEPTNO, SUM (SAL) AS TOTAL SALARY FROM EMP
        GROUP BY DEPTNO;
```

Group within Group: 'Group By Clause' can be used to provide results for 'Groups Within Groups'. Suppose we want to know the average amount of salary spent on job type 'Clerk' from department number '20'. We calculate the total amount of salary spent on each department. This is one group. Then we calculate the average amount of salary spent on each type of job from that particular department. This is group within group.

Example of Group within Group Clause:

To find the average monthly salary for each job type within department Then the query will be:

```
SQL>SELECT DEPTNO, JOB, AVG (SAL) AS AVERAGE SALARY FROM EMP
        GROUP BY DEPTNO, JOB;
```

(f) **Having:** It is similar to 'Where Clause', but it is used with group functions. It is used to filter the data. 'Having Clause' can be used with 'Group function' and cannot be used with 'Where Clause'. It restricts the groups that we return on the basis of group functions. It is used to specify which groups are to be displayed.

Syntax of Having Clause:

```
SQL> SELECT column_list FROM table_name
        [HAVING Clause];
```

Example of 'Having Clause:

To find the department who has paid the total salary more than 8.00.6 to its employees. Then the query will be:

```
SQL>SELECT DEPTNO, SUM (SAL) AS TOTAL SALARY FROM EMP GROUP
        BY DEPTNO
        HAVING SUM (SAL)>8000;
```

(g) **Distinct Clause:** The 'Distinct Clause' is used with 'Select Statement' to suppress the duplicate values if any in a column.

Example of 'Distinct Clause':

Display all the different jobs available in the table 'EMP'. Then the query will be:

```
SQL>SELECT DISTINCT JOB FROM EMP;
```

Result:

```
JOB
----------------------------
President
Clerk
Analyst
Salesman
Manager
```

3. Update

- Update command is used when there is a need to modify the data in a table.
- It is used to update existing records in a table.
- It updates single record or multiple records in a table.

Syntax of Update Command:

```
SQL> UPDATE table_name
     SET column1 = value, column2 = value2, ......
     WHERE some_column = some_value;
```

Examples of Update Command:

1. To give everybody a commission of Rs. 100 in the table 'EMP'. Then the query will be:

```
SQL>UPDATE EMP
     SET COMM = 100;
```

2. Update the Manager's salary to 8000 of department number 10 in the table 'EMP'. Then the query will be:

```
SQL>UPDATE EMP
     SET SAL = 8000
     WHERE JOB = 'Manager' AND DEPTNO = 10;
```

4. Delete

- It deletes one or more records from a table and sends it to recycle.
- It doesn't release the memory occupied by the records of the table. Data can be recovered.
- If any subset is defined with condition, then specific records or rows, are deleted, otherwise all records are deleted.
- Executing a delete command may cause triggers to rum which may cause deletion in other tables.
- Example: Sometimes two tables are linked by the foreign key. If we delete rows in one table, then we have to delete those rows from the second table to maintain the referential

integrity.

Syntax of Delete Command:

SQL> DELETE FROM table_name [where condition];

OR

SQL> DELETE from table_name;

Examples of Delete Command:

1. Delete all the records of 'Manager' from the table 'EMP'. Then the query will be:

   SQL>DELETE FROM EMP

   WHERE JOB = 'Manager';

2. Delete all the records from the table 'EMP'. Then the query will be:

   SQL>DELETE FROM EMP;

7.6.3 DCL (Data Control Language)

- It is used to control access to data in a database. It also controls the security of the database.
- To control data in a database, privileges are given to user to access the data without any problem and with proper security.
- It basically provides security to database. Without privileges, no one can access the database.
- A user can access the database according to the privileges given to him.

The following are the various DCL commands:

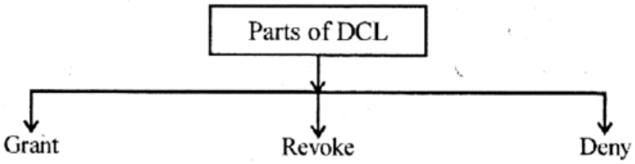


Fig. 10.5: Parts of DCL

(a) **Grant:** It is used to give the permission to the user for restricted access to the database. It allows specified users to perform specified tasks.

(b) **Revoke:** It is used to cancel the previously granted or denied permissions to the users.

(c) **Deny:** It disallows the specified users from performing specified tasks.

7.6.4 TCL (Transaction Control Language)

- TCL is used to manage the changes made by DML (data manipulation language) statements.

- These commands are used for revoking the transactions and to make the data commit to the database.
- Basically, it is used to manage the different transactions occurring within a database.
- Each transaction is completely isolated from other active transactions.
- User can make changes in the particular transaction in database with the transaction control language.
- At the end of the transaction, the database can make all the changes permanent in the database or undoes them all.
- If any problem fails in the middle of a transaction, then the database rolls back the transaction and restore the database into its former state.
- The following are the various TCL commands:

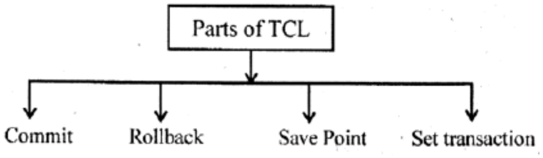


Fig. 10.6: Parts of TCL

(a) Commit

- Commit command is used to save work done. The changes made in the database by the user are not visible to other users until they become permanent in the database.
- Commit command is used to permanent any changes made to the database during the current transaction by the user.
- Commit command is used to save all the changes made to the database since the last commit or rollback command.

Syntax of Commit Command:

```
SQL> COMMIT;
```

Example of Commit Command:

To delete the records of the employees permanently, belonging to the city 'Chd'.

```
SQL>DELETE FROM EMP
WHERE CITY = 'Chd';
SQL>COMMIT;
```

(b) Rollback

- It is used to restore the database to its original state since the last 'commit'.

- It is the inverse of the commit statement.
- It is used to undo the transactions that have not already been saved to the database.
- Oracle provides a facility to-roll back to the last committed state.

Example: We are performing the operations on the database and some problem occurs into the computer system. Yet we have not performed the commit statement, and then rollback command helps to come back to the last committed state.

Syntax of Rollback Command:

SQL> ROLLBACK;

(c) Savepoint

- Savepoint command is used to identify a point in a transaction from which we can later rollback.
- The Savepoint statement defines a Savepoint within a transaction.
- It is a special mark inside a transaction that allows all commands that are executed after it was established to be rolled back, restoring the transaction state to what it was at the time of Savepoint.
- Changes made after a Savepoint can be undone at any time prior to the end of the transaction.
- A transaction can have multiple savepoints.

Syntax of Savepoint Command:

SQL> SAVEPOINT<savepoint name>;

(d) Set Transaction

- Set transaction command has no effect on any subsequent transactions.
- It is used to set the characteristics of the current transaction.
- This command is helpful to determine whether the transaction is read/write or read only.
- If a transaction is read only, then the insert, update, delete and copy commands are disallowed.

7.6.5 Embedded SQL

- Embedded SQL define how SQL statements can be embedded within general purpose programming language like C, C++, Java.
- All the SQL statements DDL, DML and DCL can be grouped into one body. Embedded SQL refers to the use of standard SQL commands embedded within a procedural

programming language.

- When the embedded statements of SQL are execute then all the statements in the body will be executed automatically.

Some of the embedded SQL statements are:

(a) ***Define:*** Define cursor

(b) ***Open:*** Open cursor

(c) ***Execute:*** To execute the command or SQL prompt.

7.6.6 Integrity

- SQL DDL includes commands for integrity constraints so that the data store in the database must satisfy the condition.

7.7 SQL OPERATORS

- SQL supports a wide variety of operators. These operators are extensively used in SQL statements used by the user for the purpose of issuing a query to the database.
- The operators are mainly used in the Where clause, Having clause to filter the data to be selected.
- An operator is a symbol which is used to manipulate the data items (operands).

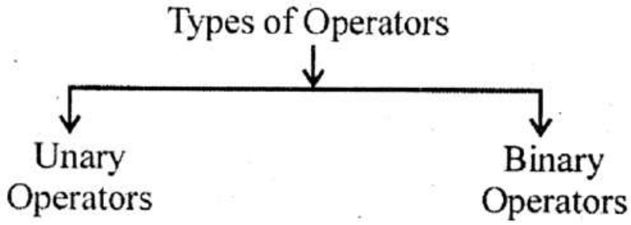


Fig. 10.7: Types of Operators

- Operators are represented by keywords or by special characters.

On the basis-of operands, there are two types-of operators:

Unary Operator: *An* unary, operator operates on only one operand.

Format □ *operator* operand.

Binary Operator: *A* binary operator operates on two operands.

Format □ operand1 *operator* operand 2

The following are the various SQL operators:

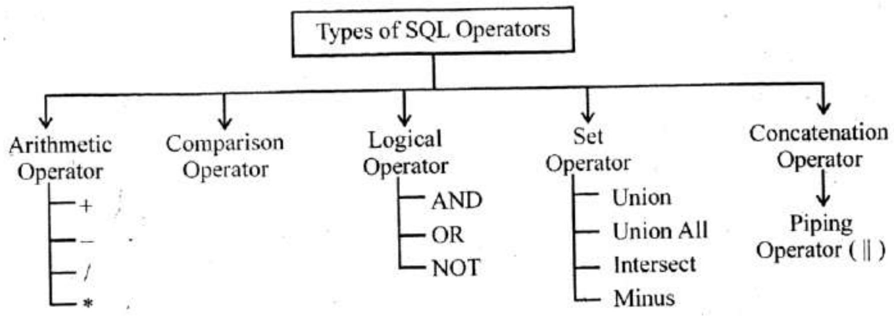


Fig. 10.8: Types of SQL Operators

7.7.1 **Arithmetic Operator**

- An arithmetic operator is used to add, subtract, multiply and divide the numeric values in an expression.
- It is used to perform the mathematical operations on one or more data items or operands of numeric data type.
- It also provides results in numeric values.

Sr. No.	Arithmetic Operator	Description
1	+	Used for addition in SQL
2	-	Used for subtraction in SQL
3	/	Used for division in SQL
4	*	Used for multiplication in SQL

Examples of Arithmetic Operator:

1. Add

Add Rs.500 in the employee's salary whose EMPNO is 6258 from the relation 'EMP'. Then the query will be:

```
SQL> SELECT SAL, SAL+500 FROM EMP
         WHERE EMPNO = 6258;
```

Result:

```
SAL        SAL + 500
------------------------------
900        1400
```

2. Subtract

Subtract the employee's commission from his salary whose EMPNO is 6388. Then the query will be:

```
SQL> SELECT SAL, SAL-COMM FROM EMP
```

```
          WHERE EMPNO = 6388;
```

Result:

```
SAL       SAL-COMM
------------------------------
1500      1200
```

3. Multiply

Multiply the salary of employee by 100 whose EMPNO is 6258 from the relation 'EMP'. Then the query will be:

```
SQL> SELECT SAL, SAL* 100 FROM EMP
          WHERE EMPNO = 6258;
```

Result:

```
SAL       SAL * 100
------------------------------
900       90000
```

7.7.2 Comparison Operator

- A comparison operator is used to compare the column data with specific values with the other column data values.
- It is also used along with the Select Statement to filter data based on specific conditions.

Sr. No.	Comparison Operator	Description
1	=	Equal to
2	!= OR o	Not equal to
3	<	Less than
4	>	Greater than
5	<=	Less than or equal to
6	>=	Greater than or equal to
7	LIKE	Performs pattern matching from columns. The LIKE operator is- used only with Char and match a pattern. % represents sequence of zero or more character.
8	IN	To check a value within a set. It is used to compare a column with more than one value.

9	BETWEEN	To check value within a range. It is used to compare data for a range of value.
10	ANY	To check whether one or more rows in the result set of a sub query meet the specified, condition
11	ALL	To check whether all rows in the result set of a sub query meet the specified condition.
12	EXISTS	To check whether a sub query returns any result.

Example of Equal to (=) Operator:

Display the records of the employees, who live in city 'Chd', from the relation 'EMP'. Then the query will be:

```
SQL> SELECT * FROM EMP
        WHERE CITY = 'Chd';
```

Result:

ENAME	DEPTNO	JOB	EMPNO	SAL	HIREDATE	MGR	CITY
Nidhi	20	Clerk	6258	900	9-5-83	6801	Chd
Siddharth	10	Manager	6671	2550	29-9-80	6728	Chd
Saira	20	Analyst	6801	3000	15-4-80	6455	Chd

Example of Not Equal to (!= OR <>) Operator:

Display the records of the employees, whose city is not equal to 'Chd', from the relation 'EMP'. Then the query will be:

```
SQL> SELECT * FROM EMP
        WHERE CITY! = 'Chd';
```

Result:

ENAME	DEPTNO	JOB	EMPNO	SAL	HIREDATE	MGR	CITY	COMM
Aastha	30	Salesman	6388	1500	1-12-89	6587	Delhi	300
Sacliin	30	Salesman	6410	1350	25-1-92	6587	Pta	500

Rohit	20	Manager	l3455	2875	27-12-91	6728	Nba	
Rahul	50	Salesman	6543	1350	28-5-87	6587	Nba_	1400
Aditya	30	Manager	6587	2750	17-8-86	6728	Pta	
Kunal	20	Analyst	6677	3000	842-82	6455	Delhi	
Akhil	10	President	6728	5000	2-11-85		Delhi	
Prathiba	30	Salesman	6733	1600	4-6-85	6587	Pta	0
Manmeet	20	Clerk	6765	1050	114-84	6677	Ldh	
Navreet	30	Clerk	6800	950	25-3-84	6587	Pta	
Amit	10	Clerk	6823 .	1400	25-8-85	6671	Ldh	

11 rows selected

Example of Less than (<) Operator:

Display the name of the employees, whose salary is less than '1400', from the table 'EMP'. Then the query will be:

```
SQL> SELECT ENAME FROM EMP
        WHERE SAL = 1400'
```

Result:

```
ENAME
---------------
Nidhi
Sachin
Rahul
Nanmeet
Naureet
```

Example of Greater than (>) Operator:

Display the name of the employees, whose salary is greater than '1400', from the table 'EMP'. Then the query will be:

```
SQL> SELECT ENAME FROM EMP
        WHERE SAI>1400;
```

Result:

```
ENAME
-------------
Nidhi
Sachin
Rahul
Manmeet
Navreet
Amit
6 rows selected.
```

Example of Less than or equal to (<=) Operator:

Display the name of the employees, whose salary is less than or equal to '1400', from the table 'EMP'. Then the query will be:

```
SQL> SELECT ENAME FROM EMP
     WHERE SAL< =1400;
```

Result:

```
ENAME
---------------
Nidhi
Sachin
Rahul
Manmeet
Navreet
Amit
6 rows selected.
```

Example of Greater than (>=) Operator:

Display the name of the employees, whose salary is greater than or equal to '1400', the table 'EMP'. Then the query will be:

```
SQL> SELECT ENAME FROM EMP
     WHERE SAL< =1400;
```

Result:

```
ENAME
```

```
---------------
Aastha
Rohit
Aditya
Siddharth
Kunal
Akhil
Prathiba
Saira
Amit
9 rows selected.
```

Examples of LIKE Operator:

1. Display the employees whose name start with 'S' from the table 'EMP'. Then the query will be:

```
SQL> SELECT ENAME FROM EMP
     WHERE ENAME LIKE 'S%';
```

Result:

```
ENAME
---------------
Sachin
Siddharth
Saira
```

2. Display the employees, whose name ends with 'S', from the table 'EMP'. Then the query will be:

```
SQL> SELECT ENAME FROM EMP
     WHERE ENAME LIKE '%S';
```

Result:

NO ROW SELECTED.

- Display the employees, where 'S' is in the middle of the name, from the Table 'EMP'. Then the query will be:

```
SQL> SELECT ENAME FROM EMP
```

WHERE ENAME LIKE '%S%';

Result:

```
ENAME
--------
Aastha
```

Example of IN Operator:

Display the names of the employees, who are analyst and clerk, from the table 'EMP'. Then the query will be:

SQL>SELECT ENAME FROM EMP

WHERE JOB IN ('Analyst', 'Clerk');

Result:

```
ENAME
Nidhi
Kunal
Manmeet
Navreet
Saira
Amit
6 rows selected.
```

Example of BETWEEN Operator:

Display the name and salary of all employees, whose salary is between 2000 and 3000, from the table 'EMP'. Then the query will be:

SQL>SELECT ENAME, SAL FROM EMP

WHERE SAL BETWEEN 2000 AND 3000;

Result:

```
ENAME                SAL
----------------------
Rohit                2875
Aditya               2750
Siddharth            2550
Kunal                3000
```

```
            Saira                   3000
```

7.7.3 Logical Operator

- Logical operators compare two or more than two conditions at a time to determine whether a row can be selected for the output.
- When retrieving data using a Select Statement, we use logical operators in the Where Clause which allows us to combine more than one condition.

Sr. No.	Logical Operator	Description
1	AND	For the row to be selected all the specified conditions must be true.
2	OR	For the row to be selected at least one of the specified conditions must be true.
3	NOT	For the row to be selected, the specified conditions must be false.

- NOT is totally opposite of AND and OR operator. When we want to find those rows that do not satisfy a condition, then we use the NOT operator.

1. **Examples of AND Operator:**

- To find the names of the clerks from the table "EMP" who are working in the department number 20, then the query will be:

```
SQL> SELECT ENAME FROM EMP
     WHERE NOB = 'CLERK' AND DEPTNO = 20;
```

Result:

```
ENAHE
----------------
Nidhi
Manmeet
```

- To find the Ename, Sal, Job from the table "EMP" where salary is greater than 1500 and deptno is 30, then the query will be:

```
SQL> SELECT ENAME, SAL, JOB FROM EMP
     WHERE SAL>1500 AND DEPTNO = 30;
```

Result:

```
ENAME                    SAL JOB
----------------------------------------
Rohit                    2175/Manager
Aditya                   2758 Manager
Prathiba                 1600 Salesman
```

- To find all the information of the employee's from the table "EMP" whose job is manager and deptno is 10, then the query will be:

```
SQL> SELECT * FROM EMP
  WHERE JOB = 'Manager' AND DEPTNO = 10;
```

Result:

ENAME	DEPTON	JOB	EMPNO	SAL	HIREDATE	MGR	CITY
Siddharth	10	Manager	6671	2550	29-9-80	6728	Chd

2. **Examples of OR Operator:**

- To find the names of the employees from the table "EMP", who are analysts and clerk, then the query will be:

```
SQL> SELECT ENAME FROM EMP
  WHERE JOB = -'Analyst' OR JOB = 'CIerk';
```

Result:

```
ENAME
---------------
Nidhi
Kunal
Navreet
Saira
Amit
6 rows selected.
```

- Display the Ename, Empno from the table "EMP", whose job is clerk or deptno is 10, then the query will be:

```
SQL> SELECT ENAME, EMPNO FROM EMP
     WHERE JOB = 'Clerk' .OR DEPTNO = 10;
```

Result:

```
ENAME                    EMPNO
-----------------------------
Nidhi                    6258
Siddharth                6671
Akhil                    6728
Manmeet                  6765
Navreet                  6888
Amit                     6823
6 rows selected.
```

3. NOT

- Display the names of the employees from the table "EMP", who are not clerks, then the query will be:

```
SQL> SELECT ENAME FROM EMP
     WHERE JOB <> 'Clerk';
               OR
     SQL> SELECT ENAME FROM EMP
          WHERE JOB! = 'Clerk';
```

Result:

```
ENAME
--------------
Aastha
Sachin
Rohit
Rahul
Aditya
Siddharth
Kunal
```

```
                  Akhil
                  Prathiba
                  Saira
                  10 rows selected.
```

- Display the name and deptno of employees from the table "EMP", who are not belonging to deptno 10 or 20, then the query will be:

SQL> SELECT ENAME, DEPTNO FROM EMP

WHERE NOT (DEPTNO = 10 OR DEPTNO = 20);

Result:

```
ENAME                    DEPTNO
-----------------------------
Aastha                   30
Sachin                   30
Rohit                    30
Rahul                    30
Aditya                   30
Prathiba                 30
Navreet                  30
7 rows selected.
```

7.7.4 Set Operator

- Set operators are used to combine the results from two or more Select statements.
- The result of each Select Statement can be treated as a SET. Set operators are applied on these SETS to achieve the final result.
- Set operators follow some rules which are as follows:
- Number of columns should be in exact same order in all the queries.
- Number of columns should be same in all the queries.
- Data types of retrieved columns (selected statements) should be matched.

UNION ALL

SELECT Column List FROM Table2;

Example of Union All Operator:

Display all the jobs in department 10 and 20 from the table 'EMP'. Then the query will be:

```
SQL> SELECT JOB FROM EMP
          WHERE DEPTNO = 10
          UNION ALL
          SELECT JOB FROM EMP
          WHERE DEPTNO = 20;
```

Result:

```
JOB
---------------
Manager
President
Clerk
Clerk
Analyst
Clerk
Analyst
7 rows selected.
```

NOTE: ***Union operator provides results with automatically removal of duplicate values whereas Union All operator provides results without removal of any duplicate value.***

3. Intersect

Intersect operator combine the two table expressions into one and return a result set which consists of rows that appear in the results of both table expressions. It also removes all the duplicate rows from the result set.

Syntax of Intersect Operator:-

```
SQL> SELECT Column List FROM Table 1
          INTERSECT
          SELECT Column List FROM Table2;
```

Example of Intersect Operator:

Display all the jobs common in department 10 and 20 from the table 'EMP'. Then the query will be:

```
SQL> SELECT JOB FROM EMP
```

```
WHERE DEPTNO = 10
INTERSECT
SELECT JOB FROM EMP
WHERE DEPTNO = 20;
```

Syntax: ->

SQL><SELECT STATEMENT><SET OPERATOR>< SELECT STATEMENT > <ORDER BY Clause>;

Sr. No.	Set Operator	Description
1	Union	Returns all distinct rows selected by either query, excluding all duplicate rows.
2	Union All	Returns all rows selected by either query, including all duplicate rows.
3	Intersect	Returns all distinct rows selected by both queries.
4	Minus	Returns all distinct rows selected by the first query but not the second.

1. Union

It combines the results of two queries (same number of columns and compatible data types) into a single table of all matching rows. Union automatically removes all the duplicate values.

Syntax of Union Operator:

```
SQL> SELECT Column List FROM Table1
     UNION
     SELECT Column List FROM Table2;
```

Example of Union Operator:

- Display the different jobs in department 10 and 20 from the table 'EMP'. Then the query will be:

```
SQL> SELECT JOB FROM EMP
  WHERE DEPTNO = 10
  UNION
  SELECT JOB FROM EMP
  WHERE DEPTNO = 20;
```

Result:

```
JOB
```

```
------------
Analyst
Clerk
Manager
President
```

2. Union All

It combines the results of two queries (same number of columns and compatible data types) into a single table of all matching rows. It includes (shows) all the duplicate values.

Syntax of Union All Operator:

```
SQL> SELECT Column List FROM Table1
```

Result:

```
JOB
-----------
Clerk
```

4. Minus

It compares each record in statement1 with a record in statement2. It returns the results with the records in statement1 that are not in statement2.

Rows retrieved by the second query are subtracted from the rows retrieved by the first query. Only those records are considered as a result which are present only in statement1 and not in statement2.

Syntax of Minus Operator:-

```
SQL> SELECT Column List FROM Table1
     MINUS
     SELECT Column List FROM Table2;
```

Example of Minus Operator:

Display all the unique jobs in the department 10 from the table 'EMP'. Then the query will be:

```
SQL> SELECT JOB FROM EMP
     WHERE DEPTNO = 10
     MINUS
     SELECT JOB FROM EMP
```

```
WHERE DEPTNO = 20
MINUS
SELECT JOB FROM EMP
WHERE DEPTNO = 30;
```

Result:

```
JOB
-----------------
President
```

7.7.5 Concatenation Operator

- Concatenation operator is used to combine the two or more data strings.
- The operands of the concatenation must be compatible strings.
- Character string cannot be concatenated with a binary string.
- Concat and vertical bars (..) both represent the concatenation operator.

Concatenation Operator	Description
Piping Operator (...)	It is used to combine two or more strings

Examples of Concatenation Operator:

- List the employee salary whose empno is 6728. Then the query will he:

```
SQL> Select 'My Salary is =' .... Sal as Salary
From EMP Where Empno = 6728.
```

Result: **My Salary is 5000.**

- List the employee name whose empno is 6728. Then the query will be:

```
SQL> Select 'My Name is =' ... Ename as Name
From EMP Where Empno = 6728.
```

Result: **My Name is Akhil.**

7.8 SQL FUNCTIONS

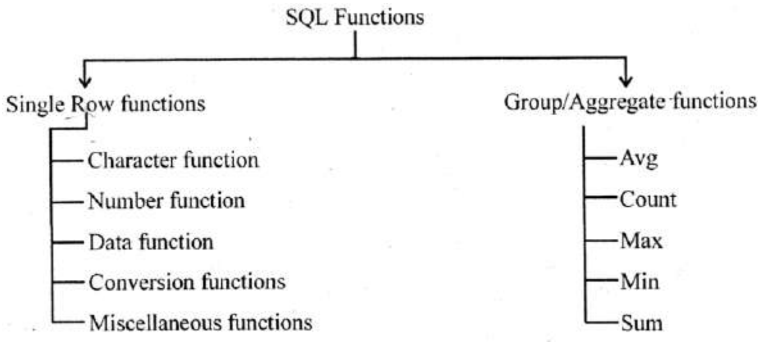


Fig. 10.9: SQL Functions

7.8.1 Single Row Functions

Single row functions operate on single rows only and returns one result per row. ***The types of single row functions are as follows:***

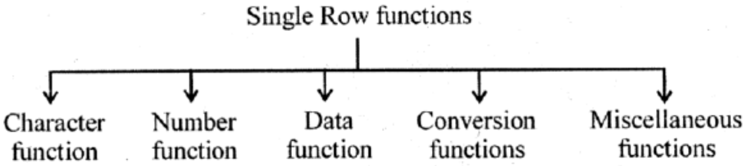


Fig. 10.10: Single Row Functions

1. Character Functions

- It is also known as text functions.
- It is used to manipulate text strings.
- It accepts character input only and returns either character or numeric values.

The following are the types of character functions:

(a) LOWER **(string):** It converts uppercase or mixed case character strings into lowercase character strings.

Example: SQL>SELECT LOWER (JOB) FROM EMP;

Result:

```
LOWER(JOB)
----------------
clerk
salesman
salesman
manager
salesman
manager
manager
analyst
president
salesman
clerk

LOWER(JOB)
---------------
clerk
analyst
clerk
14 rows selected.
```

(b) **UPPER (string):** It converts lowercase or mixed case character strings into uppercase character strings.

Example: SQL>SELECT UPPER (JOB) FROM EMP;

Result:

```
UPPER(JOB)
```

```
--------------
CLERK
SALESMAN
SALESMAN
MANAGER
SALESMAN
MANAGER
MANAGER
ANALVST
PRESIDENT
SALESMAN
CLERK

UPPER(JOB)
-------------
CLERK
ANALVST
CLERK
14 rows selected.
```

(c) **CONCAT (string1, string2):** It is equivalent to the concatenation operator. It returns string1 concatenated with string2. It joins (combines) two string values together.

Example: SQL>SELECT CONCAT ('MONIKA', 'TATHAK') FROM DUAL;

Result: MONIKA PATHAK

(d) **LENGTH (string):** It is used to get the length of a string as a numeric value. **Example: SQL>SELECT LENGTH (Akhil) FROM DUAL;**

Result: 5

(e) **ASCII (string):** It is used to return the decimal representation of the first byte of string in the database character set.

Example: SQL> ASCII (Amit) FROM DUAL;

Result: 65

2. **Number Functions**

- It is used to perform operations on numbers.
- It accepts numeric input, only and returns numeric values.

The following are the types of numeric functions:

(a) **ABS (n):** It returns absolute value of numeric value.

Example: SQL>SELECT ABS (-29) FROM DUAL;

Result: 29

(b) **CEIL (n):** It returns the next smallest integer greater than or equal to parameter passed to n.

Example: SQL>SELECT CEIL (29.8) FROM DUAL;

Result: 30

(c) **FLOOR (n):** It returns the largest integer value less than or equal to parameter passed to n.

Example: SQL>SELECT FLOOR (29.8) FROM DUAL;

Result: 29

(d) **MOD (m,n):** It returns the remainder of m divided by n. It returns m if n is 0.

Example: SQL>SELECT MOD (16,3) FROM DUAL;

Result: 1

(e) **SQRT (n):** It returns the square root of n. The value of n cannot be negative.

Example: SQL>SELECT SQRT (25) ;FROM DUAL;

Result: 5

3. Date Functions

- Date functions operate on values of the Date datatype.
- It takes values of Date datatype as input and return values of Date datatype as output, except the Months_Between function, which returns a number.

The following are the types of date functions:

(a) **SYSDATE:** It returns the current system date and time on our local database.

Example: SQL>SELECT SYSDATE FROM DUAL;

Result:

```
SYSDATE
---------
18-JUN-15
```

(b) **LAST_DAY:** It returns the date of the last day of the month specified.

Example: SQL>SELECT SYSDATE LAST DAY (SYSDATE) FROM DUAL;

(c) **CURRENT_DATE:** It returns the current date in the Gregorian calendar for the session's time zone.

Example: SQL>SELECT SYSDATE CURRENT DAY (SYSDATE) FROM DUAL;

(d) **NEXT_DAY:** It returns the date of next specified day of the week after the 'date'.

Example: SQL>SELECT SYSDATE NEXT DAY (SYSDATE) FROM DUAL;

(e) **ADD_MONTHS:** It adds or subtracts the months to or from a date.

Example: SQL>SELECT SYSDATE, ADD_MONTHS (SYSDATE, 4) FROM DUAL;

Result:

```
SYSDATE        ADD_MONTH
----------------------
18-JUN-15      18-OCT-15
```

4. **Conversion Functions**

It converts the value from one form to another form.

The following are the types of conversion functions:

(a) **Implicit Data Type Conversion:** It occurs when the expression evaluator automatically converts the data from one data type to another.

(b) **Explicit Data Type Conversion:** It occurs when we explicitly converts the data from one data type to another.

5. **Miscellaneous Functions**

The following are the types of miscellaneous functions:

(a) **GREATEST:** It returns the greatest value in the list of expressions.

Example: SQL>SELECT GREATEST (2, 11, 25, 29) FROM DUAL;

Result: 29

(b) **LEAST:** It returns the smallest value in the list of expressions.

Example: SQL>SELECT LEAST (2, 11, 25, 29) FROM DUAL;

Result: 2

(c) **USER:** It returns the username of the current user logged on.

Example: SQL>SELECT USER FROM DUAL;

Result: SCOTT

7.8.2 Group/Aggregate Functions

- Aggregate functions are also known as Group functions or Summary functions.
- SQL supports the functions which can be used to select and compute the numeric, date columns and characters of the relation.
- These functions operate on multiple rows (group of rows) and return only one value for a group or table, therefore these functions are known as aggregate functions. By default, all rows are treated as one group in a table.

The types of aggregate functions are as follows:

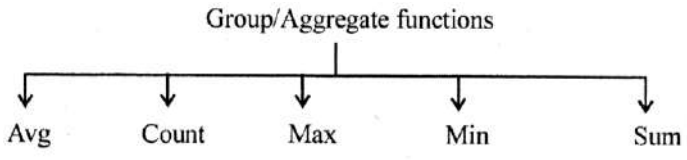


Fig. 10.11: Group/Aggregate Functions

STUD

Name	Class	Roll Number	Marks	Age
Akhil	C12	11	95	16
Monika	C12-	12	91	15
Aastha	M12	13	95	14
Rohit	E12	14	94	12
Rahul	E12	15	93	13
Ankush	C12	16	95	15
Radhika	M12	17	92	14

1. **Avg:** The Avg (average) function returns the arithmetic mean of the value of a column in a given relation. This function is applicable on numeric values.

Examples of Avg Function:->

- To find the average marks of the students from the table STUD, then the query will be:

  SQL> SELECT AVG (Marks) FROM STUD;

Result: 93.51

- To find the average salary of the employees from the table EMR Then the query will be:

  SQL> Select AVG (SAL) AS Average Salary FROM EMP;

Result:

```
Average Salary
------------------
2091.07143
```

2. **Count:** The Count function returns the number of rows in a relation (table). This function is used for numeric, character values and date. The Count function returns value only if it satisfies the condition stated in the Where Clause.

Examples of Count Function:

- To find the number of students from the table 'STUD'. Then the query will be:

  SQL> Select COUNT (*) FROM STUD;

Result: 7

- To find the total number of employees from the table EMP, Then the query will be:

  SQL> SELECT $^{\%}$COUNT (*) AS TOTAL EMPLOYEE FROM EMP;

Result:

```
TOTAL EMPLOYEE
----------------------------
          14
```

3. **Max:** The Max function returns the maximum of the values of a column from the given relation.

Examples of Max Function:

- To find the maximum marks from the table 'STUD'. Then the query will be:

  SQL> MAX (Marks) FROM STUD;

Result: 95

- To find the maximum salary drawn by the employee from the table EMP. Then the query will be:

  SQL> MAX (SAL) AS Maximum Salary FROM EMP;

Result:

```
Maximum Salary
------------------------
5000
```

4. **Min:** The Min function returns the minimum of the values of a column from the given relation.

Examples of Min Function:

- To find the minimum marks from the table STUD. Then the query will be:

SQL> MIN (Marks) FROM STUD;

Result: 91

- To find the minimum salary drawn by the employee from the table EMP. Then the query will be:

  SQL> MIN (SAL) AS Minimum Salary FROM EMP;

Result: **Minimum Salary**

```
---------------------
900
```

5. **Sum:** The Sum function returns the sum of values (numeric type) of a column.

Example of Sum Function:

- To find the sum of marks from the table STUD. Then the query will be:

  SQL> SELECT SUM (Marks) FROM STUD;

Result: **655**

- To find the total salary given to the employees from the table BMP. Then the query will be:

  SQL> SELECT SUM (SAL) AS Total Salary FROM EMP;

Result: Total Salary

```
----------------
29275
```

7.9 JOINS

- Mostly we retrieve data from one table at a time. But what will we do if we need to retrieve data from multiple tables.
- Oracle provides the facility to retrieve the data from multiple tables with the help of joins.
- Joins are used to combine columns from different tables.
- Joins allow us to retrieve the data from multiple users in a single query.
- Joins permits us to select data from more than one table in one SQL statement (query).
- A join is used to combine rows from multiple tables.
- Joins are used to relate information in different tables.
- The connection between tables is established through the Where Clause.
- Where Clause is known as join condition.
- The rows retrieved after joining the two tables based on a condition in which one table act as a primary key and other act as a foreign key. Columns in both tables should be matched.

Syntax of Join:

SQL> SELECT tablel.column, table2.column, ...tableN.column

FROM table1, table2, ...tableN.

WHERE tablel.column1 = table2. column2;

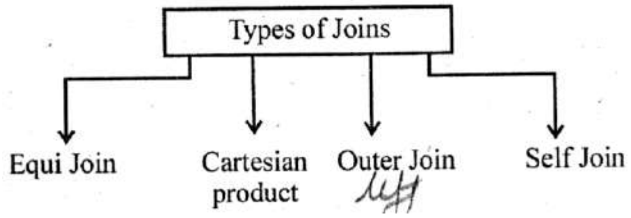


Fig. 10.12: Types of Join

7.9.2 Equi Join

- It is also known as Inner Join.
- When two tables are joined together using equality of values in one or more columns, they make an equi join.
- Equi join is used when we need to compare each record in two joined tables and comes with matching record.
- Table prefixes are utilized to prevent ambiguity.
- We use equi join (inner join) when we only want to return records where there is at least one row in both tables that match the join condition.
- Equi join uses the equal sign as the comparison operator.

Example of Equi Join:

First Table is BMP

Second Table is DEPT.

EMP

ENAME	DEPTNO	JOB	EMPNO	SAL	HIREDATE	MGR	CITY	COMM
Nidhi	20	Clerk	6258	900	9-5-83	6801	Chd	
Aastha	30	Salesman	6388	1500	1-12-89	6587	Delhi	300
Sachin	30	Salesman	6410	1350	25-1-92	6587	Pta	500
Rohit	20	Manager	6455	2875	27-12-91	6728	Nba	
Rahul	30	Salesman	6543	1350	28-5-87	6587	Nba	1400
Aditya	30	Manager	6587	2750	17-8-86	672S	Pta	

Siddharth	10	Manager	6671	2550	29-9-80	6728	Chd	
Kunal	20	Analyst	6677	3000	8-12-82	6455	Delhi	
Akhil	10	President	6728	5000	2-11-85		Delhi	
Prathiba	30	Salesman	6733	1600	4-6-85	6587	Pta	0
Manmeet	20	Clerk	6765	1050	11-1-84	6677	Ldh	
Navreet	30	Clerk	6800	950	25-3-84	6587	Pta	
Saira	20	Analyst	6801	3000	15-4-80	6455	Chd	
Amit	10	Clerk	6823	1400	25-8-85	6671	Ldh	

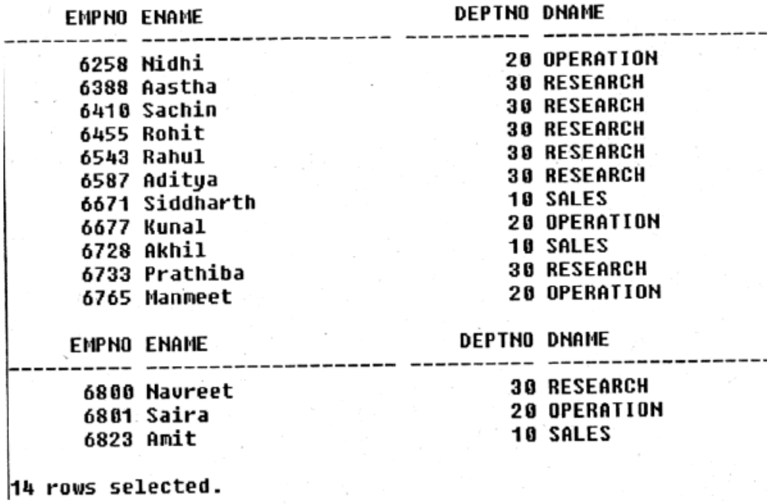
DEPT

DEPTNO	DNAME	LOG
10	Sales	London
20	Operation	Mumbai
30	Research	Paris
40	Accounting	New York

Then the query will be:

SQL> SELECT EMPNO, ENAME, EMP.DEPTNO, DNAME FROM EMP, DEPT WHERE EMP.DEPTNO = DEPT.DEPTNO;

Result:



```
     EMPNO ENAME                              DEPTNO DNAME
---------- -------------------- ---------- ------------------
      6258 Nidhi                                  20 OPERATION
      6388 Aastha                                 30 RESEARCH
      6410 Sachin                                 30 RESEARCH
      6455 Rohit                                  30 RESEARCH
      6543 Rahul                                  30 RESEARCH
      6587 Aditya                                 30 RESEARCH
      6671 Siddharth                              10 SALES
      6677 Kunal                                  20 OPERATION
      6728 Akhil                                  10 SALES
      6733 Prathiba                               30 RESEARCH
      6765 Manmeet                                20 OPERATION

     EMPNO ENAME                              DEPTNO DNAME
---------- -------------------- ---------- ------------------
      6800 Navreet                                30 RESEARCH
      6801 Saira                                  20 OPERATION
      6823 Amit                                   10 SALES

14 rows selected.
```

Explanation of Equi Join:

For Equi Join, both the table names should be mentioned.

Column name should be specified with the table name to avoid confusion.

Deptno of BMP table is joined with the deptno of DEPT table because Deptno exists in both the tables.

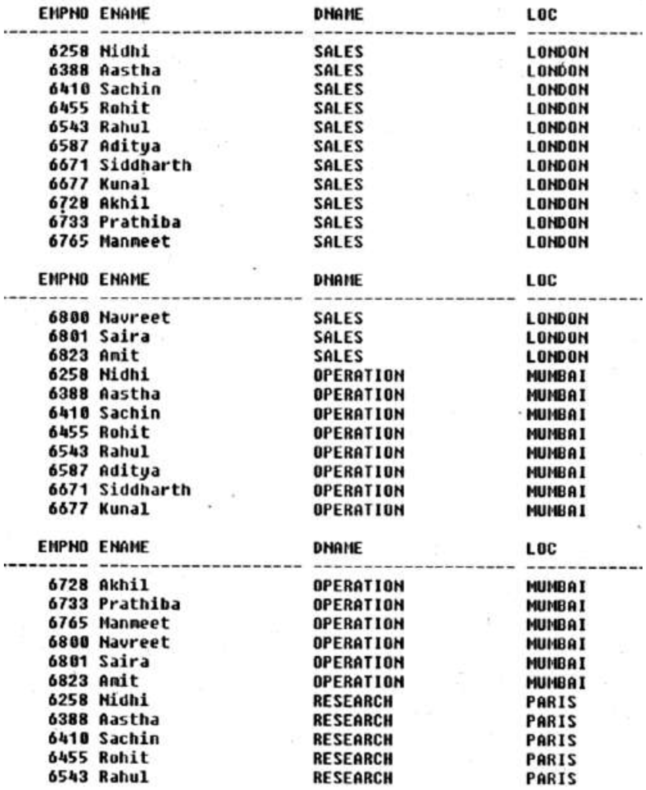
7.9.2 Cross Join

- It is also known as cartesian product or cartesian join.
- It returns the number of rows equal to the product of all rows in all rows in all the tables being joined.
- It provides results in mXn rows.
- It is used when we want to join every row of a table to every row of itself.

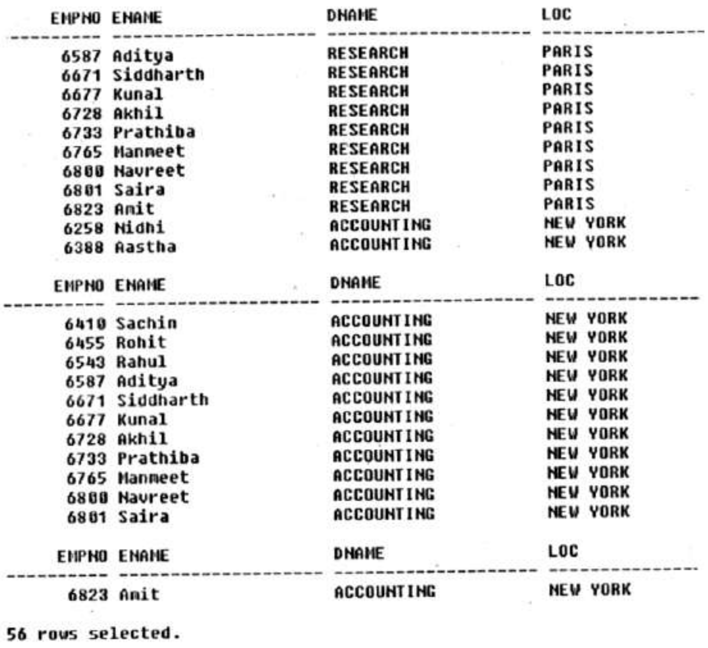
Example of Cross Join:

SQL>SELECT EMPNO, ENAME, DNAME, LOC FROM EMP, DEPT;

Result:



```
   EMPNO ENAME                 DNAME                 LOC
--------- -------------------- -------------------- ------------
     6258 Nidhi                SALES                 LONDON
     6388 Aastha               SALES                 LONDON
     6410 Sachin               SALES                 LONDON
     6455 Rohit                SALES                 LONDON
     6543 Rahul                SALES                 LONDON
     6587 Aditya               SALES                 LONDON
     6671 Siddharth            SALES                 LONDON
     6677 Kunal                SALES                 LONDON
     6728 Akhil                SALES                 LONDON
     6733 Prathiba             SALES                 LONDON
     6765 Manmeet              SALES                 LONDON

   EMPNO ENAME                 DNAME                 LOC
--------- -------------------- -------------------- ------------
     6800 Navreet              SALES                 LONDON
     6801 Saira                SALES                 LONDON
     6823 Amit                 SALES                 LONDON
     6258 Nidhi                OPERATION             MUMBAI
     6388 Aastha               OPERATION             MUMBAI
     6410 Sachin               OPERATION             MUMBAI
     6455 Rohit                OPERATION             MUMBAI
     6543 Rahul                OPERATION             MUMBAI
     6587 Aditya               OPERATION             MUMBAI
     6671 Siddharth            OPERATION             MUMBAI
     6677 Kunal                OPERATION             MUMBAI

   EMPNO ENAME                 DNAME                 LOC
--------- -------------------- -------------------- ------------
     6728 Akhil                OPERATION             MUMBAI
     6733 Prathiba             OPERATION             MUMBAI
     6765 Manmeet              OPERATION             MUMBAI
     6800 Navreet              OPERATION             MUMBAI
     6801 Saira                OPERATION             MUMBAI
     6823 Amit                 OPERATION             MUMBAI
     6258 Nidhi                RESEARCH              PARIS
     6388 Aastha               RESEARCH              PARIS
     6410 Sachin               RESEARCH              PARIS
     6455 Rohit                RESEARCH              PARIS
     6543 Rahul                RESEARCH              PARIS
```



```
     EMPNO ENAME                DNAME                LOC
---------- -------------------- -------------------- ---------------
      6587 Aditya               RESEARCH             PARIS
      6671 Siddharth            RESEARCH             PARIS
      6677 Kunal                RESEARCH             PARIS
      6728 Akhil                RESEARCH             PARIS
      6733 Prathiba             RESEARCH             PARIS
      6765 Manmeet              RESEARCH             PARIS
      6800 Navreet              RESEARCH             PARIS
      6801 Saira                RESEARCH             PARIS
      6823 Amit                 RESEARCH             PARIS
      6258 Nidhi                ACCOUNTING           NEW YORK
      6388 Aastha               ACCOUNTING           NEW YORK

     EMPNO ENAME                DNAME                LOC
---------- -------------------- -------------------- ---------------
      6410 Sachin               ACCOUNTING           NEW YORK
      6455 Rohit                ACCOUNTING           NEW YORK
      6543 Rahul                ACCOUNTING           NEW YORK
      6587 Aditya               ACCOUNTING           NEW YORK
      6671 Siddharth            ACCOUNTING           NEW YORK
      6677 Kunal                ACCOUNTING           NEW YORK
      6728 Akhil                ACCOUNTING           NEW YORK
      6733 Prathiba             ACCOUNTING           NEW YORK
      6765 Manmeet              ACCOUNTING           NEW YORK
      6800 Navreet              ACCOUNTING           NEW YORK
      6801 Saira                ACCOUNTING           NEW YORK

     EMPNO ENAME                DNAME                LOC
---------- -------------------- -------------------- ---------------
      6823 Amit                 ACCOUNTING           NEW YORK

56 rows selected.
```

Explanation:

Table BMP has 14 rows.

Table DEPT has 4 rows.

Then, total number of rows = mXn

=14X4

=> Total number of rows =56 rows

7.9.3 Outer Join

- Outer join has symbol (+).
- It is used if there is any value in one table that do not have corresponding value in other table. Such rows are forcefully selected by it.
- It is used on one side of the join condition only and the corresponding columns for that row will have NULL value.

Example of Outer Join:

SQL>SELECT EMPNO, ENAME, E-MP.DEPTNO, DNAME, LOC FROM EMP,DEPT

WHERE EMP.DEPTNQ (+) = DEPT.DEPTNO;

Result:

EMPNO	ENAME	DEPTNO	DNAME	LOC
6258	Nidhi	20	Operation	Mumbai
6388	Aastha	30	Research	Paris
6410	Sachin	30	Research	Paris
6455	Rohit	30	Research	Paris
6543	Rahul	30	Research	Paris
6587	Aditya	30	Research	Paris
6671	Siddharth	10	Sale	Paris
6677	Kunal	20	Research	Paris
6728	Akhil	10	Sale	London
6733	Prathiba	30	Research	Mumbai
6765	Manmeet	20	Operation	London
6800	Navreet	30	Research	Paris
6801	Saira	20	Operation	Mumbai
6823	Amit	10	Sale	London

7 .9.4 Self Join

- Self join is used when a table is joined/compared to itself.
- A table is joined to itself means each row of the table is combined with itself and with every row of the table.
- If we want to use self join, then we need to open the two copies of same table by using table aliases
- Table name aliases are defined in the From Clause of the query.
- Table alias is used to avoid confusion among two same tables.

Example of Self Join:

SQL>SELECT WORKER.ENAME AS ENAME, MANAGER.ENAME AS MANAGER

FROM EMP WORKER, EMP MANGER

WHERE WORKER.MGR = MANAGER.EMPNO;

7.10 ROLL UP OPERATION

- It is just an extension of the Group of Clause.
- It appears only with Group by Clause.
- It is useful in generating reports (result set) that contain subtotals and totals.
- It is used by the report writers to extract statistics and summary information from result sets.
- It calculates a grand total. First, it calculates the standard aggregate values specified in the Group by Clause. Then, it creates progressively higher level subtotals, moving from right to left through the list of grouping columns. Finally, it creates a grand total.
- It is much like the Group by Clause except it gives subtotal and grand totals at the end.

Syntax of Roll up Operation:

SQL> SELECT............

FROM.............

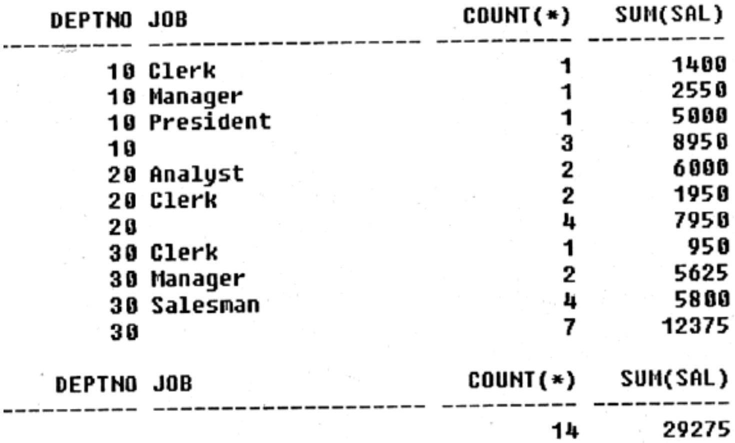
WHERE............

GROUP BY ROLLUP [column1, column2, ...];

Example of Roll up Operation:

SQL>SELECT DEFTNO, JOB, COUNT (*), SUM (SAL) FROM EMP GROUP BY ROLLUP (DEPTNO/JOB);

Result:



```
    DEPTNO JOB                    COUNT(*)   SUM(SAL)
---------- -------------------- ---------- ----------
        10 Clerk                         1       1400
        10 Manager                       1       2550
        10 President                     1       5000
        10                               3       8950
        20 Analyst                       2       6000
        20 Clerk                         2       1950
        20                               4       7950
        30 Clerk                         1        950
        30 Manager                       2       5625
        30 Salesman                      4       5800
        30                               7      12375

    DEPTNO JOB                    COUNT(*)   SUM(SAL)
---------- -------------------- ---------- ----------
                                        14      29275
```

```
12 rows selected.
```

7.11 CUBE OPERATION

- It is a simple extension of Group by Clause with select statement.
- It takes aggregation one step further than Roll Up.
- The cube operation generate result set (subtotals) contains a cross tabulation of all the possible combinations of the grouping columns.
- Therefore, the result of a cube operation will contain all subtotals produced by an equivalent Roll Up operation and some additional subtotals.
- It applied to all aggregate functions. It produces subtotals and a grand total.

Syntax of Cube Operation:

SQL> SELECT............

FROM.............

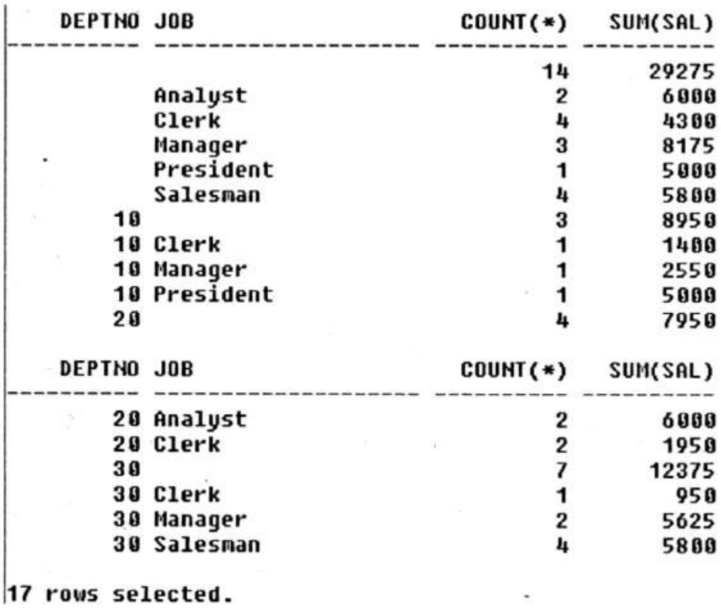
WHERE............

GROUP BY CUBE [column1, column2, ...];

Example of Cube Operation:

SQL>SELECT DEPTNO, JOB, COUNT (*), SUM (SAL) FROM EMP GROUP BY CUBE (DEPTNO, JOB);

Result:



```
    DEPTNO JOB                    COUNT(*)   SUM(SAL)
---------- -------------------- ---------- ----------
                                        14      29275
           Analyst                       2       6000
           Clerk                         4       4300
           Manager                       3       8175
           President                     1       5000
           Salesman                      4       5800
        10                               3       8950
        10 Clerk                         1       1400
        10 Manager                       1       2550
        10 President                     1       5000
        20                               4       7950

    DEPTNO JOB                    COUNT(*)   SUM(SAL)
---------- -------------------- ---------- ----------
        20 Analyst                       2       6000
        20 Clerk                         2       1950
        30                               7      12375
        30 Clerk                         1        950
        30 Manager                       2       5625
        30 Salesman                      4       5800

17 rows selected.
```

Difference between Roll Up Operation and Cube Operation

Sr. No.	Rollup Operation	Cube Operation

1	Roll up operation produces only a fraction of possible subtotal combinations.	Cube operation produces subtotals of possible combinations and a grand total.
2	Roll up operation provides us only the sets in the order listed which shows aggregates for a hierarchy of values in the selected columns	Cube operation provides us the subtotals of possible combinations of columns which show aggregates for all combinations of values in the selected columns.

7.12 NESTED QUERY

- Nested query means query within another query.
- First, we evaluate the inner query (sub query) within the Where Clause.
- Then, the result of inner query (sub query) is constituted in the condition of outer query.
- The result of inner query will pass to the outer query for the preparation of final result.
- In a nested query, there can be any number of sub queries.
- Sub query is used with Where or Having Clause. Sub query cannot be used with the Order by Clause.
- A query is called a sub query and complete select statement is called a nested query.

Syntax of Nested Query:

```
SQL> SELECT <column,.......>
FROM<table>
WHERE expression operator
(SELECT <column,...>
FROM <Table>
WHERE <condition>);
```

Examples of Nested Query:

List the employee names belonging to the department of Akhil from table "EMP", then the query will be:

```
SQL> SELECT ENAME FROM EMP
WHERE DEPTNO = (SELECT DEPTNO FROM EMP
WHERE ENAME = 'Akhil');
```

Result:
```
ENAME
    -------------
    Siddharth
```

```
        Akhil
        Amit
```

Explanation:

We divide this nested query into two parts because we do not know the department to which Akhil belongs.

In the first part: SELECT DEPTNO FROM EMP

WHERE ENAME ='Akhil';

In the inner query, we have to find out the department of Akhil.

In the second part: SELECT ENAME FROM EMP

WHERE DEPTNO = 10;

In the outer query, we use the department number (result of inner query) to find out the other employee names of that department.

7.13 SUBQUERY

Subquery is a inner query within another main query (Outer query).

Special operators in subquery: There are some special operators used in subqueries to perform specific tasks which are as follows:

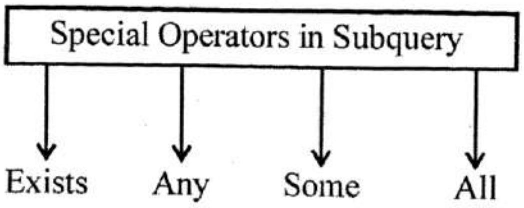


Fig. 10.13: Special Operators in Subquery

(a) Exists

It is used to check the existence of values.

It provides results in the form of true/false.

If the subquery provides any output, then the result will be true.

If the subquery does not provide any output, then the result will be false.

Example of Exists:

Display the employee details if and only if more than 8 employees working in the department number 20 from the table 'BMP'. Then the query will be:

SQL>SELECT * FROM EMP

WHERE DEPTNO = 20 AND EXISTS (SELECT COUNT (*) FROM EMP)

WHERE DEPTNO = 20

```
GROUP BY DEPTNO
HAVING COUNT (*) >20;
```

(b) Any

It is used to compares the lowest value from the set.

Any operator returns value TRUE if the comparison value matches any of the values in the list.

Examples of Any:

Display the employees whose salary is greater than any 'Manager' and he is not 'Manager'. Then the query will be:

```
SQL>SELECT ENAME, JOB, SAL FROM EMP
WHERE SAL>ANY (SELECT SAL FROM EMP WHERE JOB = 'Manager')
AND JOB<> 'Manager';
```

Result: NO ROW SELECTED

(c) Some

The Some operator and Any operator are equivalent.

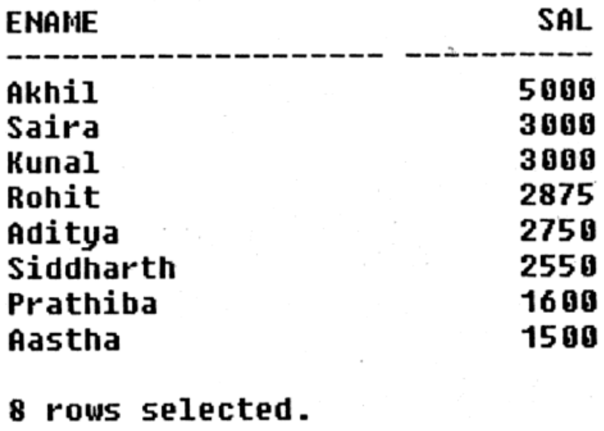
Some operator works same like Any operator i.e. it compares the lowest value from the set.

Example of Some:

Display the employee names along with their salaries from the table 'EMP', whose salary is greater than the lowest salary of an employee belonging to department number '10'. Then the query will be:

```
SQL>SELECT ENAME, SAL FROM EMP
WHERE SAL>SOME (SELECT SAL FROM EMP WHERE DEPTNO = 10);
```

Result:



```
ENAME                       SAL
-------------------- ----------
Akhil                      5000
Saira                      3000
Kunal                      3000
Rohit                      2875
Aditya                     2750
Siddharth                  2550
Prathiba                   1600
Aastha                     1500

8 rows selected.
```

(d) All

It is used to compare the highest value from the set.

All operator returns value true If the comparison value matches with all the. values in the list.

Example:

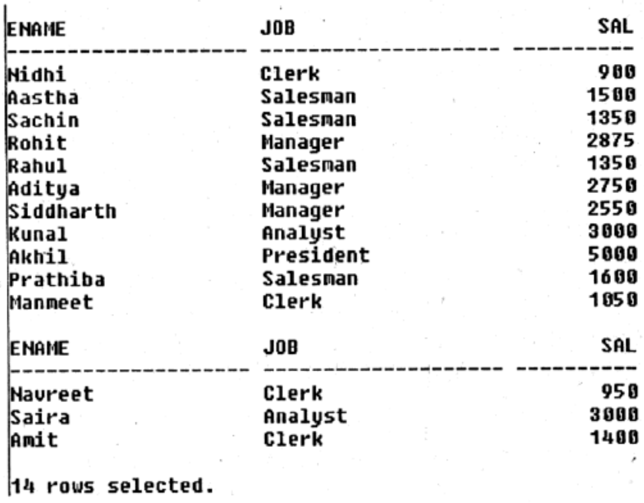
Display the employees with salary less than those whose job is 'Manager'. Then the query will be:

SQL>SELECT ENAME, JOB, SAL FROM EMP

WHERE SAL < ALL (SELECT SAL FROM EMP WHERE JOB = 'Manager')

AND JOB<> 'Manager’;

Result:



```
ENAME                JOB                         SAL
-------------------- -------------------- ----------
Nidhi                Clerk                       900
Aastha               Salesman                   1500
Sachin               Salesman                   1350
Rohit                Manager                    2875
Rahul                Salesman                   1350
Aditya               Manager                    2750
Siddharth            Manager                    2550
Kunal                Analyst                    3000
Akhil                President                  5000
Prathiba             Salesman                   1600
Manmeet              Clerk                      1050

ENAME                JOB                         SAL
-------------------- -------------------- ----------
Navreet              Clerk                       950
Saira                Analyst                    3000
Amit                 Clerk                      1400

14 rows selected.
```

7.14 VIEW

- For security reasons, it is desired that a user sees only a part of a table or group of tables.
- On the other hand, a user may like to view some of the information from many tables.
- Hence, each user views the database from his own angle without seeing the entire database.
- With the help of view, we can see a selective portion of data from one or more tables.
- A view is a virtual table. Views do not contain data their own and always tasks the data from base table.
- A view is a specific representation of data from one or more tables.
- It is stored as a select statement in the database.
- The tables on which a view is based are known as base tables.
- We can query, insert into, delete from and update from views in almost the same way as tables.
- If we make changes in the tables, then changes automatically reflects in the views.

For example: We wish that a 'Manager' should have access only to employee's name, job and department in the table 'BMP'. It means 'Manager' should not have any access to the salaries or other details of the employees. It is possible only with the help of view.

Syntax of View:

```
SQL> CREATE VIEW view_name
As
SELECT column_list
FROM table_name [WHERE Condition];
```

7.14.1 Creating a View

- We must have .the create view privilege to create a view in our schema.
- We must have to create any view system privilege to create a view in another user's schema.
- The owner of the view must have been explicitly granted privileges to access all objects referenced in the view definition.
- View's functionality depends on the privileges of the view owner. It means if the owner of the view has only the select privilege, then he/she can only select row, cannot insert, update or delete rows.

Example of Creating View:

Create a view for the 'Akhil' from the table 'EMP'. He can access only employee's name, job and department. He cannot access the salary of all the employees.

SQL>CREATE VIEW Akhil

AS SELECT ENAME, JOB AND DEPTNO FROM EMP;

7.14.2 Modifying a View

- We use the 'OR REPLACE' option to modify the view.
- If view already exists, then it will replace with new definition or a new view will be created.
- The view will become invalid whenever the base table.is altered.

Example of Modifying a View:

SQL>CREATE OR REPLACE VIEW Akhil Garg

AS SELECT * FROM EMP;

View Created.

7.14.3 Inline View

- Inline view is a subquery with an alias (correlation name) that we can use like a view inside a SQL statement.
- It appears in the 'From Clause', of the select statement and enclosed in parenthesis.

Example of Inline View:

We want to-select first 3 employee hired by the company.

SQL>SELECT ENAME<HIREDATE FROM

(SELECT ENAME<HIREDATE FROM EMP ORDER BY HIREDATE)

WHERE ROWNUM<=3;

7.14.4 Materialized View

- Materialized view is a standard object which is used to summarize, replicate, precompute and distribute data.
- It is a database object that contains the results of a query.
- It provides indirect access to table data by storing the results of a query in a separate schema object.
- A materialized view can query tables, views and other materialized views.
- Materialized views are local copies, of data located remotely or are used to create summary tables based on aggregations of a table's data.

- Materialized views, which store data based on remote tables, are also known as snapshots.
- Materialized views can be used to replicate the data at distributed sites.
- Materialized views are suitable in various computing environments such as data warehousing, decision support, distributed computing or mobile, computing.
- It is used in data warehouses so as to increase the speed of queries on a large database.
- Materialized view improves query performance by precalculating expensive join and aggregation operations.
- It is used in mobile computing to download a subset of data from central servers to mobile clients.

7.14.5 Dropping a View

- We can drop any view contained in the schema.
- The 'Drop View' statement is used to drop a view from a database.
- The 'Drop Any View' privilege is used to drop a view in another's schema.

Syntax of Dropping View:

```
SQL>DROP VIEW view_name;
```

7.14.6 Advantages of View

- The database becomes secured because user is allowed a limited view of database.
- View is used to restrict access to the database or to hide data complexity.
- The user queries become simplified.
- The user gets consisted view of database.
- We can rename the table columns by giving the different names to columns while creating views.
- Views take very little space to store because database contains only definition of view not all the present data.
- It solves the redundancy (duplication) problem.
- As the data is taken from base table, accurate and up to date information is provided by the view.
- Different views can be created on the same base table for different users.

7.15 DISADVANTAGES OF SQL

- SQL statements are passed to Oracle engine/server one at a time which generates the problem of network traffic. Every time when a SQL statement is executed, a call is made to Oracle engine/server.

- SQL cannot use the PL/SQL statements.
- SQL does not provide any procedural capabilities i.e. conditional checking, branching, looping etc.
- If any error occurs in the statement, SQL cannot fails to handle it, and then Oracle engine displays its own error message.
- PL/SQL is required to overcome the limitations of SQL. We will discuss PL/SQL in chapter 11.

7. 16 KEY POINTS

7.16.1 SQLAlias

- SQL Alias is used for columns and tables.
- SQL Alias is created to make the columns and tables more readable.
- It is used for columns when column names are big or not readable.
- It is used for tables when there are more than one tables involved in a query.

(a) SQL Alias for Columns: Display the names of all the employees from the relation 'EMP' through column alias. Then the query will be:

SQL> Select ENAME AS NAME From EMP;

In the above query, ENAME is given an alias 'NAME'. In the result, column name looks as 'NAME' instead of 'ENAME'.

Result:

```
SQL> SELECT ENAME AS NAME FROM EMP;
NAME
-------------
Nidhi
Aastha
Sachin
Rohit
Rahul
Aditya
Siddharth
```

```
Kunal
Akhil
Prathiba
Manmeet

NAME
-----------
Navreet
Saira
Amit
14 rows selected.
```

(b) ***SQL Alias for Tables:*** Display the names of all the employees from the relation 'EMP' through table alias. Then the query will be:

SQL> Select E.ENAME From EMP E;

In the above query, alias ‘E’ is defined in the relation ‘EMP'.

Result:

```
ENAME
------------
Nidhi
Aastha
Sachin
Rohit
Rahul
Aditya
Siddharth
Kunal
Akhil
Prathiba
Manmeet

ENAME
---------------
```

```
Navreet
Saira
Amit
14 rows selected.
```

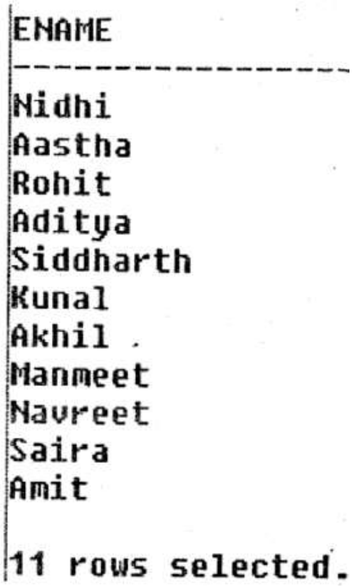
7.16.2 Null Values in SQL

- Null value represents an inapplicable or unknown value.
- It is not 0 or a blank space.
- For example: Display the names of the employees who are not eligible for the commission. Then the query will be:

  SQL>SELECT ENAME FROM EMP

  WHERE COMM IS NULL;

Result:



```
ENAME
----------------
Nidhi
Aastha
Rohit
Aditya
Siddharth
Kunal
Akhil
Manmeet
Navreet
Saira
Amit

11 rows selected.
```

7.16.3 Difference between Delete and Truncate Command

	Delete	Truncate
1	Delete command delete the table and send it to recycle bin.	Truncate command removes data from memory.
2	Delete command is slow as compared to truncate.	Truncate command is faster as compared to delete command.

3	Data can be recovered.	Data can't be recovered.
4	DML	DDL
5	In delete command, we may or may not give condition like where.	In truncate command, we do not give where condition.
6	Doesn't release the memory occupied by records of table.	Releases the memory occupied by records of table.

Questions

1. What is SQL? Write its characteristics in detail.
2. Define DCL and explain different SQL commands come under DCL.
3. Define DML and explain different command with suitable example,
4. What is the use of DDL language? Explain with suitable example
5. Query to display the different designation in department no. 20 and 30 of from table ‘EMP’.
6. Query to find the average of particular column.
7. Query to display the employees no. and name in department no. 10 and 3 0 from table 'EMP'
8. Query to select department that has total salary paid for its employees more than. 8000.

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 8 : PL/SQL

8.1 INTRODUCTION

1. PL/SQL stands for procedural language/structured query language.
2. It is an extension of SQL.
3. It was developed by Oracle Corporation to enhance the capabilities of SQL.
4. It is a combination of SQL and procedural features of standard programming languages.
5. The aim of PL/SQL is to removethe restrictions of SQL language.
6. It is not a case sensitive language. It means, we can use both the lower case and uppercase letters.
7. It is a powerful transaction processing language which extends the capabilities of SQL by adding control statements, procedures and functions.
8. It supports all the data manipulation operations and all data types of SQL.
9. It is used to overcome the limitations of SQL. It. is used to write triggers and stored procedures.
10. In short, PL/SQL is a database oriented programming language which extends Oracle SQL with procedural capabilities.

8.2 ADVANTAGES OF PL/QL

1. **Support for SQL:** It allows us touseall the SQL commands, as wellas all SQL functions,operators and data types sothat we can manipulate Oracle's data flexibly and safely.

2. **Better Performance:** Without PL/SQL, Oracle server processes SQL statements one ata time. Every time a SQL statement is issued, it must be sent over the network whichcreates more traffic. But with PL/SQL, an entire block can be sent to Oracle server at atimereduces network traffic and improves the performance.

3. **Error Handling:** PL/SQL handles errors during the execution of its program. When, anerror (exception) is found, it takes some specifications depending upon the type of theerror.

4. **BlockStructure:**It is a block structures language. It consists of blocks of code. Each program written as a block. Blocks can be nested. Each block performs a specific task.Blocks can be reused.

5. **Portability:** Programs written in PL/SQL can run anywhere. These programs can be used in another new environment without any change. It means programs written in PL/SQL are portable to any platform (any operating system) on which Oracle runs. PL/SQL programs can be reused in different environments.

6. **Higher Productivity:** PL/SQL increases productivity by adding functionality to nonprocedural tools such as forms and reports. We can use an entire PL/SQL block in an Oracle from trigger without using multiple trigger steps(macros).

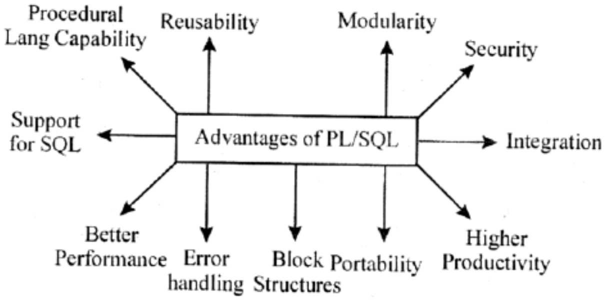


Fig. 8.1: Advantages of PL/SQL

7. **Integration:**It integrates well with SQL*PLUS and other application development products of Oracle Corporation.

8. **Security:** We can prevent client applications from modifying the sensitive data by using PL/SQL stored procedures. It enables the user to partition the application

logic from client to server and protect the data by giving proper access power to different users.

9. **Modularity:** PL/SQL divides an application/process into manageable and well defined modules such as procedures and functions. These modules are known as subprograms.

10. **Reusability:** PL/SQLprovides reusability because a single stored procedure can be used by different applications. If at any point of time procedure changes, then it will not affect the application program.

11. **Procedural Language Capability:** PL/SQLnot only supports SQL data manipulation commands but also allow control structures such as conditional statements (if else statements) and loops like (for loops). Control structures controls the procedural flow of the program and are very important PL/SQL extension to SQL.

8.3 DIFFERENCE BETWEEN SQL AND PL/SQL

Sr. No.	SQL	PL/SQL
1	SQL stands for Structured Query Language.	PL/SQL stands for Procedural Language/Structured Query Language.
2	SQL is used to manage database operations. It does not have any procedural capability.	PL/SQL is the procedural language extension to the non-procedural language SQL. It is a com6ination of SQL and procedural features of standard programming languages.
3	Only one statement is executed at a time.	The block of code is executed at a time.
4	SQL statements can be embedded within a PL/SQL program.	PL/SQL code cannot be embedded within a SQL statement.
5	It tell the database what to do and not how to do it.	It tells database how to do things.

6	There is no provision of error handling in SQL.	There is a provision of error handling in PL/SQL
7	It is used to code queries, data definition and manipulation statements.	It is used to code program blocks, functions,procedures and packages.

8.4 BLOCK STRUCTURE OF PL/SQL

1. Each program of PL/SQL consists of SQL and PL/SQL statements. It forms a PL/SQL block.
2. In PL/SQL, programs can be divided into logical blocks.
3. Comments can be used to document the code.
4. PL/SQL blocks can be nested within the other PL/SQL blocks.
5. ***PL/SQL block consists of three sections which are as follows:***

   (a) Declarative Section (Optional)

   (b) Executable Section (Mandatory)

   (c) Exception/Error Handling Section (Optional)

6. ***The structure of PL/SQL block is as follows:***

```
DECLARE
<declarations>
BEGIN
<executafole statements>
EXCEPTION
<exception handlers>
END;
```

7. ***The description of these sections of PL/SQL block structure is as follows:***

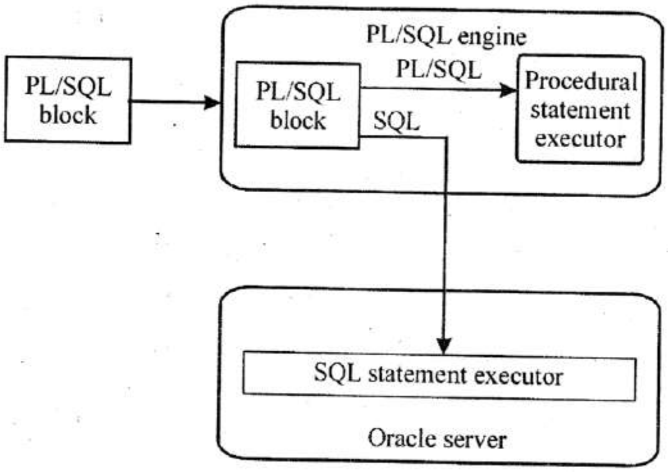
   (a) **Declarative Sections:**This section is optional. This section starts with the reserved keyword 'DECLARE'. This section is used to declare any place holders like variables, constants, records and cursors, which are used to manipulate data in the executable form.

(b) **Executable Section:** This section is mandatory. This section starts with the reserved keyword 'BEGIN' and ends with 'END'. This is the section where the program logic is written toperform any task.

(c) **Exception Section:** This section is optional. This section starts with the reserved keyword 'EXCEPTION'.

8.5 ARCHITECTURE OF PL/SQL

1. PL/SQL is not an independent product.
2. It is a run time technology.
3. It is like an engine which installed in an Oracle Server or in application development tools such as Oracle Form Builder, Oracle Reports Builders etc.
4. PL/SQL engine executes PL/SQL blocks and subprograms.
5. PL/SQL resides in two environments:
   (a) Oracle Server
   (b) Oracle Tools



Fig, 8.2: Architecture of PL/SQL

6. These two environments (Oracle Server and Oracle Tools) are independent of each other.
7. ThePL/SQL engine executes-the procedural part of the statements and sends the SQL statements to the 'SQL Statement Executor' in the Oracle server.
8. PL/SQL code is stored in the;Oracle Server.

8.6 ELEMENTS OF PL/SQL

The elements of object are used to represent real world objects, and operations. ***The PL/SQL has the following set of elements:***

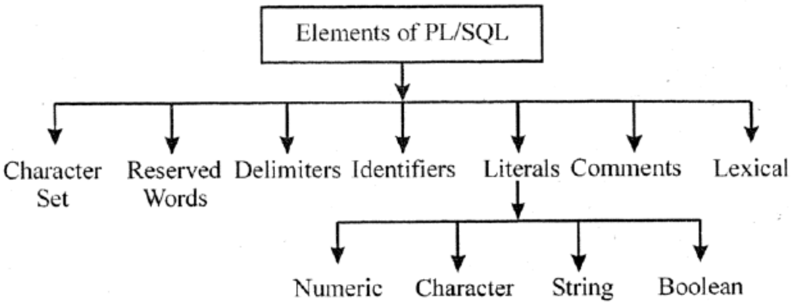


Fig. 8.3: Elements of PL/SQL

I. Character Set

1. PL/SQL programs are written as lines of text using a specific set of characters.
2. PL/SQL is not a case sensitive. In PL/SQL, programs can be written in both lower case and upper case.
3. The PL/SQL character set includes:
   - The upper case letters (A.....Z) and lower case letters (a...z).
   - Numerals (0...9)
   - Tabs, spaces and carriage returns
   - Symbols: (),+i-,*,/,<>,!,:,{}, Q,& etc.

II. Reserved Words

1. Reserved words have special, syntactic meaning to PL/SQL and cannot be redefined.
2. If we will try to redefine a reserved word, we will get a compilation error.
3. Reserved words can be written in both upper case and lower case. Generally, reservedwords are written in upper case to promote readability,
4. For example: The words 'BEGIN' and ^{C}END^{9} are reserved words, which bracket the executable part of a block or subprogram.

III. Delimiters

1. A *delimiter is* a simple or compound symbol that has a special meaning to PL/SQL.
2. Simple symbols consist of one character whereas compound symbols consist of two characters.
3. The list of simple symbols and compound symbols is as follows:

List of Simple Symbols

Simple Symbol	**Meaning**
+	Addition Operator
-	Subtraction/Negation Operator
/	Division Operator
*	Multiplication Operator
;	Statement Terminator
=	Relational Operator
<	Relational Operator
>	Relational Operator
%	Attribute Indicator
\	Character String Delimiter
(	Expression or List Delimiter
)	Expression or List Delimiter
.	Component Selector
:	Host Variable Indicator
,	Item Separator
"	Quoted Identifier Delimiter
@	Remote Access Indicator

List of Compound Symbols

Compound Symbol	**Meaning**
:=	Assignment Operator

=>	Association Operator
‖	Concatenation Operator
**	Exponentiation Operator
<>	Relational Operator
!=	Relational Operator
~=	Relational Operator
^=	Relational Operator
..	Range Operator
<<	Label Delimiter (Begin)
>>	Label Delimiter (End)
/*	Multi-line comment delimiter (begin)
*/	Multi-line comment delimiter (end)

IV. Identifiers

1. Identifiers are used to name PL/SQL program items and units, which include constants, variables, exceptions, cursors, cursor variables, subprograms and packages.
2. Identifiers follow the following rules:

- Identifiers must start will an alphabetic character.
- An Identifier consists of a letter optionally followed by more letters, numerals, underscores dollar signs and number signs. Other characters such as hyphens, slashes and spaces are not allowed.
- Identifiers can have maximum of 30 characters.
- Identifier name and column name cannot be same in a table used in the block.
- Identifiers are not case sensitive.

V. Literals

A literal is an explicit numeric, character, string or Boolean value which cannot be represented by an identifier.

(a) **Numeric Literal:** Numeric integers are represented either by simple value (129, 2.11) or by scientific integer (2E5 which means 2* 105). Numeric literals are of following two types:

- ***An Integer Literal:*** It is an optionally signed, whole number without a decimal point. For example: 129,211, +112, -29.
- ***A Real Literal:***Itis an optionally signed whole or fractional number with a decimal point. For example: 129,2.11,+1.12,-2.9.

(b) **Character Literal:** It is an individual character enclosed by single quotes (‘...‘). PL/SQL is case sensitive within the character literal. For example: It considers the literals ‘M’ and ‘m’ to be different.

(c) **String Literal:** A string literal is a sequence of zero or more characters enclosed by the single quotes. All the string literals except the null string have data types CHAR.

(d) **Boolean Literal:** Boolean literals are values not strings. Boolean literals are predefined values(True, False) and non-value (Null) which stands for a missing value (inapplicable or unknown value).

VI. Comments

1. We use comments to describe the purpose and use of each code segment.
2. It promotes the readability and aids understanding.
3. PL/SQL supports two comment styles:

(a) ***Single Line Comment:*** Single line comment begins with a double hyphen (—) and extends to the end of line.

(b) ***Multi Line Comment:***Multi line comment begins with a slash-asterisk (/*) and ends with an asterisk-slash (*/) and can span multiple lines.

VII. Lexical Units

1. A line of PL/SQL text contains the group of characters known as lexical unit.
2. It can be classified as: Delimiters, Identifiers, Literals and Comments.

8.7 DATA TYPES OF PL/SQL

1. A data type specifies the storage format, constraints and valid range of values.

2. ***In PL/SQL, data types are divided into the following four categories:***

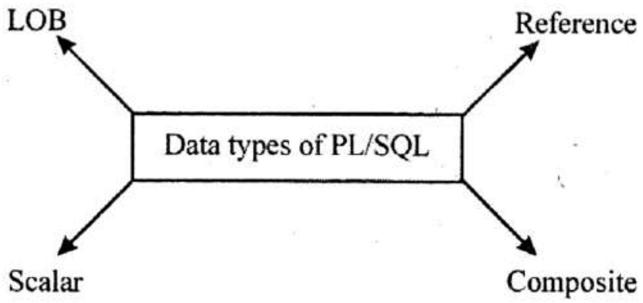


Fig. 8.4: Data Types of PL/SQL

(a) Scalar

- A scalar data type represents a single value.
- It has no internal components.
- For example: Number, Char, Boolean, Date.

(b) Composite

- *A* composite data type represents a collection of components.
- It is one that has components within it.
- It has internal components which can be manipulated individually.
- For example: Table, Record

(c) Reference

- A variable declared as reference variable can point to different storage locations over the life of a program.
- A reference data type represents a pointer that points to another item.
- For example: Ref Cursor, Ref object type.

(d) LOB

- LOB stands for large object.
- LOB data type includes BFILE, BLOB, CLOB and NCLOB.
- With LOB data type, we can store blocks of unstructured data up to 4 GB in size.
- Unstructured data means text, images, video clips and sounds.
- LOB data type allows efficient, random and piece-wise access to the data.

8.8 PL/SQL VARIABLES

1. Variables are placeholders that store the data values that can change during the execution of PL/SQL block.
2. We must first declare the variable and then use it
3. Variables can have SQL data type(such as Number, Date, Char, etc.) and PL/SQL datatype (Boolean, Binary etc.)
4. Forward references are not allowed in variables.
5. We can assign values to the variables directly from the database columns by using a 'Select Statement'.
6. **Syntax:**

   Variable _Nanie Data Type [NOT NULL: = Value];
7. ***Variables are of two types:***
   - **(a) Local Variable:** Local variables are declared in an inner block and cannot be referenced by outside the block.
   - **(b) Global Variable:** Global variables are declared in both an inner block and an outer block. It can also b6 referenced by itself.

8.9 PL/SQL CONSTANTS

1. In PL/SQL, the value of a constant remains unchanged throughout the program.
2. We can declare a constant and use it instead of actual value.
3. It is a user .defined literal value.
4. We must assign a value to a constant at the time we declare it
5. ***Syntax:***

   Constant_Name CONSTANT Data Type: = Value;

8.10 CONTROL STRUCTURES OF PL/SQL

1. Control structures are the most important PL/SQL extension to SQL.
2. It allows us to control, the behavior of the block.
3. PL/SQL supports programming language features such as conditional statements, iterative statements and sequential statements.
4. ***The various control structures are as follows:***
   - ***(i) Conditional Statement***

(ii) ***Iterative Statement***

(iii) ***Sequential Statement***

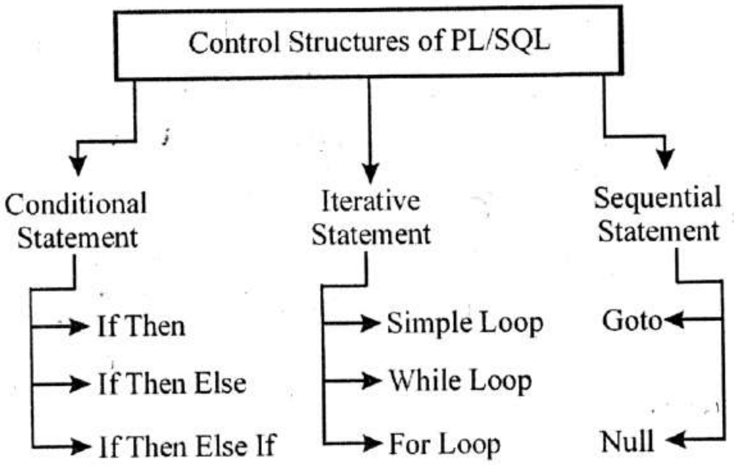


Fig.8.5: Control Structures of PL/SQL

8.10.1 Conditional Statements

It allows PL/SQL to perform the actions selectively based on conditions. The following are the forms of conditional statements:

(a) **If Then Statement:** It is the simplest form of If statement. It associates a condition with a sequence of statements enclosed by the keywords 'Then' and 'End If'.

Syntax:

```
IF condition THEN
Sequence of Statements;
END IF;
```

(b) **If Then Else Statement:**It adds the keyword 'Else' followed by the alternative sequence of statements.

Syntax:

```
IF condition THEN
Statement 1;
ELSE
Statement 2;
END IF;
```

(c) If Then Elself Statement: It adds the keyword 'Elself' to introduce the additional conditions.

Syntax:

```
IF condition1 THEN
Sequence of Statements1;
ELSEIF condition 2 THEN
Sequence of Statements2;
ELSE
Sequence of Statements3;
END IF;
```

8.10.2 Iterative Statements

These are also known as Loop Control Structures. These statements are used when we want to repeat the execution of one or more statements for specified number of times. The following are the forms of iterative control statements:

(a) Simple Loop Statement: It is also known as basic or infinite loop. It is used when a set of statements is to be executed at least once before the loop terminates. An 'Exit Condition' must be specified in the loop, which exits the process from the loop. If 'Exit Condition' is not specified in the loop, the loop will get into an infinite number of iterations (loops).

Syntax:

```
LOOP
Statements;
EXIT [WHEN CONDITION];
LOOP;
```

(b) While Loop Statement: It is used to repeat a sequence of statements until the controlling condition is 'True'. It evaluates the condition at the start of each iteration and terminates when the condition is'False'.

Syntax:

```
WHILE <condition>
```

LOOP Statements;

END LOOP;

(c) For Loop Statement: It is used to execute a set of statements for a predefined number of times. Between the given start and end integer values. Iteration occurs. The counter is incremented by 1 and loop exits when the counter reaches the value of the end integer.

Syntax:

```
FOR counter IN <Start Integer Value> ...<End Integer Value>
LOOP Statements;
END LOOP;
```

8.10.3 Sequential Statements

In PL/SQL all the blocks execute in top-down sequential process (begin statement to end statement). We use sequential statements to change the sequence of execution of statements. The following are the forms of sequential statements:

(a) GOTO Statement: It immediately transfers program control unconditionally to a named statement label. The statement label should be unique.

Syntax:

```
GOTO <<LabelI_name>>;
```

(b) NULL Statement: Itdoes nothing and passes control to the next statement.

Syntax:

```
NULL;
```

8.11 CURSORS IN PL/SQL

1. Oracle uses work area to execute SQL statements and store processing information.
2. In PL/SQL, when we want to execute a SQL statement, Oracle server just opens the memory area (private SQL work area) to execute that command. That private SQL work area is known as cursor.
3. A cursor is a memory area (private SQL work area) that Oracle uses to execute SQL statements.

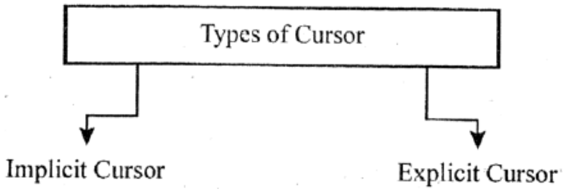
4. It is a temporary workarea created in a system memory when a SQL statement is executed.

5. This memory area (private SQL work area) is also used to store the data retrieved from the database.

6. The set of rows the cursor holds is known as Active Set.

7. A cursor can hold-more than one row, but can process only one row at a time.

8.11.1 TYPES OF CURSOR



Fig, 1L6: Types of Cursor

I. Implicit Cursor

1. Implicit cursors are declared by PL/SQL implicitly.

2. Implicit cursor is created to execute the DML (data manipulation language) statements such as Select, Insert, Update and Delete.

3. When DML operations are performed with implicit cursor, it automatically reserves somememory area for the execution of operations.

4. After the completion of DML operations, it releases the memory area.

5. Attributes of Implicit Cursor:

Oracle provides some attributes (%Found, %Not found, %'Isopen, %Rowcount) to check the status of DML operations. These attributes are known as implicit cursor attributes.

Sr. No.	Attribute	Retpra Value	Example
1	%FOUND	• The return value is 'TRUE', if the DML statements like Insert, Update and Delete affect at least one row and if Select.... Into statement return at least	SQL%FOUND

		one row. • The return value is 'FALSE', if the DML statements like Insert, Update and Delete do not affect any row and if Select.... Into statement does not return any row.	
2	%NOTFOUND	• The return value is 'TRUE', if the DML statements like Insert, Update and Delete do not affect any row and if Select.... Into statement does not return any row. • The return value is 'TALSE', if the DML statements like Insert, Update and Delete affect at least one row and if Select.... Into statement return at least one row.	SQL%NOTFOUND
3	%ISOPEN	The return value is always 'False' because Oracle automatically closes an implicit cursor after the successful execution of SQL operations.	SQL%ISOPEN
4	%ROWCOUNT	It returns (counts) the number of rows affected by the DML operations (Select, Insert, Update and Delete).	SQL%ROWCOUNT

6. Disadvantages of Implicit Cursor:

- Implicit cursors are less efficient and slightly slow as compared to explicit cursors.
- Implicit cursors provide less programming control. Unlike explicit cursors, implicit cursors cannot be opened and fetched automatically.

- Implicit cursors are more vulnerable to data errors because it has less programming control.

II. Explicit Cursor

1. Explicit cursors are declared explicitly by the user. Explicit cursors also are known as User Defined Cursors.
2. An explicit cursor is defined in the declaration sectionof the PL/SQL block.
3. Explicit cursor is declared in the declarative part of PL/SQL block to take control over query operations.
4. Explicit cursor can store multiple records, but can process one record (current row) at a time.
5. After processing the rows in the explicit cursor, we deallocate the memory occupied by the cursor using CLOSE statement.

6. Attribute of Implicit Cursor:

Orcle provides some attributes (%Found, %Not found, %Isopen, %Rowcount) to avoid errors which accessing cursors through OPEN, FETCH and CLOSE statements. These attributes are known as implicit cursor attributes.

Sr. No.	**Attribute**	**Return Value**	**Example**
1	%FQUND	• The return value is 'TRUE', iffetch statement returns at least one row. • The return value is 'FALSE', iffetch statement does not return any row.	Cursor_name%FOUND
2	%NOTFOUND	• The return value is 'TRUE', if not statement does not return any row. • The return value is 'FALSE', if fetch statement returns at	Cursor_name%NOTFOUND

		least one row.	
3	%ISOPEN	• TRUE,if the cursor is already open in the program. • FALSE, if the cursor is not opened in the program.	Cursor_name%ROWCOUNT
4	%ROWCOUNT	It returns (counts) the number of rowsfetchedby the fetchstatement. If no row is returned, the PL/SQL statement an error.	Cursor_name%ROWCOUNT

8.12 EXCEPTION HANDLING IN PL/SQL

1. Exception is basically an error.
2. Exceptions are identifiers in PL/SQL to terminate its main body of actions.
3. When an exception is raised, then Oracle searches for an appropriate 'exception handler'in the exception section.
4. Exception handler is used to handle the exceptions and to perform actions before the block terminates.
5. Exception handling part is used to specify the statements to be executed when an exception occurs.
6. When an error/exception arises during program execution, then the normal execution stops and the control transfers to the exception handling part of the PL/SQL block or subprogram.
7. Only one exception can be raised in the block. After the error is handled, the control does not return to the execution section.
8. An exception cannot be declared twice in the same block.

9. Exceptions declared in a block are considered as local to that block and global to its sub blocks.

10. An enclosing block cannot access exceptions declared in its sub block.

11. The PL/SQL exception message consists of three parts:

- ***Type of Exception***
- ***An Error Code***
- ***A Message***

12. Syntax:

```
DECLARE
        Declaration Section
BEGIN
        Execution Section
EXCEPTION
        When Ex_Name1Then
        Error Handling Statements
        When Ex_Name2 Then
        Error Handling Statements
        When Others Then    .
        Error Handling Statements
END;
```

8.12.1 Guidelines to Avoid and Handle the Exceptions

The following are some guidelines to avoid and handle the exception:

- When there is any possibility of occurring an error, then add error checking code topredict an error.
- When there is any possibility of occurring an error, then use exception handler tohandle it.
- Test the code with different combinations of bad data to check its potential.

- Learn the names and causes of possible errors so that we can easily find and handle them.

8.12.2 Types of Exception

The exceptions are of following two types:

(a) Predefined Exceptions

- It is also known as Internal Exceptions.
- These are automatically raised by the Oracle team
- We can handle them directly within our program without declaring them.
- The following table defines some predefined exceptions:

Sr. No.	Exception Name	Oracle Error Number	Reason
1	CURSOR_ALREADY_OPEN	ORA-06511	When we open a cursor that is already open.
2	INVALID__CURSOR	ORA-01001	When we perform an invalid operation on a cursor that in not opened.
3	LOGIN_DENIED	ORA-01017	When we want to login Oracle with wrong username and password.
4	INVALID_NUMBER	ORA-01722	Conversion from character to number denied.
5	NO_DATA_FOUND	ORA-Q14Q3	When a Select...Into clause does not return any row from a table,
6	ZERO_DIVIDE	ORA-01476	When we attempt to divide a number by zero.

(b) User-defined Exceptions

- User defined exceptions are declared and defined by the user.
- These are explicitly declared in the declaration section.

- These are raised explicitly by raise statements,unlike predefined exceptions that are raised implicitly.

8.13 EXCEPTION PROPAGATION

When an exception is raised in the executable section of PL/SQL block, then it will handle in the following manner.

1. If an exception is handled in the current block, then the control passes to the enclosing block.
2. If an exception is not handled in the current block, then exception propagates.
   - Propagates means circulates. We just circulate an exception until it handles.
   - It means exception is sent to enclosing blocks from inside to outside until a handler is found or no more blocks to search,
   - If no handler is found for exception, then an exception is sent to the host environment.

8.14 SUBPROGRAMS

1. In PL/SQL, programs can be stored in the database as stored programs. Such stored programs are known as Subprograms.
2. Subprograms can be invoked whenever required.
3. We can declare and define a subprogram within either a PL/SQL block or anothersubprogram.
4. The main function of subprogram is to break the program into smaller and manageable parts because smaller programs are easier to maintain and debug as compared to large program.
5. These smaller and manageable parts are known as modules and this process is known as Modularization.
6. Subprograms provide easy maintenance because code is located at one place and we can easy modify it in this single location.

8.14.2 Advantages of Subprograms

- Modularity: Subprogram breaks the program into smaller and manageable parts which are easier to maintain and debug.
- Extensibility: It provides the facility to add functionality. It allows creatingnew program modules without affecting the old ones.
- Reusability: Any number of applications can use and execute the subprograms. A subprogram can be used by various number of applications is known as reusability.
- Better Performance: Subprogram can reduce the number of calls from application to Oracle, Reducing calls automatically increase the performance.
- Security: Subprogram helps to maintain the database security. It can restrict the users to perform specific tasks with security privileges.
- Abstraction: Subprograms aid abstraction as all the internal and compilation details are hidden from the user.
- Memory Allocation: Subprograms require less memory because it loads only a single copy of subprogram into memory for multiple users.

8.14.2 Block Structure of PL/SQL Subprograms

PL/SQL Subprograms consists of three sections which are as follows:

(a)Declarative Section

(b) Executable Section

(c)Exception/Error Handling Section

(i) ***Declarative Section: It*** contains the declarations of types, cursors, constants, variables,exceptions and nested subprograms.

(ii) ***Executable Section:*** It contains statements that assign values, control execution andmanipulate the data.

(iii) ***Exception/Error Handling Section: It*** contains exception handlers which deal with exceptions (errors) raised during execution.

8.14.3 Types of Subprograms

The subprograms are of following two types:

(a) ***Procedures***

(b) Functions

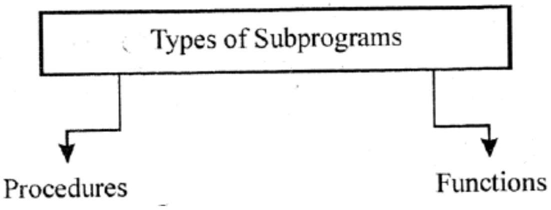


Fig. 11.7:Types of Subprograms

8.14.4 Procedures

1. We use procedure to perform an action.
2. It is a subprogram that performs specific task.
3. It may or may not return a value.
4. It has declaration section, an executable section and exception handling section.
5. It accepts parameters of type IN, OUT and IN OUT.
6. ***Syntax:***

```
CREATE [OR REPLACE] PROCEDURE Procedure_Name
        [{Parameter 1 [Mode 1] Datatype1, Parameter 2 [Mode 2] Datatype2,…}]
        IS I AS
        [Local Declaration]
BEGIN
PL/SQLExecutable Statements
[EXCEPTION
        Exception Handler]
END [PROCEDURE NAME];
```

7. ***The following are the two types of procedures:***

(a) Local Procedures

- Local procedure is defined in the declaration section of PL/SQL block.
- Its scope is limited to the parent block from where it belongs.
- It cannot be defined outside the block from where it created.
- It can be called anywhere in the module section area.

(b) Stored Procedures

- It performs one or more specific tasks.
- It displays an error message if any error occurs.

8.14.5 Functions

1. The difference between a procedure and a function is that a function must return a value but a procedure may or may not return a value.
2. We use function to compute a value.
3. Function has a return clause.
4. Like procedure, a function has declaration section, an executable section and exception handling section.
5. ***Syntax:***

```
CREATE [OR REPLACE] FUNCTIONFunction_Name
        [{Parameter1 [Mode 1] Datatype1, Parameter 2 [Mode 2]Datatype1,........}]
        IS I AS
        [Local Declaration]
BEGIN
PL/SQL Executable Statements
[EXCEPTION
        Exception Handler]
END [FUNCTION NAME];
```

6. ***The following are the two types of functions:***
   (a) *Local* ***Functions:*** It is defined in the declaration section of PL/SQL block. It cannotbe called by any block of PL/SQL outside the inner block.
   (b) ***Stored Functions:*** It is also known as user defined function. It is set of PL/SQLstatements we can call by name.

8.15 STORED PACKAGES

1. A stored packageor package is the collection of PL/SQL elements.
2. It is a database object which groups logically,related PL/SQL objects such as cursors,exceptions, subprograms,procedures, functions, variables etc. into a single container.
3. It stores PL/SQL objects which perform similar tasks into a single container.

4. It is like a library which stored once in Oracle database and used by many applications.

5. **The stored packages are of following two parts:**

   (a) Package Specification

   (b) Package Body

(a) Package Specification

- In short form, package specification is known as 'Spep'.
- It holds public declarations, which are visible to the application.
- The scope of these declarations is local to the database schema andglobal to the package.
- It is required when we create a new package.
- We use the 'Create Package Statement' to create a new package or package specification.
- ***Syntax:***

```
PACKAGE Package_Name
IS
        [Declaration of Variables and Types]
        [Headers of Cursors]
        [Headers of Procedures and Functions]
END [Package Name];
```

(b) Package Body

- It holds the implementation details and private declarations.
- It contains the code that implements the package specification
- It fully defines the cursors and subprograms declared in the package specification.
- We use the 'Create Package Body Statement' to create a package body.
- ***Syntax:***

```
PACKAGE BODY Package_Name
IS
```

```
            [Declaration of Variables and Types]
            [Header and Select Statement of Cursors]
      [BEGIN
            Executable Statements]
      [EXCEPTION
            Exception Handlers]s
      END [Package Name];
```

6. 'Drop Package Command' is used to drop a package. It drops the specification and body of the package.

Syntax:

```
      DROP PACKAGE <Package_Name>;
```

8.16 TRIGGERS

1. A trigger is a PL/SQL block structure which is fired when DML statements (Select, Insert,Update, Delete) execute on a database table.
2. It is stored in the database and executed automatically when specific event (user actions orsystem actions) occurs in the database.
3. It is a PL/SQL program unit which associates with specific database table.
4. Triggers are stored as text and compiled at execution time.
5. It does not include much code in them but it call out previously stored procedures or packages.
6. A database trigger includes SQL and PL/SQL statements to executes a unit and invokeother stored procedures.
7. It can be defined on tables and on views.
8. It is used to improve the performance of Oracle in order to provide a more convenientdatabase.
9. Trigger cannot perform commit or rollback operations.

8.16.1 Guidelines for Designing Triggers

1. Triggers can be used only when it is necessary. The excessive use of triggers can result in complex interdependencies which may be difficult to maintain in large applications.
2. Triggers guarantee that when a specific operation is performed, related actions are performed.
3. Limit the size of triggers.
4. If the logic for the trigger is very lengthy, create stored subprograms, put the code into stored subprograms and invoke them in the trigger body.
5. Do not create recursive triggers.
6. Do not define triggers that duplicate features already built into the Oracle database.
7. Use database triggers only for centralized, global operations that must fire for the triggering statements, regardless of which user or database application issues the statement.

8.16.2 States of Triggers

A trigger can be in either of two states:

I. Enabled State

1. An enabled trigger executes its trigger body if a triggering statement is entered and the trigger restriction (if any) evaluates to True.
2. By default, a trigger is created in enabled state.
3. To enable all triggers defined for a specific table, use the *'Alter Table Statement'* with *'Enable Clause'* and *'All Triggers Option'*.

   For example: To enable all the triggers defined for the table 'Student', the statement will be:

   ```
   SQL>ALTER TABLE STUDENT ENABLE ALL TRIGGERS;
   ```
4. To enable a disable trigger, use the *'Alter Table Statement'* with *'Enable Clause'*.

   For example: To enable the disabled trigger named 'Student', the statement will be:

   ```
   SQL>ALTER TRIGGER STUDENT ENABLE;
   ```

II. Disabled State

1. A disabled trigger does not execute its trigger body, even it a triggering statement is entered and the trigger restriction (if any) evaluates to True.
2. To create a trigger in disabled state, we use the disable clause in the create trigger statement.
3. To disable all triggers defined for a specific table, use the *'Alter Table Statement'* with *'Disable Clause'* and *'All Triggers Option'*.

   For example: To disable all the triggers defined for the table 'Student', the statement will be:

   SQL>ALTER TABLE STUDENT ENABLE ALL TRIGGERS;
4. To disable a trigger, use the *'Alter Table Statement'* with *'Disable Clause'*.

   For example: To disabled the trigger named 'Student', the statement will be:

   SQL>ALTER TRIGGER STUDENT DISABLE;

8.16.3 Parts of Trigger

A trigger has the following three parts:

I. Triggering Event or Statement

II. Trigger Restriction

III. Trigger Action

When any of the events occur, the trigger fires automatically and PL/SQL block performs the action. The trigger action is a procedure thatcontains the code to be executed when the trigger fires.

I. Triggering Event or Statement

1. It is the SQL statement that causes a trigger to be fired.
2. It can specify multiple SQL statement.
3. It can be an Insert, Update or Delete statement on a table.

II. Trigger Restriction

1. It must specify a Boolean expression that must be 'True' for the trigger to fire.
2. Its function is to control the execution of a trigger conditionally.

3. The trigger action is not run (executed) if the trigger restriction evaluates to 'False' or 'Unknown'.

III. Trigger Action

1. It is the procedure that contains the code to be executed when the trigger fires.
2. Trigger Action executes the PL/SQL block when the triggering statement is issued and trigger restriction evaluatesto'True'.

8.16.4 Types of Trigger

The different types of triggers are as follows:

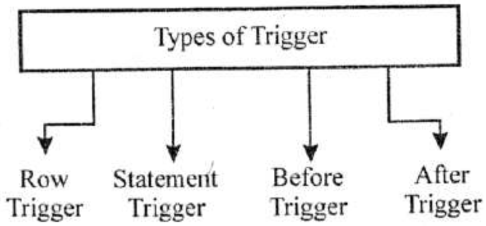


Fig. 11.8: Types of Triggers

I. Row Triggers

1. A row trigger is fired each time the table is affected by the triggering statement. A row trigger fires once onbehalf of the triggering statement.
2. If a triggering statement affects no rows; a row trigger is notrun.
3. Row triggers are useful if trigger action depends on number of rows affected.

II. Statement Triggers

A statement trigger is fired once on behalf of the triggering statement, regardless of the number of rows in the table that the triggering statement affects, even no rows are affected. It is default type of trigger.

III. Before Triggers

1. Before triggers run the trigger action before the triggering statement is run.
2. These triggers are used to check the validity of data before the action is performed.

IV. After Triggers

1. After triggers run the trigger action after the triggering statement is run.
2. These triggers are used when we want the triggering statement to complete before executing the trigger action.
3. Instead of Triggers

(i) Instead of trigger provide a transparent way of modifying views that cannot be modifieddirectly through DML statement.

(ii) With the help of this trigger, Oracle fires the trigger instead of executing the triggeringstatement.

8.16.5 Creating and Dropping A Trigger

Creating a Trigger

1. We use the 'Create Trigger Statement' to create a trigger.
2. By default, a trigger is created in enabled state.
3. If we want to create a trigger in disabled state, thenwe use the 'Disable Clause' of the 'Create Trigger Statement'.
4. **Syntax:**

```
CREATE [OR REPLACE] TRIGGER Trigger_Name
BEFORE I AFTER I INSTEAD OF
        INSERT [OR] I UPDATE [OR] I DELETE [OF Column_Name]
        ONTable_Name
[REFERENCING OLD AS Old, NEW AS New]]
[FOR EACH ROW [WHEN CONDITION]]
BEGIN
———SQL STATEMENTS
END;
```

Dropping a Trigger

1. We can delete the trigger with the help of 'Drop Command'.
2. If we want to delete the trigger'Monika$^{9}_{5}$ then the statement will be:

   SQL>DROP TRIGGER MONIKA;

8.16.7 Advantages of Triggers

- It maintains the replicate tables.
- It prevents the invalid transactions.
- It implements complex security authorizations.
- It implements complex business rules which cannot be implemented by using integrity constraints.
- It automatically generates derived columns.
- It automatically performs an action when another concerned action takes place.

Questions

1. What is the difference between SQL and PL/SQL? Explain.
2. Discuss the block structure, of PL/SQL block.
3. Explain the elements of PL/SQL in detail.
4. Discuss the control structures of PL/SQL in detail.
5. Discuss the architecture of PL/SQL block.
6. Explain the following:
   (a) PL/SQL Variables.
   (b) PL/SQL Constants.
   (c) Data types of PL/SQL.
7. What is exception? Discuss the usages of pre-defined and user-defined exceptions in PL/SQL.
8. What is cursor? Discuss the role of implicit and explicit cursor.
9. How is exception handling performed in PL/SQL?
10. What is subprogram? Explain the types and block structure of PL/SQL subprogram in detail.
11. What are stored packages?
12. What is trigger? Explain the various types of triggers in Oracle.

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University
(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM
ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-I
Course: PROBABILITY & STATISTICAL ANALYSIS (MSCS-1-03T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

M.Sc. (Computer Science) Programme Coordinator:
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:
Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Business Management & Commerce:
Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Social Sciences and Liberal Arts:
Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

\

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA
A State University Established by Govt. of Punjab vide Act No. 19
of 2019 and Approved Under section 2(f) of UGC

PROGRAMME COORDINATOR

Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

COURSE COORDINATOR & EDITOR

Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Course: Probability & Statistical Analysis	
Course Code: MSCS-1-03T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Apply measures of central tendency for analysis of data.
CO2	Learn tabulated and graphical representation techniques for discrete and continuous data.
CO3	Infer the concept of correlation and regression for two or more related variables.
CO4	Understand the fundamentals of statistics to apply descriptive measures and probability for data analysis.
CO5	Understand the concepts of Random Variable, Probability Mass Function and Density Function.

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,

Dean Academic Affairs

Name of Programme: M.Sc. (Computer Science)
Name of Course: Probability & Statistical Analysis
Course Code: MSCS-1-03T
Semester 1

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

INSTRUCTIONS FOR THE PAPER SETTER/EXAMINER

1. The syllabus prescribed should be strictly adhered to.
2. The question paper will consist of three sections: A, B, and C. Sections A and B will have four questions from the respective sections of the syllabus and will carry 10 marks each. The candidates will attempt two questions from each section.
3. Section C will have fifteen short answer questions covering the entire syllabus. Each question will carry 3 marks. Candidates will attempt any ten questions from this section.
4. The examiner shall give a clear instruction to the candidates to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.
5. The duration of each paper will be three hours.

INSTRUCTIONS FOR THE CANDIDATES

Candidates are required to attempt any two questions each from the sections A and B of the question paper and any ten short q questions from Section C. They have to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.

SECTION-A

Unit I: Origin and Development of Statistics: Scope, limitation and misuse of statistics. Types of data: primary, secondary, quantitative and qualitative data. Types of Measurements: nominal, ordinal, discrete and continuous data.

Unit II: Presentation of Data by Tables: construction of frequency distributions for discrete and continuous data, graphical representation of a frequency distribution by histogram and frequency polygon, cumulative frequency distributions. Classification and Graphical representation of data (Pie Chart, Bar Diagram, Histogram, Frequency Polygon, Ogive Curve, etc.).

Unit III: Measures of Central Tendency: Arithmetic Mean, Median and Mode and its Graphical representation, Measures of dispersion – range, variance, mean deviation, standard deviation and Coefficient of variation, Concepts and Measures of Skewness and Kurtosis.

Unit IV: Descriptive Statistics: Exploratory data analysis, Coefficient of variation, Data visualization, Scatter diagram, Grouped data. 27

SECTION-B

Unit V: Correlation: Scatter plot, Karl Pearson coefficient of correlation, Spearman's rank correlation coefficient, multiple and partial correlations (for 3 variates only). Regression: Introduction to regression analysis: Modelling a response, overview and applications of regression analysis, Simple linear regression (Two variables)

Unit VI: Mathematical and Statistical probability: Elementary events, Sample space, Compound events, Types of events, Random experiment, sample point and sample space, event, algebra of events.

Unit VII: Definition of Probability: classical, empirical and axiomatic approaches to probability, properties of probability. Theorems on probability, conditional probability and independent events

Unit VIII: Statistical inference: Concept of Random Variable, Probability Mass Function & Density Function, Mathematical Expectation (meaning and properties), Moments, Moment Generating Function and Characteristic Function.

Reference Books:

Gupta, S.C. and Kapoor, V.K., “Fundamentals of Mathematical Statistics”, Sultan & Chand & Sons, New Delhi, 11th Ed.

Hastie, Trevor, et al. “The elements of Statistical Learning”, Springer.

Ross, S.M., “Introduction to Probability and Statistics”, Academic Foundation.

Papoulis, A. and Pillai, S.U., “Probability, Random Variables and Stochastic Processes”, TMH.

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

UNIT I: ORIGIN AND DEVELOPMENT OF STATISTICS

STRUCTURE

1.0 OBJECTIVES

In this module, we will try to understand about the historical development of statistics. Various definitions given by different scientists will also be explored in this module. Further need of statistics along with its applications will also be investigated. Limitations and misuse of statistics will also be looked upon. In the end, various types of data used in statistics will also be explored.

1.1 INTRODUCTION

This module is designed to know about the development of statistics using historical background. Statistics is not a subject that can be studied alone, rather it proves to be basis for almost all other subjects as data handling is essential in almost all fields of life. Due to this reason, a number of definitions are given to statistics. Some of the major fields where statistics is prominently used are planning, finance, business, agriculture, biology, economics, industry, education, etc. Actually a country‟s growth is very much dependent on statistics as without statistics it would not be possible to estimate the requirements of the country.However, statistics is based on probabilistic estimations and therefore not actual (in some cases), therefore can‟t be believed with 100% guarantee. Also some people may misuse statistics for their own benefits. But, still statistics is very essential and very much need of thelife. There are various classifications of the data used in statistics viz., continuous, discrete, nominal, ordinal, etc. The data can be used as per requirement for a particular application.

1.2 MAIN CONTENT

1.2.1 Origin and Development of Statistics

The term Statistics, is not a new term rather it is as old as being human society. However, it has come up leaps and bounds with the emergence of data analytics and machine learning and the role of statistics in those areas. Directly and indirectly, statistics has been used right from the ancient days, although as a subject it was introduced much later.

In the earlier days it was regarded more or less as a by-product of the administrative activity for analysing the various activities. The word Statistics is said to be derived from a Latin word „status‟ or an Italian word „statista‟ or may be the German word „statistik‟ or the even from the French word „statistique‟, each of which actually means a political state. The term was used to collect information about the population in a country so that various schemes could be implemented depending on the requirement and size of population.

If we talk specifically about India, various efficient mechanisms for collecting various type of official statistics existed even 2000 years ago, during the tenure of Chandragupta Maurya. In history, evidences have been found for the existence of an excellent system of collecting important statistics and registration of births and deaths, even before 300 B.C. are available in Kautilya‟s „Arthashastra‟. Further, the records of land, agriculture and wealth statistics were maintained by Todermal, a well-known land and revenue minister in the era of Akbar (1556-1605 A.D). A detailed information about the administrative and statistical surveys conducted during Akbar‟s era is very much available in the book “Ain-e- Akbari” written by Abul Fazl (1596-97), one of the nine gems of Akbar.

The Statistics was applied for collecting the data related to the movements of heavenly bodies in the 16th century, viz., stars and planets so that their position may be known and various Eclipses may be predicted. Further, in the seventeenth century, the vital statistics was originated. Captain John Graunt of London (1620-1674) is known as the Father of vital statistics. He was the person behind a systematic study of the birth and death statistics.

Modern stalwarts involved in the development of the subject of Statistics, are various Englishmen, who did revolutionary work in applying the Statistics to different disciplines. Francis Galton (1822-1921) pioneered the study of „Regression Analysis‟ in Biometry; Karl Pearson (1857-1936), founder of the greatest statistical laboratory in England pioneered the study of „Correlation Analysis‟. His Chi-Square test (X^2-test) of Goodness of Fit is one of the most important tests of significance in Statistics; W.S. Gosset introduced t-test, which escorted the era of exact (small) sample tests.

However most of the work in the statistical theory during the past few decades can be attributed to a single person Sir Ronald A. Fisher (1890-1962), who applied statistics to a variety of diversified fields such as genetics, biometry, psychology and education, agriculture, etc., and who is rightly termed as the Father of Statistics. He not only enhanced the existing statistical theory, but also he is the pioneer in Estimation Theory; Exact (small) Sampling Distributions; Analysis of Variance and Design of Experiments. One can easily saythat R.A. Fisher is the real giant in the development of the theory of Statistics. It is due to the outstanding contributions of R. A. Fisher that put the subject of Statistics on a very firm footing and earned for it the status of a full-fledged science.

1.2.2 Definitions of Statistics

Statistics has been defined by number of authors in different ways. The main reason for the various definitions are the changes that has taken place in statistics from time to time. Statistics in general is defined in two different ways viz., as „statistical data‟, i.e., based on numerical statement of data and facts, and as 'statistical methods', i.e., based on the principles and techniques used in collecting and analysing such data. Some of the important definitions under these two categories are given below.

Statistics as Statistical data

Webster defines Statistics as "classified facts representing the conditions of the people in a State, especially those facts which can be stated in numbers or in any other tabular or classified arrangement." Bowley defines Statistics as "numerical statements of facts in any department of enquiry placed in relation to each other."

A more exhaustive definition is given by Prof. Horace Secrist as follows: "By statistics we mean aggregation of facts affected to a marked extent by multiplicity of causes numerically expressed, enumerated or estimated according to reasonable standards of accuracy, collected in a systematic manner for a predetermined purpose and placed in relation to each other."

Statistics as Statistical Methods

Bowley himself has defined Statistics in a number of ways:

(i) Statistics may be called the science of counting.

(ii) Statistics may rightly be called the science of averages.

(iii) Statistics is the science of the measurement of social organism, regarded as a whole in all its manifestations.

However, these definitions are not complete in any sense as they don‟t provide the complete view of statistics. According to Boddington, "Statistics is the science of estimates and probabilities." Again this definition is not complete as statistics is not just probabilities and estimates but more than that.

Some other definitions are: "The science of Statistics is the method of judging collective, natural or social phenomenon from the results obtained from the analysis or enumeration or collection of estimates."- as provided by King.

"Statistics is the science which deals with collection, classification and tabulation of numerical facts as the basis for explanation, description and comparison of phenomenon." as given by Lovitt.

But the best definition is the one given by Croxton and Cowden, according to whom Statistics may be defined as "the science which deals with the collection, analysis and interpretation of numerical data."

1.2.3 Importance and Scope of Statistics

Statistics is primarily used either to make predictions based on the data available or to make conclusions about a population of interest when only sample data is available. In both cases statistics tries to make sense of the uncertainty in the available data. When making predictions statisticians determine if the difference in the data points are due to chance or if there is a systematic relationship. The more the systematic relationship that is observed the better the prediction a statistician can make. The more random error that is observed the moreuncertain the prediction.

Statisticians can provide a measure of the uncertainty to the prediction. When making inference about a population, the statistician is trying to estimate how good a summary statistic of a sample really is at estimating a population statistic.

For computer students, knowing the basic principles and methods in statistics could help them in doing their research work like comparing the speed of internet connection in different countries and the probability of how many times does each experience the same level of internet connection speed in a week, month or year. It could also be helpful in determining the best operating system to use. Whenever there is the need to compare data and know the best option that we should take statistics can give the answer.

Statistics is having applications in almost all sciences - social as well as physical such as biology, psychology, education, economics, business management, etc. It is hardly possible to think of even a single department of human activity where statistics is not involved. It has rather become indispensable in all phases of human endeavour.

Statistics and Planning

Statistics is mother of planning. In the modern age which is termed as 'the age of planning', almost all over the world, particularly of the upcoming economies, are resorting to planning for the economic development. In order that planning is successful, it must be based soundly on the correct analysis of complex statistical data.

Statistics and Economics

Statistical data and technique of statistical analysis have proved immensely useful in solving various economic problems, such as wages, prices, analysis of time series and demandanalysis. A number of applications of statistics in the study of economics have led to the development of new disciplines called Economic Statistics and Econometrics.

Statistics and Business

Statistics is an essential tool for production control. Statistics not only helps the business executives to know the requirements of the consumers, but also for many other purposes. The success of a business actually depends upon the accuracy and precision of his statistical forecasting. Wrong analysis, due to faulty and inaccurate analysis of various causes affecting a particular phenomenon, might prove to be a disaster. Consider an examples of manufacturing readymade garments. Before starting one must have an overall idea as to „howmany garments are to be manufactured', 'how much raw material and labour is needed for that', and 'what is the quality, shape, color, size, etc., of the garments to be manufactured'. If these questions are not analysed statistically in a proper manner, the business is bound to be failed. Therefore, most of the large industrial and commercial enterprises are employing trained and efficient statisticians.

Statistics and Industry

In industry, statistics is very widely used in 'Quality Control'. In production engineering, to find whether the product is conforming to specifications or not, statistical tools, viz. inspection plans, control charts, etc., are of extreme importance.

Statistics and Mathematics

Statistics and mathematics arc very intimately related. Recent advancements in statistical techniques are the outcome of wide applications of advanced mathematics. Main contributors to statistics, namely, Bernouli, Pascal, Laplace, De-Moirve, Gauss, R. A. Fisher, to mention only a few, were primarily talented and skilled mathematicians. Statistics may be regarded as that branch of mathematics which provided us with systematic methods of analysing a large number of related numerical facts. According to Connor, " Statistics is a branch of Applied Mathematics which specialises in data."

Statistics and Biology, Astronomy and Medical Science

The association between statistical methods and biological theories was first studied byFrancis Galton in his work in Regression. According to Prof. Karl Pearson, the whole 'theory of heredity' rests on statistical basis. He said, "The whole problem of evolution is a problemof vital statistics, a problem of longevity, of fertility, of health, of disease and it is impossible to discuss the national mortality without an enumeration of the population, a classification of deaths and knowledge of statistical theory." In astronomy, the theory of Gaussian 'Normal Law of Errors' for the study of the movement of stars and planets is developed by using the 'Principle of Least Squares'. In medical science also, the statistical tools for the collection, presentation and analysis of observed facts relating to the causes and diseases and the results obtained from the use of various drugs and medicines, are of great importance. Moreover, theefficacy of a manufactured drug or injection or medicine is tested by analysing the 'tests of significance'.

Statistics and Psychology and Education

In education and psychology, too, statistics has found wide applications, e.g., to determine the reliability and validity of a test, 'Factor Analysis', etc., so much so that a new subject called 'Psychometry' has come into existence.

Statistics and War

In war, the theory of 'Decision Functions' can be of great assistance to military and technical personnel to plan 'maximum destruction with minimum effort'. Thus, we see that the science of Statistics is associated with almost all the sciences - social as well as physical. Bowley has rightly said, "A knowledge of Statistics is like a knowledge of foreign language or algebra; it may prove of use at any time under any circumstance."

1.2.4 Limitations of Statistics

Statistics, with its wide applications in almost every sphere of human activity; is not without limitations. The following are some of its important limitations:

(i) **Statistics is not suited to the study of qualitative phenomenon.** Statistics, being a science dealing with a set of numerical data, is applicable to the study of only those subjects of enquiry which are capable of quantitative measurement. As such; qualitative phenomena like honesty, poverty, culture, etc., which cannot be expressed numerically, are not capable of direct statistical analysis. However, statistical techniques may be applied indirectly by first reducing the qualitative expressions to precise quantitative terms. For example, the intelligence of a group of candidates can be studied on the basis of their scores in a certain test.

(ii) **Statistics does not study individuals.** Statistics deals with an aggregate of objects and does not give any specific recognition to the individual items of a series. Individual items, taken separately, do not constitute statistical data and are meaningless for any statistical enquiry. For example, the individual figures of agricultural production, industrial output or national income of any country for a particular year are meaningless unless, to facilitate comparison, similar figures of other countries or of the same country for different years are given. Hence,statistical analysis is suited to only those problems where group characteristics areto be studied.

(iii) **Statistical laws are not exact.** Unlike the laws of physical and natural sciences, statistica1laws are only approximations and not exact. On the basis of statistical analysis, we can talk only in terms of probability and chance and not in terms of certainty. Statistical conclusions are not universally true, rather they are true only on an average.

1.2.5 Misuse of Statistics

Statistics is liable to be misused. As they say, "Statistical methods are the most dangerous tools in the hands of the in experts. Statistics is one of those sciences whose adepts must exercise the self-restraint of an artist." The use of statistical tools by inexperienced and untrained persons might lead to very fallacious conclusions. One of the greatest shortcomingsof statistics is that by just looking at them one can‟t comment about their quality and as such can be represented in any manner to support one's way of argument and reasoning. As King said, "Statistics are like clay of which one can make a god or devil as one place." The requirement of experience and judicious use of statistical methods restricts their use toexperts only and limits the chances of the mass popularity of this useful and important science.

It may be pointed out that Statistics neither proves anything nor disproves anything. It is only a tool which if rightly used may prove extremely useful and if misused might be disastrous. According to Bowley, "Statistics only furnishes a tool necessary though imperfect, which is dangerous in the hands of those who do not know its use and its deficiencies." It is not the statistics which can be blamed but those persons who twist the numerical data and misuse them either due to ignorance or deliberately for personal selfish motives. As King pointed out, "Science of Statistics is the most useful servant but only of great value to those who understand its proper use."

A few interesting examples showing the impact of misrepresentation of statistical data are:

(i) A statistical report, "The number of accidents taking place in the middle of the road is much less than the number of accidents taking place on its side. Hence it issafer to walk in the middle of the road." This conclusion is obviously wrong since we are not given the proportion of the number of accidents to the number of persons walking in the two cases.

(ii) Another saying that, "The number of students taking up Computer Science in a University has increased 5 times during the last 3 years. Thus, Computer Science is gaining popularity among the students of the university." Again, the conclusion is faulty since we are not given any such details about the other subjects and hence comparative study is not possible.

(iii) One more interesting examples says that, "99% of the people who drink alcohol die before attaining the age of 100 years. Hence drinking is harmful for longevity of life." This statement, too, is incorrect since nothing is mentioned about the number of persons who do not alcohol and die before attaining the age of 100 years. Thus, statistical arguments based on incomplete data often lead to fallaciousconclusions.

1.2.6 Types of Data

In statistics, the data are the individual pieces of factual information recorded, and it is used for the purpose of the analysis process. The two processes of data analysis are interpretation and presentation. Statistics are the result of data analysis. Data classification and data handling are an important process as it involves a multitude of tags and labels to define the data, its integrity and confidentiality. The data can be classified as shown in figure 1.1 and has been described as follows:

Qualitative or Categorical Data

Qualitative data, also known as the categorical data, describes the data that fits into the categories. Qualitative data are not numerical. The categorical information involves categorical variables that describe the features such as a person‟s gender, home town etc. Categorical measures are defined in terms of natural language specifications, but not in terms of numbers.

Sometimes categorical data can hold numerical values (quantitative value), but those values do not have mathematical sense. Examples of the categorical data are birthdate, favourite sport, school postcode. Here, the birthdate and school postcode hold the quantitative value, but it does not give numerical meaning. It can be further classified as nominal and ordinal data.

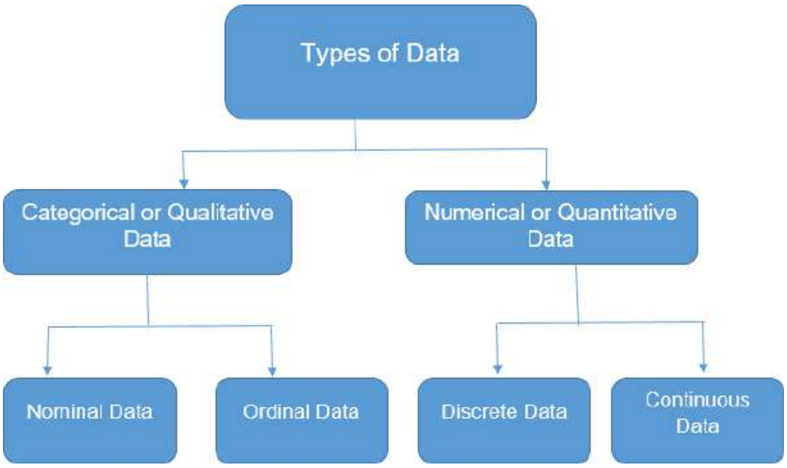


Figure 1.1: Classification of Data used in Statistics

Nominal Data: Nominal data is one of the types of qualitative information which helps to label the variables without providing the numerical value. Nominal data is also called the nominal scale. It cannot be ordered and measured. But sometimes, the data can be qualitative and quantitative. Examples of nominal data are letters, symbols, words, gender etc.

The nominal data are examined using the grouping method. In this method, the data are grouped into categories, and then the frequency or the percentage of the data can be calculated. These data are visually represented using the pie charts.

Ordinal Data: Ordinal data is a type of data which follows a natural order. The significant feature of the nominal data is that the difference between the data values is not determined. This variable is mostly found in surveys, finance, economics, questionnaires, and so on.

The ordinal data is commonly represented using a bar chart. These data are investigated and interpreted through many visualisation tools. The information may be expressed using tables in which each row in the table shows the distinct category.

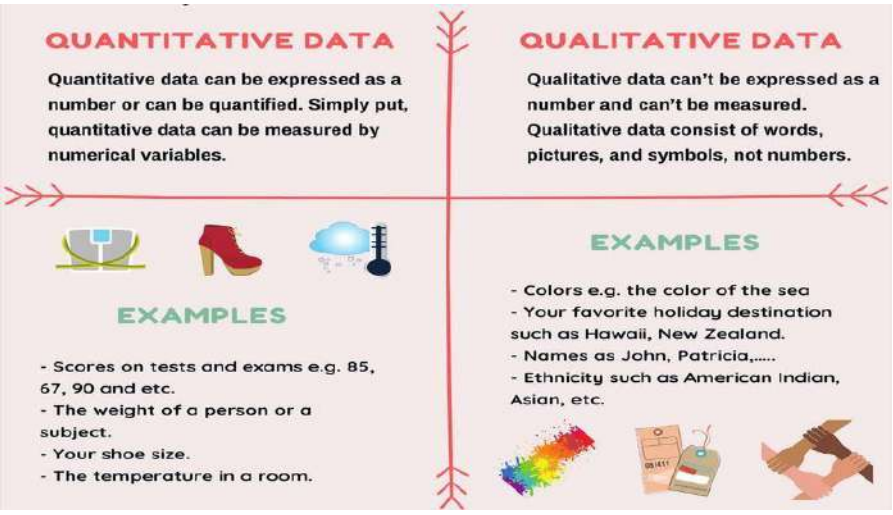
Quantitative or Numerical Data

Quantitative data is also known as numerical data which represents the numerical value (i.e., how much, how often, how many). Numerical data gives information about the quantities of a specific thing. Some examples of numerical data are height, length, size, weight, and so on. The quantitative data can be classified into two different types based on the data sets. The two different classifications of numerical data are discrete data and continuous data.

Discrete Data: Discrete data can take only discrete values. Discrete information contains only a finite number of possible values. Those values cannot be subdivided meaningfully. Here, things can be counted in the whole numbers e.g. Number of students in the class

Continuous Data: Continuous data is data that can be calculated. It has an infinite number of probable values that can be selected within a given specific range e.g. Temperature range.

The quantitative and qualitative data can be represented as in figure 1.2.



QUANTITATIVE DATA

Quantitative data can be expressed as a number or can be quantified. Simply put, quantitative data can be measured by numerical variables.

EXAMPLES

- Scores on tests and exams e.g. 85, 67, 90 and etc.
- The weight of a person or a subject.
- Your shoe size.
- The temperature in a room.

QUALITATIVE DATA

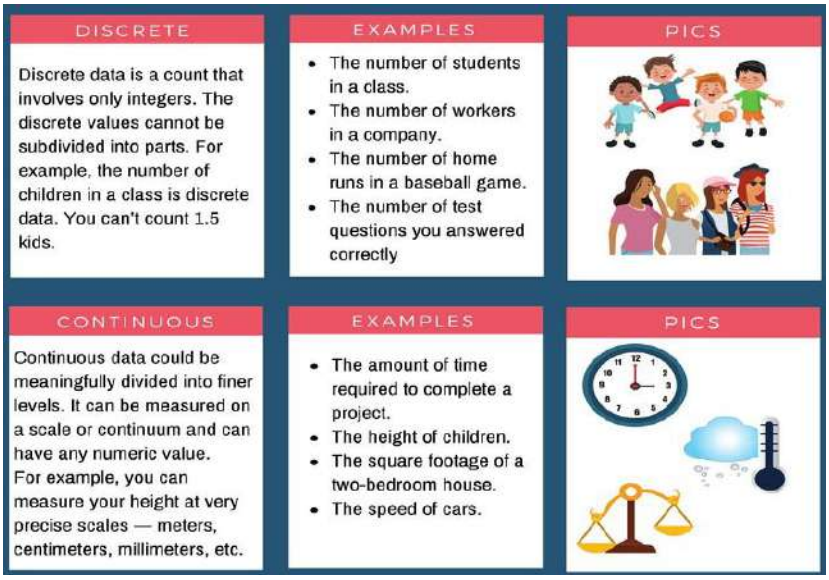
Qualitative data can't be expressed as a number and can't be measured. Qualitative data consist of words, pictures, and symbols, not numbers.

EXAMPLES

- Colors e.g. the color of the sea
- Your favorite holiday destination such as Hawaii, New Zealand.
- Names as John, Patricia,.....
- Ethnicity such as American Indian, Asian, etc.

Figure 1.2: Quantitative and Qualitative Data

Figure 1.3 shows the types of qualitative data i.e. discrete and continuous data.



DISCRETE

Discrete data is a count that involves only integers. The discrete values cannot be subdivided into parts. For example, the number of children in a class is discrete data. You can't count 1.5 kids.

EXAMPLES

- The number of students in a class.
- The number of workers in a company.
- The number of home runs in a baseball game.
- The number of test questions you answered correctly

PICS

CONTINUOUS

Continuous data could be meaningfully divided into finer levels. It can be measured on a scale or continuum and can have any numeric value. For example, you can measure your height at very precise scales — meters, centimeters, millimeters, etc.

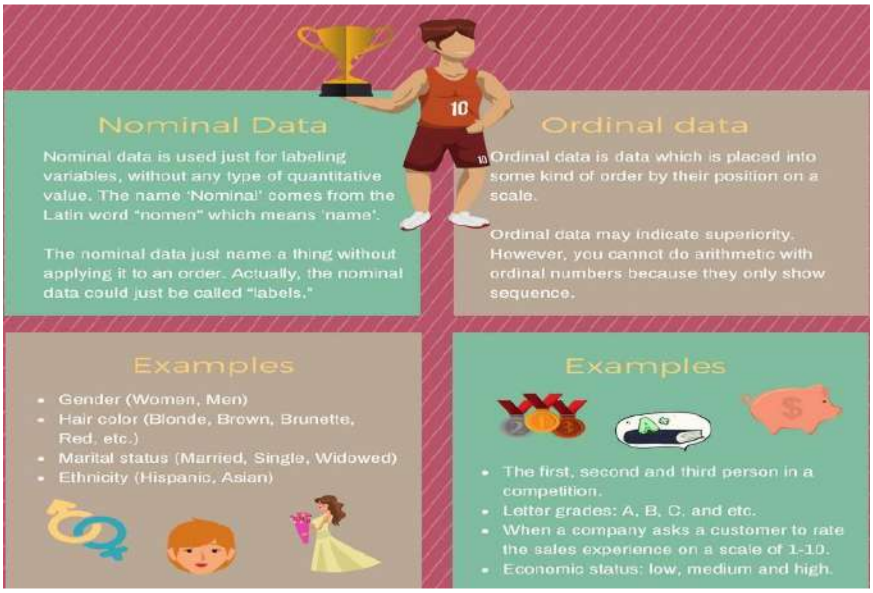
EXAMPLES

- The amount of time required to complete a project.
- The height of children.
- The square footage of a two-bedroom house.
- The speed of cars.

PICS

Figure 1.3: Types of Qualitative Data viz., Discrete and Continuous

Figure 1.4 shows the types of quantitative data i.e. nominal and ordinal data.



Nominal Data

Nominal data is used just for labeling variables, without any type of quantitative value. The name 'Nominal' comes from the Latin word "nomen" which means 'name'.

The nominal data just name a thing without applying it to an order. Actually, the nominal data could just be called "labels."

Ordinal data

Ordinal data is data which is placed into some kind of order by their position on a scale.

Ordinal data may indicate superiority. However, you cannot do arithmetic with ordinal numbers because they only show sequence.

Examples

- Gender (Women, Men)
- Hair color (Blonde, Brown, Brunette, Red, etc.)
- Marital status (Married, Single, Widowed)
- Ethnicity (Hispanic, Asian)

Examples

- The first, second and third person in a competition.
- Letter grades: A, B, C, and etc.
- When a company asks a customer to rate the sales experience on a scale of 1-10.
- Economic status: low, medium and high.

Figure 1.4: Types of Quantitative Data viz., Nominal and Ordinal

1.2.7 Data Collection

Depending on the source, it can be classified as primary data or secondary data. Let us take a look at them both.

Primary Data

These are the data that are collected directly by an investigator for a specific purpose. Primary data are „pure‟ in the sense that no statistical operations have been performed on them and they are original. An example of primary data is the Census of India.

Secondary Data

They are the data that are sourced from someplace that has originally collected it.This means that this kind of data has already been collected by some researchers or investigators in the past and is available either in published or unpublished form. Thisinformation is impure as statistical operations may have been performed on them already. An example is an information available on the Government of India, the Department of Finance‟s website or in other repositories, books, journals, etc.

1.3 SUMMARY

In this module, the overall development and history of statistics has been discussed. Various definitions given be various authors have been provided and discussed. Following this, the applications along with advantages and limitations of statistics have also been discussed in detail. One of the very important aspect, misuse of statistics has also been explored along with few examples for misuse of the statistics. Then the various classifications of statistical data have also been discussed in detail. In the end, the types of data collection methods have been discussed in brief. Overall, this module provides an overview of what is statistics along with its applications in depth.

1.4 QUESTIONS FOR PRACTICE

- Give a historical background of statistics.
- Write various definitions of statistics and discuss these definitions in brief.
- State and explain various applications of statistics.
- What are the various limitations of statistics?
- “Statistics don‟t lie”. Comment on this statement.
- Provide a few examples which can lead to incorrect conclusion due to wrong analysis of statistics.
- Give any two examples of collecting data from day-to-day life.
- How can you classify the statistical data?
- Categorize the following data in various types: (i) Speed (ii) Gender (iii) Height (iv) Grades (v) No. of Employees (vi) Time (vii) Colour (viii) Score (ix) Weight.
- The word statistics seems to have been derived from which word?
- From__________it is known that even before 300 B.C. a very good system of collecting “Vital Statistics” and registration of births and deaths was in vogue____ .
- Who is known as the father of “Vital Statistics”?

REFERENCES

- A. Abebe, J. Daniels, J.W.Mckean, “Statistics and Data Analysis”.
- Clarke, G.M. & Cooke, D., “A Basic course in Statistics”, Arnold.
- David M. Lane, “Introduction to Statistics”.
- S.C.Gupta and V.K.Kapoor, “Fundamentals of Mathematical Statistics”, Sultan Chand & Sons, New Delhi.

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

UNIT II: PRESENTATION OF DATA BY TABLES

STRUCTURE

2.0 OBJECTIVES

In this module, we will try to understand about the representation of data through tables. The various types of distribution of data and tabular presentation will be discussed in this module. Further, graphical representation of data through some popular methods such as histogram and frequency polygons will also be discussed in detail. Thereafter, the classification and graphical representation of statistical data through various methods will be explained. Further, we shall also learn the various types of representations (both tabular and graphical) of data in Python.

2.1 INTRODUCTION

This module is designed to know about the representation of data in tabular and graphical forms. For analysing the statistical data, it must be represented in a tabular form and this module does the same, i.e., describe the techniques to convert the data in tabular forms. For the purpose of planning and interpreting the data, visual effects are very useful and necessary.The visual effects in statistics can be obtained by representing the data through graphs. In thismodule, various types of graphs, viz., histogram, frequency polygons, bar graphs, pie charts, ogives, etc. have been discussed. Further, an automated tool for representing such type ofdata must be used for quick analysis and better representation. One such tool Python is usedin this chapter to represent the data.

2.2 MAIN CONTENT

2.2.1 Presentation of Data by Tables

Whenever we want to analyse and interpret the data, it can be done in an effective manner only when it is represented in tabular and/or graphical manner. The data in tabular form can be represented by using frequency distributions as explained in the following section.

2.2.2 Construction of Frequency Distribution

Frequency Distributions

When observations, discrete or continuous, are available on a single characteristic of a large number of individuals, often it becomes necessary to condense the data as far as possible without losing any information of interest. For condensing the data, it is represented using either discrete or continuous frequency distribution tables.

Discrete frequency distribution: In discrete frequency distribution, values of the variable is arranged individually. The frequencies of the various values are the number of times each value occurs. For examples, the weekly wages paid to the workers are given below.

300, 240, 240, 150, 120, 240, 120, 120, 150, 150, 150, 240, 150, 150, 120, 300, 120, 150, 240, 150, 150, 120, 240, 150, 240, 150, 120, 120, 240, 150.

There are various ways to form a frequency distribution for this data. In the first case, let us assume that data is represented in terms of tally marks in a tabular manner as shown below in the table 2.1:

Table 2.1: Representation of Data using Tally Marks

Weekly Wages	Tally Marks	No. of Workers
120	~~IIII~~ III	8
150	~~IIII~~ ~~IIII~~ II	12
240	~~IIII~~ III	8
300	II	2

This data can also be represented without using tally marks i.e. using frequency only as shown in table 2.2 and is known as frequency table.

Table 2.2: Frequency Table for the Data in Table 2.1

Weekly Wages (x)	120	150	240	300	Total
No. of Workers (f)	8	12	8	2	30

The frequency table 2.1 is ungrouped frequency table. We can also draw a grouped frequency table depending on the data we are having. For designing a grouped frequency table, let us consider the following example regarding daily maximum temperatures in in a city for 50 days.

28, 28, 31, 29, 35, 33, 28, 31, 34, 29, 25, 27, 29, 33, 30, 31, 32, 26, 26, 21, 21, 20, 22, 24, 28, 30, 34, 33, 35, 29, 23, 21, 20, 19, 19, 18, 19, 17, 20, 19, 18, 18, 19, 27, 17, 18, 20, 21, 18, 19.

Table 2.3: Grouped Frequency Table

Temperature	Frequency
17-21	17
22-26	9
27-31	13
32-36	11
Total	50

The classes of type 17-21 and 22-26 are inclusive in nature i.e. both the lower bound and upper bound are included in the limit.

Although there are no hard and fast rules that have been laid down for it The following points may be kept in mind for classification:

(i) The classes should be clearly defined and should not lead to ambiguity.

(ii) The classes should be exhaustive, i.e., each of the given values should be included in one of the classes.

(iii) The classes should be mutually exclusive and non-overlapping.

(iv) The classes should be of equal width. The principle, however, cannot be rigidly followed.

(v) Indeterminate classes, e.g., the open-end classes such as less than 'a' or greater than 'b' should be avoided as far as possible since they create difficulty in analysis and interpretation.

(vi) The number of classes should neither be too large nor too small. It should preferably lie between 5 and 15. However. the number of classes may be more than 15 depending upon the total frequency and the details required. But it is desirable that it is not less than 5 since in that case the classification may not reveal the essential characteristics of the population.

Terms used in frequency distribution

Class Interval: The whole range of variable values is classified in some groups in the form of intervals. Each interval is called a class interval.

Class Frequency: The number of observations in a class is termed as the frequency of the class or class frequency.

Class limits and Class boundaries: Class limits are the two endpoints of a class interval which are used for the construction of a frequency distribution. The lowest value of the variable that can be included in a class interval is called the lower class limit of that class interval. The highest value of the variable that can be included in a class interval is called the upper-class limit of that class interval. In the table 2.3, the class intervals are 17-21, 22-26, 27-31 and 32-36. Here, say for the class 17-21, the lower-class limit is 17 and the upper-class limit is 21. Both 17 and 21 are part of this class. This is called inclusive class. Another typeof class is exclusive class as shown below in table 2.4:

Table 2.4: Exclusive Class Grouped Frequency Table

Temperature	Frequency
17-21	17
21-25	7
25-29	10
29-33	9
33-37	7
Total	50

In table 2.4 upper values are excluded from the class i.e., in the class 17-21 only values from 17 to 20 are taken and the values of 21 in considered in the next class. Such type of distribution is known as exclusive class.

Open-end classes: It may be the case that some values in the data set are extremely small compared to the other values of the data set and similarly some values are extremely large in comparison. Then what we do is we do not use the lower limit of the first class and the upper limit of the last class. Such classes are called open end classes.

Table 2.5: Open-end Class Grouped Frequency Table

Temperature	Frequency
Below 21	17
21-25	7
25-29	10
29-33	9
Above 33	7
Total	50

Size of the Class: The length of the class is called the class width. It is also known as class size.

Class interval or size of the class = Upper Limit – Lower Limit

Mid-point of the Class: The midpoint of a class interval is called Mid-point of the Class. It is the representative value of the entire class.

Mid-point of the class = (Upper Limit + Lower Limit) / 2

Continuous Frequency Distribution: If we deal with a continuous variable, it is not possible to arrange the data in the class intervals of above type. Let us consider the distribution of age in years. If class intervals are 15-19, 20-24 then the persons with ages between 19 and 20 years are not taken into consideration. In such a case we form the class intervals as shown below in table 2.6.

Table 2.6: Continuous Data

Age(in Years)
Below 5
5 or more but less than 10
10 or more but less than 15
15 or more but less than 20
20 or more but less than 25
…

As all cases have been covered in this table. But it is difficult to perform calculations using this table, therefore data is represented as in the table 2.7.

Table 2.7: Continuous Data using Classes

Age(in Years)
0-5
5-10
10-15
15-20
20-25
25-30

This form of frequency distribution is known as continuous frequency distribution. It should be clearly understood that in the above classes, the upper limits of each class are excluded from the respective classes. Such classes in which the upper limits are excluded from the respective classes and are included in the immediate next class are known as 'exclusive classes' and the classification is termed as 'exclusive type classification'.

2.2.3 Graphical Representation of a Frequency Distribution

It is often useful to represent a frequency distribution by means of a diagram which makes the data easily understandable and conveys the general information about the data. Diagrammatic representation also facilitates the comparison of two or more frequency distributions.

Graphs are charts consisting of points, lines and curves. Charts are drawn on graph sheets. Scales are to be chosen suitably in both X and Y axes so that entire data can be presented in the graph sheet. Statistical measures such as quartiles, median and mode can be found from the appropriate graph. Graphs are useful for analysis of time series, regression analysis, business forecasting, interpolation, extrapolation, etc.

Types of graphs

Graphs in statistics are broadly divided into two categories.

i) Graphs of time series or Historigrams
ii) Graphs of frequency distribution

Graphs of time series or Historigrams: A historigram is a graph to show a time series. It shows the fluctuation of a variable over a given period. X axis is used to denote the time and Y axis the value of the variable. Each pair of (time, variable) is denoted by a point on the graph. After plotting all such points, successive points are joined by straight lines. The resulting curve is historigram.

For example, let us draw a historigram (as in figure 2.1) to show the population in various census years with the given data as in table 2.8.

Table 2.8: Population in Various Census Years

Census Year	1951	1961	1972	1981	1998	2017
Population(in Million)	33.44	42.88	65.31	83.78	130.58	200.17

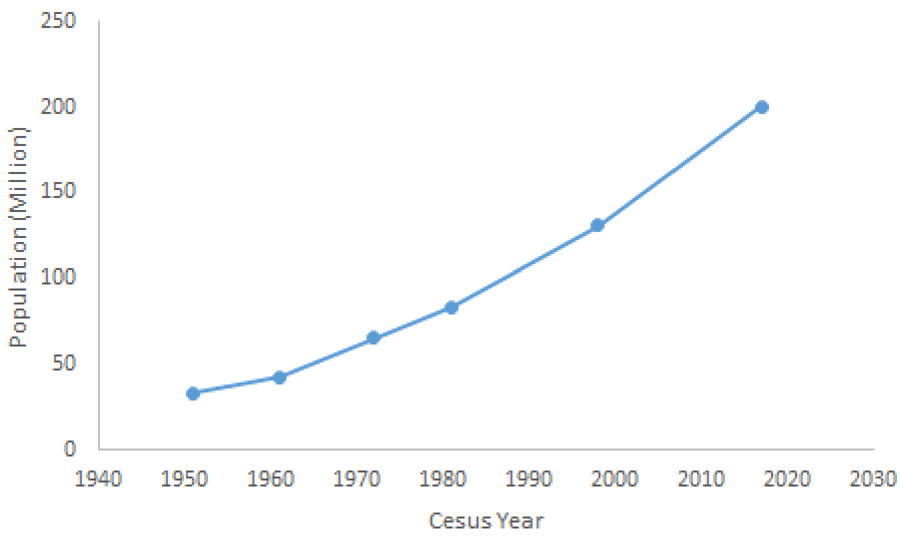


Figure 2.1: Historigram for data in table 2.8

Graphs of frequency distribution: There are various types of graphs of frequency distribution such as:

a) Histogram

b) Frequency polygon used to present continuous frequency distribution

c) Frequency curve

d) Ogive curve used to represent cumulative frequency distribution

e) Pie chart used to represent relative frequency.

f) Bar Diagram used to compare the frequencies.

HISTOGRAM: In drawing the histogram of a given continuous frequency distribution we first mark off along the x-axis all the class intervals on a suitable scale. On each class interval rectangles are drawn with heights proportional to the frequency of the corresponding class interval. The diagram of continuous rectangles so obtained is called histogram.

For examples, the table 2.9 gives the life times of 400 bulbs.

Table 2.9: Lifetime of Bulbs

Lifetime (in hours)	Number of bulbs
300 – 400	14
400 – 500	56
500 – 600	60
600 – 700	86
700 – 800	74
800 – 900	62
900 – 1000	48

The histogram for table 2.9 is:

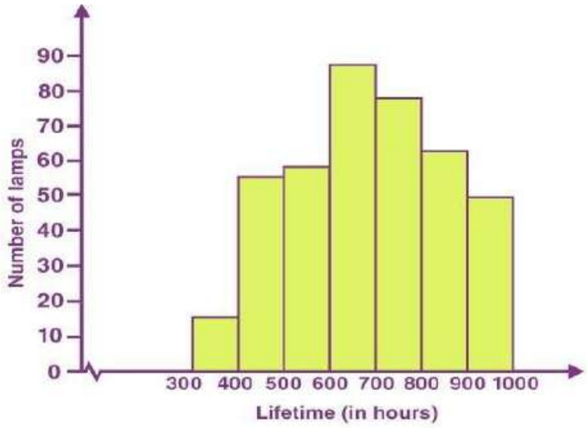


Figure 2.2: Histogram for data in table 2.9

Histogram in Python: It is important to represent the given data using histogram. But it is more important to represent it using a tool. In this course, we will be using Python as a tool for representing any data. From this point onwards, we will learn the various representations using Python too.

Creating Numpy Histogram: Numpy has a built-in numpy.histogram() function which represents the frequency of data distribution in the graphical form. The rectangles having equal horizontal size corresponds to class interval called bin and variable heightcorresponding to the frequency. It can be created using following statement:

numpy.histogram(data, bins=10, range=None, normed=None, weights=None,density=None).

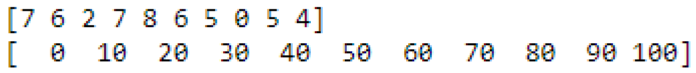
Where,

Attribute	*Parameter*
data	*array or sequence of array to be plotted*
bins	*int or sequence of str defines number of equal width bins in a range, default is 10*
range	*optional parameter sets lower and upper range of bins*
normed	*optional parameter same as density attribute, gives incorrect result for unequal bin width*
weights	*optional parameter defines array of weights having same dimensions as data*
density	*optional parameter if False result contain number of sample in each bin, if True result contain probability density function at bin*
data	*array or sequence of array to be plotted*
bins	*int or sequence of str defines number of equal width bins in a range, default is 10*
range	*optional parameter sets lower and upper range of bins*
normed	*optional parameter same as density attribute, gives incorrect result for unequal bin width*
weights	*optional parameter defines array of weights having same dimensions as data*
density	*optional parameter if False result contain number of sample in each bin, if True result contain probability density function at bin*

The creation of Numpy histogram can be better understood by the following programs:

```
# Program 2.1: Histogram Numeric Representation
# Import libraries
import numpy as np
# Creating dataset
a = np.random.randint(100, size =(50))
```

```
# Creating histogram
np.histogram(a, bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
hist, bins = np.histogram(a, bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
# printing histogram
print  (hist)
print  (bins)
Output:
```



```
[7 6 2 7 8 6 5 0 5 4]
[  0  10  20  30  40  50  60  70  80  90 100]
```

The above numeric representation of histogram can be converted into a graphical form. The plt() function present in pyplot submodule of Matplotlib takes the array of dataset and array of bin as parameter and creates a histogram of the corresponding data values.

```
# Program 2.2: Histogram
# import libraries
from matplotlib import pyplot as plt
import numpy as np
# Creating dataset
a = np.random.randint(100, size =(50))
# Creating plot
fig = plt.figure(figsize =(10, 7))
plt.hist(a, bins = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100])
plt.title("Numpy  Histogram")
# show plot
plt.show()
```

Output:

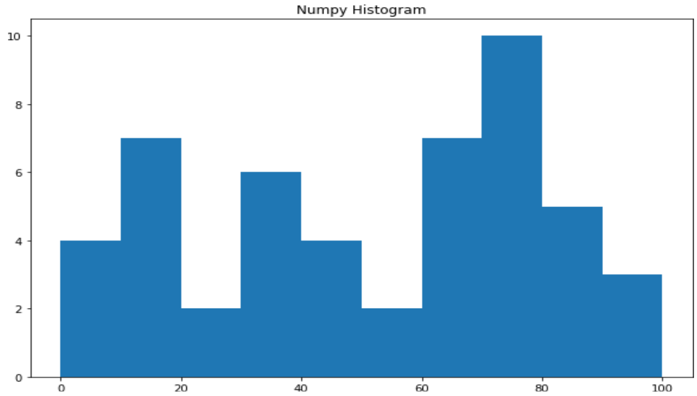


Figure 2.3: Histogram using Program 2.2

Creating Histogram using Matplotlib: To create a histogram the first step is to create bin of the ranges, then distribute the whole range of the values into a series of intervals, and the count the values which fall into each of the intervals. Bins are clearly identified as consecutive, non-overlapping intervals of variables. The matplotlib.pyplot.hist() function is used to compute and create histogram of x.

The following table shows the parameters accepted by matplotlib.pyplot.hist() function :

Attribute	*parameter*
X	*array or sequence of array*
bins	*optional parameter contains integer or sequence or strings*
density	*optional parameter contains boolean values*
range	*optional parameter represents upper and lower range of bins*
histtype	*optional parameter used to creae type of histogram [bar, barstacked, step, stepfilled], default is "bar"*
align	*optional parameter controls the plotting of histogram [left, right, mid]*
weights	*optional parameter contains array of weights having same dimensions as x*
bottom	*location of the baseline of each bin*
rwidth	*optional parameter which is relative width of the bars with respect to bin width*

Attribute	*parameter*
color	*optional parameter used to set color or sequence of color specs*
label	*optional parameter string or sequence of string to match with multiple datasets*
Log	*optional parameter used to set histogram axis on log scale*

Frequency Polygon and Curves: For an ungrouped distribution, the frequency polygon is obtained by plotting points with corresponding frequencies and joining the plotted points by means of straight lines. For a grouped frequency distribution, the points are mid-values of the class intervals. For equal class intervals the frequency polygon can be obtained by joining the middle Points of the upper sides of the adjacent rectangles of the histogram by means of straight lines. If the class intervals are of small width the polygon can be approximated by a smooth curve. The frequency curve can be obtained by drawing a smooth freehand curve through the vertices of the frequency polygon.

For example, let us present the following data given for **a batch of 400 students, the height of students is provided in the table 2.10, using frequency polygon.**

Table 2.10: Heights of Students Data

Height (In cm)	Number of Students
140-150	74
150-160	163
160-170	135
170-180	28
Total	400

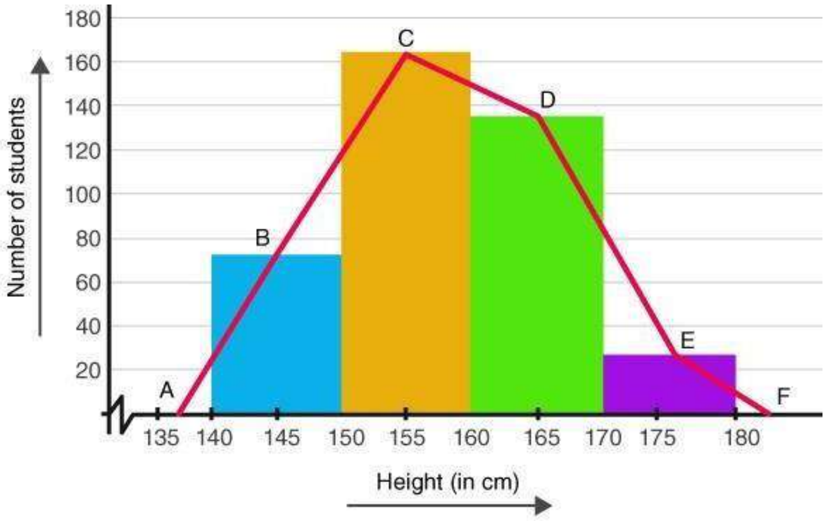


Figure 2.4: Frequency Polygon for the data in table 2.10

ABCDEF represents the given data graphically in form of frequency polygon as shown above in figure 2.4.

Now let us consider an example to draw a frequency curve.

Table 2.11: Data for Frequency Curve

Seed Yield (gms)	No. of Plants
2.5-3.5	4
3.5-4.5	6
4.5-5.5	10
5.5-6.5	26
6.5-7.5	24
7.5-8.5	15
8.5-9.5	10
9.5-10.5	5

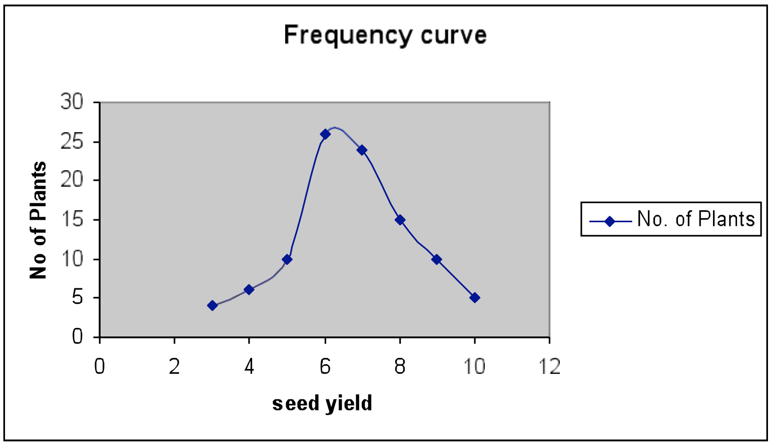


Figure 2.5: Frequency Curve for data in table 2.11

Drawing a frequency polygon in Python may be understood from the following example.

Suppose you have only the angle values for a set of data. Now you need to plot an angle distribution curve i.e., angle on the x axis v/s no. of times/frequency of angle occurring on the y axis. These are the angles sorted out for a set of data: -

[98.1706427, 99.09896751, 99.10879006, 100.47518838, 101.22770381, 101.70374296, 103.15715294, 104.4653976,105.50441485, 106.82885361, 107.4605319, 108.93228646, 111.22463712, 112.23658018, 113.31223886, 113.4000603, 114.14565594, 114.79809084, 115.15788861, 115.42991416, 115.66216071, 115.69821092, 116.56319054, 117.09232139, 119.30835385, 119.31377834, 125.88278338, 127.80937901, 132.16187185, 132.61262906, 136.6751744, 138.34164387,]

The data can easily be represented using Python with the help of following code:

```
# Program 2.3: Frequency Polygon
from matplotlib import pyplot as plt
import numpy as np
angles = [98.1706427, 99.09896751, 99.10879006, 100.47518838, 101.22770381,
101.70374296, 103.15715294, 104.4653976, 105.50441485, 106.82885361, 107.4605319,
108.93228646, 111.22463712, 112.23658018, 113.31223886, 113.4000603, 114.14565594,
114.79809084, 115.15788861, 115.42991416, 115.66216071, 115.69821092, 116.56319054,
117.09232139, 119.30835385, 119.31377834, 125.88278338, 127.80937901, 132.16187185,
132.61262906, 136.6751744, 138.34164387, ]
hist,edges = np.histogram(angles, bins=20)
bin_centers = 0.5*(edges[:-1] + edges[1:])
bin_widths = (edges[1:]-edges[:-1])
plt.bar(bin_centers,hist,width=bin_widths)
plt.plot(bin_centers, hist,'r')
plt.xlabel('angle [$^\circ$]')
plt.ylabel('frequency')
plt.show()
```

Output:

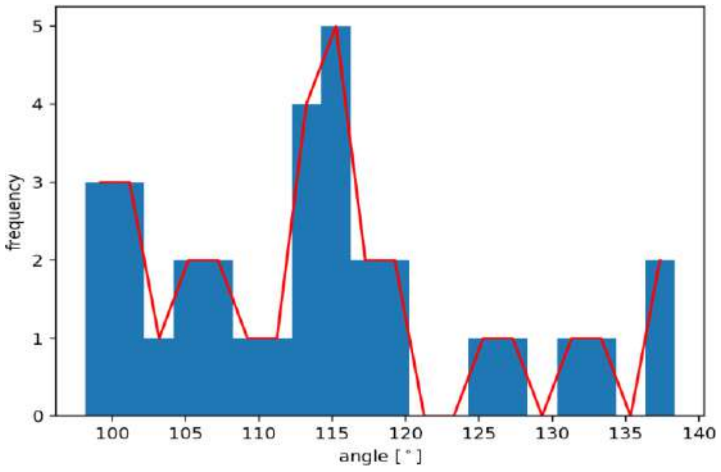


Figure 2.6: Frequency Polygon

2.2.4 Cumulative Frequency Distribution

Cumulative frequency is defined as a running total of frequencies. The frequency of an element in a set refers to how many of that element there are in the set. Cumulative frequencycan also be defined as the sum of all previous frequencies up to the current point.

Consider an example which shows the ages of participants in a certain class. We need to draw a cumulative frequency table for the data given in table 2.12.

Table 2.12: Frequency Table

Age	Frequency
10	3
11	18
12	13
13	12
14	7
15	27

The cumulative frequency table for the above data can be drawn as table 2.13. In this frequencies are the sum of the current frequency and previous frequencies. In other words, we can say that cumulative frequency shows the number of participants under or equal to the age of 10, 11, 12, 13, 14 and 15 respectively.

Table 2.13: Cumulative Frequency Table

Age	Frequency	Cumulative Frequency
10	3	3
11	18	18+3=21
12	13	21+13=34
13	12	34+12=46
14	7	46+7=53
15	27	53+27=80

However, there are two kinds of cumulative frequency distribution.

i) Less than cumulative frequency distribution

ii) More than cumulative frequency distribution

Less than cumulative frequency distribution: Frequency distribution both discrete and continuous are to be taken in ascending order. The total of the frequencies from the beginning up to and including each frequency is found. That cumulative frequency shows how many items are less than or equal to the corresponding value of the class interval.

More than cumulative frequency distribution: Frequency distribution both discrete and continuous are to be taken in ascending order. The total of the frequencies from the end up to and including each frequency is found. That cumulative frequency shows how many items are more than or equal to the corresponding value of the class interval.

Consider an example for both these types of cumulative frequencies for ungrouped data using the following table 2.14.

Table 2.14: Less than and More than Cumulative Frequency Curve

Weekly Wages (X)	Number of Workers (F)	Less than Cumulative Frequency	More Than Cumulative Frequency
120	8	8	30
150	12	20	22
240	8	28	10
300	2	30	2

Consider another example for cumulative frequencies using grouped data as shown in the following table 2.15.

Table 2.15: Less than and More than Cumulative Frequency Curve for Grouped Data

Marks (x)	No. of Students (f)	Marks below	No. of students	Marks above	No. of students
		Upper limit	Less than Cumulative Frequency (C.F.)	Lower limit	More than Cumulative Frequency (C.F.)
0-20	2	20	2	0	40
20-40	7	40	9	20	38
40-60	15	60	24	40	31
60-80	9	80	33	60	16
80-100	7	100	40	80	7
Total	40				

Ogive Curve for Cumulative Frequency

Let us now draw Ogive curve for both less than and greater(more) than using an example. Suppose we are given with weekly wages of various workers as shown in the table 2.16:

Table 2.16: Grouped Data

Weekly Wages (x)	No. of Workers (f)
0-20	41
20-40	51
40-60	64
60-80	38
80-100	7

First let us convert this table into less than c.f. and more than c.f.. **Table 2.17: Less than and More than Cumulative Frequency Curve for data in table 2.16**

Weekly Wages (x)	No. of Workers (f)	C.F.(Less than)	C.F.(More than)
0-20	41	41	201
20-40	51	92	160
40-60	64	156	109
60-80	38	194	45
80-100	7	201	7

Less than ogive: Upper limits of class intervals are marked on the x-axis and less than type cumulative frequencies are taken on y-axis. For drawing less than type curve, points (20, 41), (40, 92), (60, 156), (80, 194), (100, 201) are plotted on the graph paper and these are joined by free hand to obtain the less than ogive.

Greater than ogive: Lower limits of class interval are marked on x-axis and greater than type cumulative frequencies are taken on y-axis. For drawing greater than type curve, points (0, 201), (20, 160), (40, 109), (60, 45) and (80, 7) are plotted on the graph paper and these are joined by free hand to obtain the greater than type ogive.

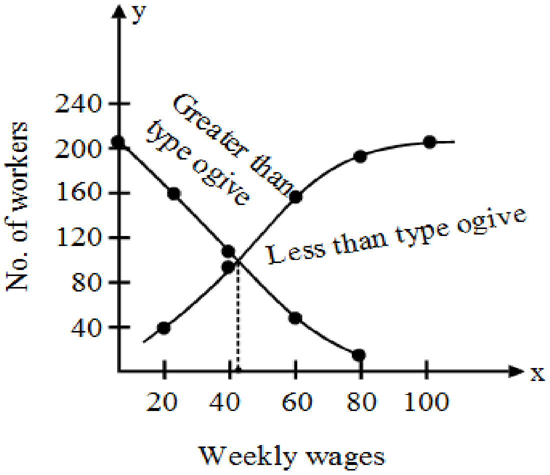


Figure 2.7: Less than and Greater than Ogive

Drawing Ogive in Python

We can draw both types of ogives in python. Let us understand by taking suitable examples. First we consider more than ogive. The more than ogive graph shows the number of values greater than the class intervals. The resultant graph shows the number of values in between the class interval, e.g., 0-10,10-20 and so on. Let us take a dataset, and we will now plot it‟s more than ogive graph- [22,87,5,43,56,73, 55,54,11,20,51,5,79,31,27]. For this data the table 2.18 can be created as follows:

Table 2.18: Data for drawing more than Ogive

Class-Interval (x)	Frequency (f)	Cumulative Frequency (Less than)
0-10	2	2
10-20	1	3
20-30	3	6
30-40	1	7
40-50	1	8
50-60	4	12
60-70	0	12
70-80	2	14
80-90	1	15

Approach for drawing the ogive follows three steps:

(i) Import the modules (matplotlib and numpy)

(ii) Calculate the frequency and cumulative frequency of the data.

(iii) Plot it using the plot() function.

```
# Program 2.4: More than Ogive
# importing modules
import numpy as np
import matplotlib.pyplot as plt
# creating dataset
data = [22, 87, 5, 43, 56, 73, 55, 54, 11, 20, 51, 5, 79, 31, 27]
# creating class interval
classInterval = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
# calculating frequency and class interval
values, base = np.histogram(data, bins=classInterval)
# calculating cumulative sum
cumsum = np.cumsum(values)
# plotting the ogive graph
plt.plot(base[1:], cumsum, color='red', marker='o', linestyle='-')
plt.title('Ogive Graph')
plt.xlabel('Marks in End-Term')
plt.ylabel('Cumulative Frequency')
```

Output:

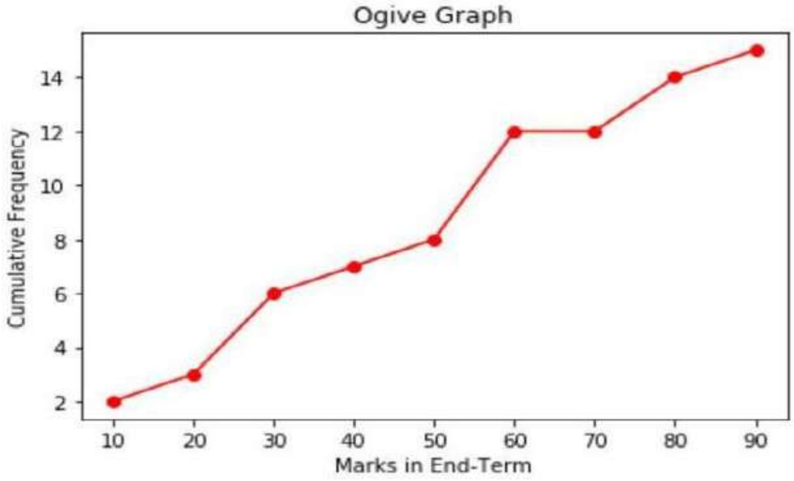


Figure 2.8: More than Ogive

Similarly, we can draw the less than ogive using following example.

In this example, we will plot less than Ogive graph which will show the less than values of class intervals. Dataset: [44,27,5,2,43,56,77,53,89,54,11,23, 51,5,79,25,39]. For this data the table can be created as follows:

Table 2.19: Data for drawing less than Ogive

Class-Interval (x)	Frequency (f)	Cumulative Frequency (More than)
0-10	3	17
10-20	1	14
20-30	3	13
30-40	1	10
40-50	2	9
50-60	4	7
60-70	0	3
70-80	2	3
80-90	1	1

Approach is same as above only the cumulative sum that we will calculate will be reversed using **flipud()** function present in the numpy library.

```
# Program 2.5: Less than Ogive
# importing modules
import numpy as np
import matplotlib.pyplot as plt
# creating dataset
data = [44, 27, 5, 2, 43, 56, 77, 53, 89, 54, 11, 23, 51, 5, 79, 25, 39]
# creating class interval
classInterval = [0, 10, 20, 30, 40, 50, 60, 70, 80, 90]
# calculating frequency and intervals
values, base = np.histogram(data, bins=classInterval)
# calculating cumulative frequency
cumsum = np.cumsum(values)
# reversing cumulative frequency
res = np.flipud(cumsum)
# plotting ogive
plt.plot(base[1:], res, color='brown', marker='o', linestyle='-')
plt.title('Ogive Graph')
plt.xlabel('Marks in End-Term')
plt.ylabel('Cumulative  Frequency')
```

Output:

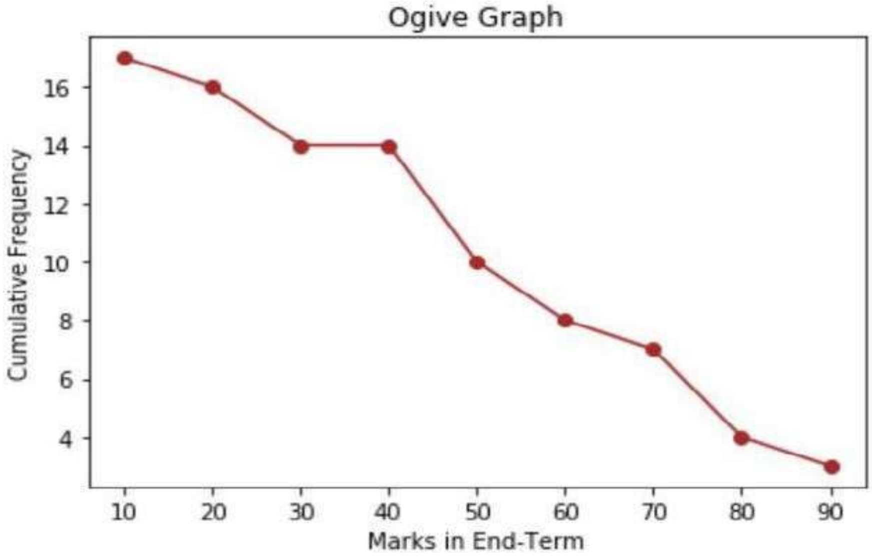


Figure 2.9: Less than Ogive

2.2.5 Pie Charts for Data Representation

A pie chart is a type of graph that represents the data in the circular graph. The slices of pie show the relative size of the data. It is a type of pictorial representation of data. A pie chart requires a list of categorical variables and the numerical variables. Here, the term “pie” represents the whole, and the “slices” represent the parts of the whole. Each slice denotes a proportionate part of the whole. The pie chart is an important type of data representation. It contains different segments and sectors in which each segment and sectors of a pie chart forms a certain portion of the total(percentage). The total of all the data is equal to 360°. **The total value of the pie is always 100%.**

The steps to design a pie chart are:

- Categorize the data
- Calculate the total
- Divide the categories
- Convert into percentages
- Finally, calculate the degrees
- Therefore, the pie chart formula is given as

Formula for pie chart = (Given Data/Total value of Data) × 360°.

Consider an example to draw a pie chart in a step by step manner:

Imagine a teacher surveys her class on the basis of their favourite Sports:

Football	**Hockey**	**Cricket**	**Basketball**	**Badminton**
10	**5**	**5**	**10**	**10**

This can be represented in the following table using above-said steps:

Table 2.20: Data based on favourite sports

Sports	No. of Students	Percentage	Degree(pie)
Football	10	(10/40)*100=25%	(10/40)*360=90^0
Hockey	5	(5/40)*100=12.5%	(5/40)*360=45^0
Cricket	5	(5/40)*100=12.5%	(5/40)*360=45^0
Basketball	10	(10/40)*100=25%	(10/40)*360=90^0
Badminton	10	(10/40)*100=25%	(10/40)*360=90^0
Total	40		

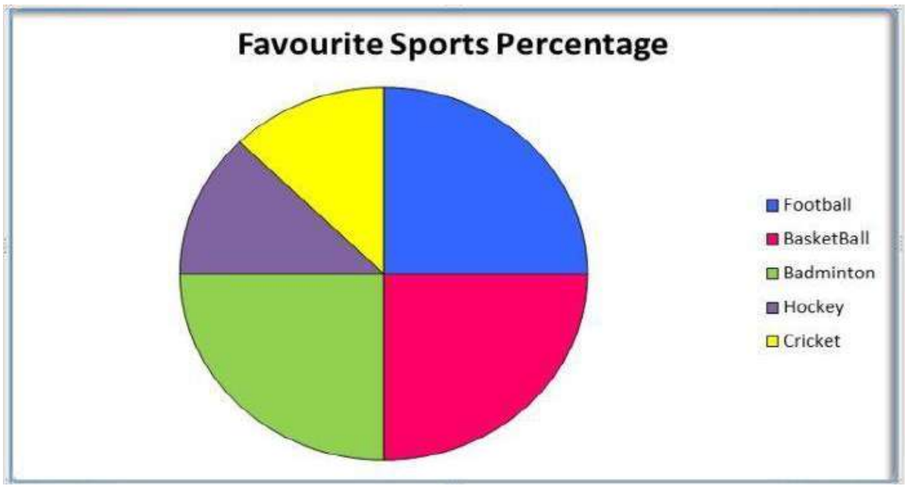


Figure 2.10: Pie Chart for table 2.20

Pie-chart using Python

Matplotlib API has pie() function in its pyplot module which create a pie chart representing the data in an array.

Syntax: matplotlib.pyplot.pie(data, explode=None, labels=None, colors=None, autopct=None, shadow=False)

Where,

data *represents the array of data values to be plotted, the fractional area of each slice is represented by data/sum(data). If sum(data)<1, then the data values returns the fractional area directly, thus resulting pie will have empty wedge of size 1-sum(data).* ***labels*** *is a list of sequence of strings which sets the label of each wedge.****color*** *attribute is used to provide color to the wedges.****autopct*** *is a string used to label the wedge with their numerical value.* ***shadow*** *is used to create shadow of wedge.*

Let‟s create a simple pie chart using the pie() function:

```
# Program 2.6: Pie Chart

# Import libraries

from matplotlib import pyplot as plt
```

```
import numpy as np
# Creating dataset
cars = ['AUDI', 'BMW', 'FORD', 'TESLA', 'JAGUAR', 'MERCEDES']
data = [23, 17, 35, 29, 12, 41]
# Creating plot
fig = plt.figure(figsize =(10, 7))
plt.pie(data, labels = cars)
# show plot
plt.show()
```

Output:

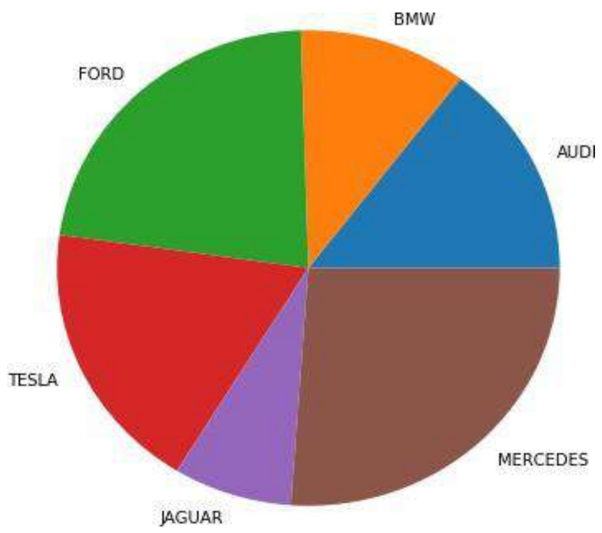


Figure 2.11: Pie Chart

2.2.6 Bar Diagram for Data Representation

Bar graphs are the pictorial representation of data in the form of vertical or horizontal rectangular bars, where the length of bars are proportional to the measure of data. They are also known as bar charts. Bar graphs are one of the means of data handling in statistics.

The bars drawn are of uniform width, and the variable quantity is represented on one of the axes. Also, the measure of the variable is depicted on the other axes. The heights or the lengths of the bars denote the value of the variable, and these graphs are also used to comparecertain quantities. The frequency distribution tables can be easily represented using bar chartswhich simplify the calculations and understanding of data.

The three major attributes of bar graphs are:

- The bar graph helps to compare the different sets of data among different groups easily.
- It shows the relationship using two axes, in which the categories on one axis and the discrete values on the other axis.
- The graph shows the major changes in data over time.

Types of Bar Charts

The bar graphs can be vertical or horizontal. The primary feature of any bar graph is its length or height. If the length of the bar graph is more, then the values are greater than any given data.

Bar graphs normally show categorical and numeric variables arranged in class intervals. They consist of an axis and a series of labelled horizontal or vertical bars. The bars represent frequencies of distinctive values of a variable or commonly the distinct values themselves. The number of values on the x-axis of a bar graph or the y-axis of a column graph is calledthe scale.

The types of bar charts are as follows:

- Vertical bar chart
- Horizontal bar chart

Even though the graph can be plotted using horizontally or vertically, the most usual type of bar graph used is the vertical bar graph. The orientation of the x-axis and y-axis are changed depending on the type of vertical and horizontal bar chart. Apart from the vertical and horizontal bar graph, the two different types of bar charts are:

- Grouped Bar Graph
- Stacked Bar Graph

Now, let us discuss the four different types of bar graphs.

Vertical Bar Graphs: When the grouped data are represented vertically in a graph or chart with the help of bars, where the bars denote the measure of data, such graphs are called vertical bar graphs. The data is represented along the y-axis of the graph, and the height ofthe bars shows the values.

Horizontal Bar Graphs: When the grouped data are represented horizontally in a chart with the help of bars, then such graphs are called horizontal bar graphs, where the bars show the measure of data. The data is depicted here along the x-axis of the graph, and the length of the bars denote the values.

Grouped Bar Graph: The grouped bar graph is also called the clustered bar graph, which is used to represent the discrete value for more than one object that shares the same category. In this type of bar chart, the total number of instances are combined into a single bar. In other words, a grouped bar graph is a type of bar graph in which different sets of data items are compared. Here, a single colour is used to represent the specific series across the set. The grouped bar graph can be represented using both vertical and horizontal bar charts.

Stacked Bar Graph: The stacked bar graph is also called the composite bar chart, which divides the aggregate into different parts. In this type of bar graph, each part can be represented using different colours, which helps to easily identify the different categories.The stacked bar chart requires specific labelling to show the different parts of the bar. In a stacked bar graph, each bar represents the whole and each segment represents the different parts of the whole.

Drawing a Bar Graph: In order to visually represent the data using the bar graph, we need to follow the steps given below.

- First, decide the title of the bar graph.
- Draw the horizontal axis and vertical axis.
- Now, label the horizontal axis.
- Write the names on the horizontal axis.
- Now, label the vertical axis.
- Finalise the scale range for the given data.
- Finally, draw the bar graph.

Bar Graph Examples: To understand the above types of bar graphs, consider the following examples:

In a firm of 400 employees, the percentage of monthly salary saved by each employee is given in the following table. Represent it through a bar graph.

Table 2.21: Data of Savings

Savings (in percentage)	Number of Employees (Frequency)
20	105
30	199
40	29
50	73
Total	400

The given data can be represented as a vertical bar graph:

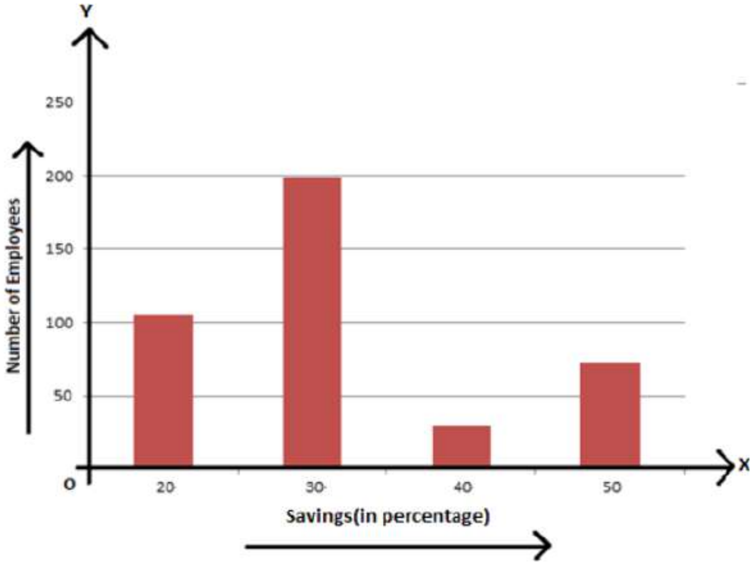


Figure 2.12: Vertical Bar Diagram for table 2.21

This can also be represented using a horizontal bar graph as follows:

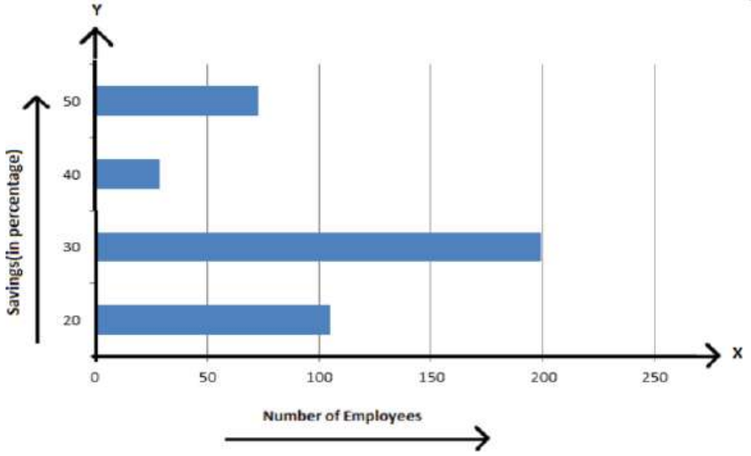


Figure 2.13: Horizontal Bar Diagram for table 2.21

Let as consider another example of grouped bar diagram: **A cosmetic companymanufactures 4 different shades of lipstick. The sale for 6 months is shown in the table. Represent it using bar charts.**

Table 2.22: Data of Lipsticks

Month	Sales (in units)			
	Shade 1	Shade 2	Shade 3	Shade 4
January	4500	1600	4400	3245
February	2870	5645	5675	6754
March	3985	8900	9768	7786
April	6855	8976	9008	8965
May	3200	5678	5643	7865
June	3456	4555	2233	6547

The graph given below depicts the following data:

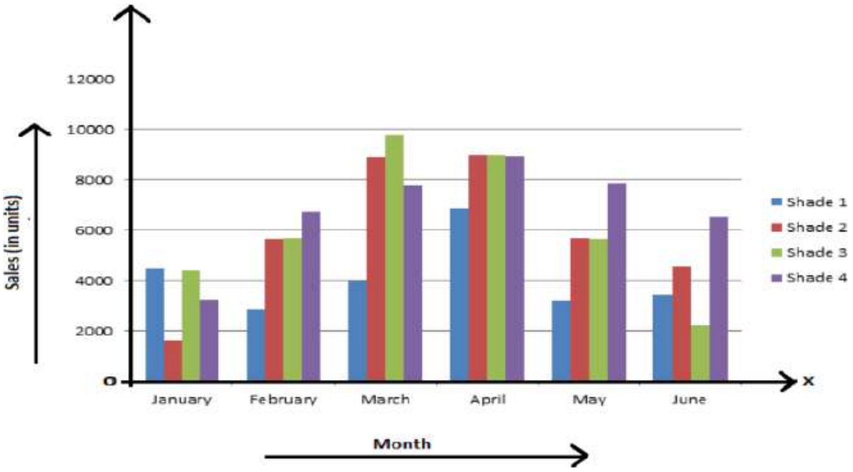


Figure 2.13: Grouped Bar Diagram for table 2.22

Bar diagram in Python

The **matplotlib** API in Python provides the bar() function. The syntax of the bar() function to be used with the axes is as follows:-

plt.bar(x, height, width, bottom, align)

The following program creates a bar plot bounded with a rectangle depending on the given parameters. Following is a simple example of the bar plot, which represents the number of students enrolled in different courses of an institute.

```
# Program 2.7: Bar Diagram
import numpy as np
import matplotlib.pyplot as plt
# creating the dataset
data = {'C':20, 'C++':15, 'Java':30,     'Python':35}
courses = list(data.keys())
values = list(data.values())
fig = plt.figure(figsize = (10, 5))
# creating the bar plot
plt.bar(courses, values, color ='maroon',      width = 0.4)
plt.xlabel("Courses offered")
plt.ylabel("No. of students enrolled")
plt.title("Students enrolled in different courses")
plt.show()
```

Output:

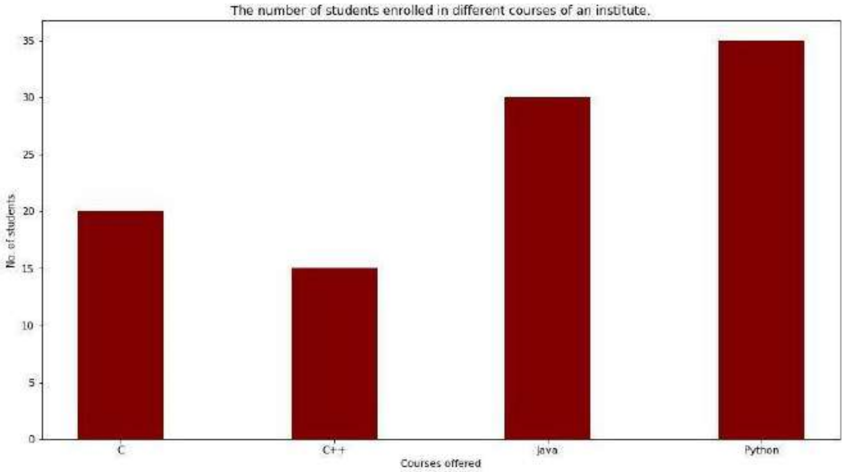


Figure 2.14: Bar Diagram

Horizontal charts can also be designed using Matplotlib. To create a horizontal bar chart:

```
#Program 2.8: Horizontal Bar Diagram

import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt
objects = ('Python', 'C++', 'Java', 'Perl', 'Scala', 'Lisp')
y_pos = np.arange(len(objects))
performance = [10,8,6,4,2,1]
plt.barh(y_pos, performance, align='center', alpha=0.5)
plt.yticks(y_pos, objects)
plt.xlabel('Usage')
plt.title('Programming language usage')
plt.show()
```

Output:

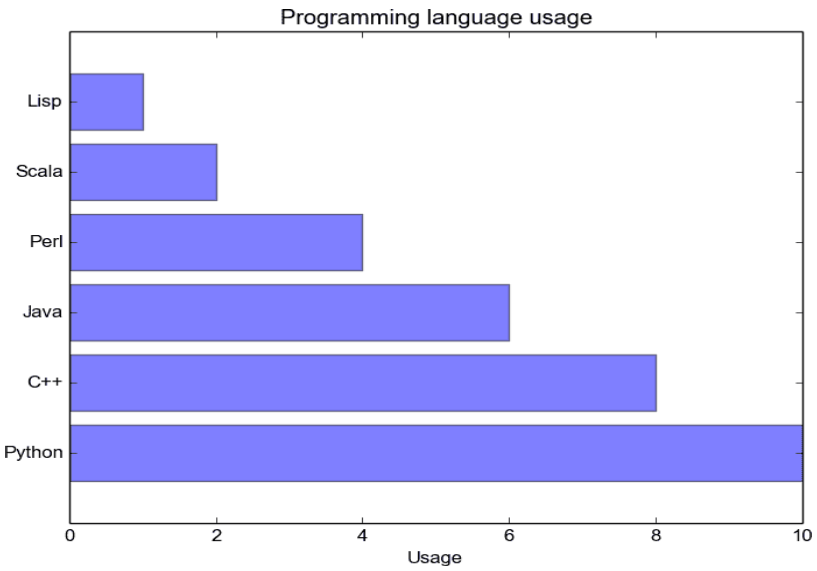


Figure 2.15: Horizontal Bar Diagram

Multiple bar plots: Multiple bar plots are used when comparison among the data set is to be done when one variable is changing. It can be drawn using python as shown in the following program.

```
#Program 2.9: Multiple Bar Plot
import numpy as np
import matplotlib.pyplot as plt
# set width of bar
barWidth = 0.25
fig = plt.subplots(figsize =(12, 8))
# set height of bar
IT = [12, 30, 1, 8, 22]
ECE = [28, 6, 16, 5, 10]
CSE = [29, 3, 24, 25, 17]
# Set position of bar on X axis
br1 = np.arange(len(IT))
br2 = [x + barWidth for x in br1]
br3 = [x + barWidth for x in br2]
# Make the plot
plt.bar(br1, IT, color ='r', width = barWidth, edgecolor ='grey', label ='IT')
plt.bar(br2, ECE, color ='g', width = barWidth, edgecolor ='grey', label ='ECE')
plt.bar(br3, CSE, color ='b', width = barWidth, edgecolor ='grey', label ='CSE')
# Adding Xticks
plt.xlabel('Branch', fontweight ='bold', fontsize = 15)
plt.ylabel('Students passed', fontweight ='bold', fontsize = 15)
plt.xticks([r + barWidth for r in range(len(IT))], ['2015', '2016', '2017', '2018', '2019'])
plt.legend()
plt.show()
```

Output:

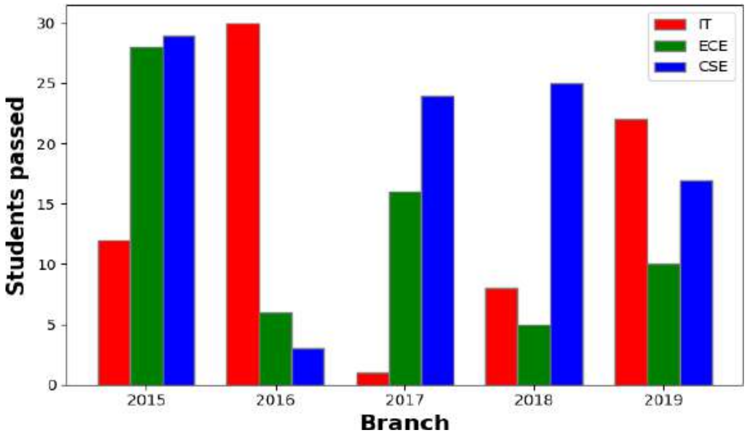


Figure 2.16: Multiple Bar Diagram

Stacked bar plot: Stacked bar plots represent different groups on top of one another. The height of the bar depends on the resulting height of the combination of the results of the groups. It goes from the bottom to the value instead of going from zero to value. The following bar plot represents the contribution of boys and girls in the team.

```
#Program 2.10: Stacked Bar Plot
import numpy as np
import matplotlib.pyplot as plt
N = 5
boys = (20, 35, 30, 35, 27)
girls = (25, 32, 34, 20, 25)
boyStd = (2, 3, 4, 1, 2)
girlStd = (3, 5, 2, 3, 3)
ind = np.arange(N)
width = 0.35
```

```
fig = plt.subplots(figsize =(10, 7))
p1 = plt.bar(ind, boys, width, yerr = boyStd)
p2 = plt.bar(ind, girls, width, bottom = boys, yerr = girlStd)
plt.ylabel('Contribution')
plt.title('Contribution by the teams')
plt.xticks(ind, ('T1', 'T2', 'T3', 'T4', 'T5'))
plt.yticks(np.arange(0, 81, 10))
plt.legend((p1[0], p2[0]), ('boys', 'girls'))
plt.show()
```

Output:

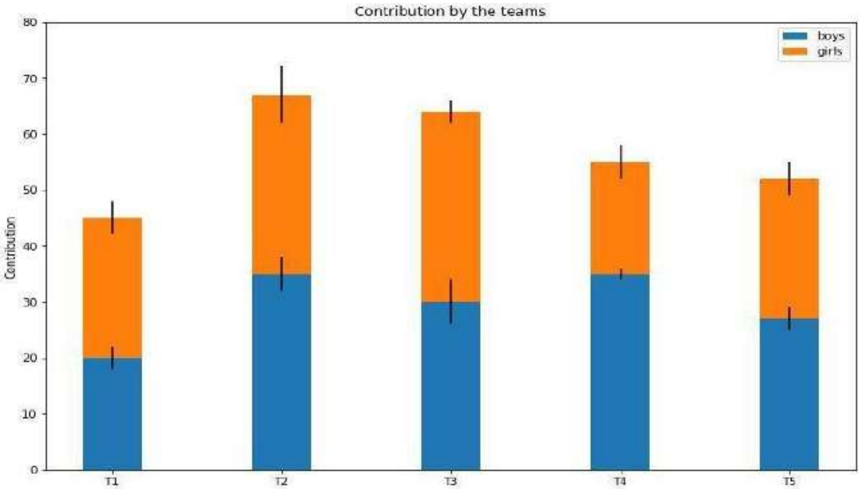


Figure 2.17: Stacked Bar Diagram

2.3 SUMMARY

In this module, representation of statistical data in the tabular and graphical forms has been discussed. The data representation in tabular form has been discussed terms of both grouped and ungrouped data. The presentation of continuous and discrete data has also been explained in this module. Further, the classification of data and its graphical representation has alsobeen discussed. The various types of graphs such as histogram, frequency polygon, bar diagram for presentation of data has been discussed at length. The concept of cumulative frequency distribution along with its tabular and graphical representation has also beenexplained in detail. The implementation of various types of graphs has been done in python. Overall, this module provides an insight of how to represent the statistical data.

2.4 PRACTICE QUESTION

Q.1 What are grouped and ungrouped frequency distributions? What are their uses? What are the considerations that one has to bear in mind while forming the frequency distribution?

Q.2 Explain the method of constructing Histogram and Frequency Polygon. Which out of these two is better representative of frequencies of (i) a particular group and (ii) whole group.

Q.3 What are the principles governing the choice of (i) Number of class intervals, (ii) The length of the class interval, (iii) The mid-point of the class interval.

Q.4 Write short notes on: (i) Frequency distribution, (ii) Histogram, frequency. polygon and frequency curve, (iii) Ogive.

Q.5 Write a program in python to draw various types of bar diagram considering your own data.

Q.6 Explain various types of bar diagram using suitable examples.

Q.7 What is cumulative frequency distribution? How can you represent c.f. graphically?

Q.8 The following numbers give the weights of 55 students of a class. Prepare a suitable frequency table.

42 74 40 60 82 115 41 61 75 83 63 53 110 76 84 50 67 65 78 77 56 95 68 69 104 80 79 79 54 73 59 81 100 66 49 77 90 84 76 42 64 69 70 80 72 50 79 52 103 96 51 86 78 94 71

(i) Draw the histogram and frequency polygon of the above data. (ii) For the above weights, make a cumulative frequency table and draw the less than ogive.

Q.9 A sample consists of 34 observations recorded correct to the nearest integer, ranging in value from 20l to 331. If it is decided to use seven classes of width 20 integers and to begin the first class at 199·5, find the class limits and class marks of the seven classes.

REFERENCES

- A. Abebe, J. Daniels, J.W.Mckean, “Statistics and Data Analysis”.
- A. Martelli, A. Ravenscroft, S. Holden, “Python in a Nutshell”, OREILLY.
- Clarke, G.M. & Cooke, D., “A Basic course in Statistics”, Arnold.
- David M. Lane, “Introduction to Statistics”.
- Eric Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction to Programming”.
- S.C.Gupta and V.K.Kapoor, “Fundamentals of Mathematical Statistics”, Sultan Chand & Sons, New Delhi.
- Weiss, N.A., “Introductory Statistics”, Addison Wesley

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

UNIT III: MEASUREMENT OF CENTRAL TENDENCY

STRUCTURE

3.0 OBJECTIVES

In this module, we will try to understand about the various measures of central tendency. The various measures of central tendency i.e., mean, median and mode will be discussed in detail. The various ways to find these measures will be explored in this module. Various measures of dispersion will also be discussed in this module. The various measures of dispersion that would be discussed are range, standard deviation, variance, coefficient of variation and mean deviation. In the end the concept of skewness and kurtosis will also be deliberated.

3.1 INTRODUCTION

This module is designed to know about the different measures of central tendency. In statistics, the measure of central tendency plays very important role as most of the analysis and interpretation surrounds these measures. In most of the cases either mean or median is used for interpreting any data. Mode is used to know about the locality of references in the data. Further, the scatteredness of data also plays an important role in data analysis. The scatteredness can be easily identified by measures of dispersion. There are number of measures of dispersion viz., range, mean deviation, variance, standard deviation, etc. Even though the combination of measures of central tendency and dispersion are good enough for very good interpretation about data, yet there are two more measures named as skewness and kurtosis with the help of which it can be known about the shape of the curve and hence better interpretation of data. This module is a helping hand to the basics of interpreting and analysing the data.

3.2 MAIN CONTENT

3.2.1 Measures of Central Tendency

According to Professor Bowley, averages are "statistical constants which enable us to comprehend in a single effort the significance of the whole." They give us an idea about the concentration of the values in the central part of the distribution. Plainly speaking, an average of a statistical series is the value of the variable which is representative of the entire distribution. The following are the five measures of central tendency that are in common use:
(i) Arithmetic Mean or Simply Mean, (ii) Median, (iii) Mode, (iv) Geometric Mean, and (v) Harmonic Mean.

However, in this course, we will be focussing only on first three measures.

Requirements for an Ideal Measure of Central Tendency

According to Professor Yule, the following are the characteristics to be satisfied by-an ideal measure of central tendency;

(i) It should be rigidly defined.
(ii) It should be readily comprehensible and easy to calculate.
(iii) It should be based on all the observations.
(iv) It should be suitable for further mathematical treatment.
(v) It should be affected as little as possible by fluctuations of sampling.
(vi) It should not be affected much by extreme values.

Arithmetic Mean

Arithmetic mean of a set of observations is their sum divided by the number of observations. e.g the arithmetic mean x of n observations $\bar{x}$ of n observations x_1, x_2, …, x_n, is given by

$$\bar{x} = \frac{1}{n}(x_1 + x_2 + \dots + x_n) = \frac{1}{n}\sum_{i=1}^{n} x_i$$

In case of frequency distribution, $x_i|f_i$, i = 1, 2, ..., n. where f_i is the frequency of the variable x_i;

$$\bar{x} = \frac{x_1 f_1 + x_2 f_2 + \cdots + x_n f_n}{f_1 + f_2 + \cdots + f_n} = \frac{\sum_{i=1}^{n} f_i x_i}{\sum_{i=1}^{n} f_i} = \frac{1}{N}\sum_{i=1}^{n} f_i x_i, \text{ where } \sum_{i=1}^{n} f_i = N.$$

In case of grouped or continuous frequency distribution. X is taken as the mid. value of the corresponding class.

Let us understand it via some examples.

- First consider the following data: 1600, 1590, 1560, 1610, 1640, 1630. Find the arithmetic mean

$$\bar{x} = \frac{1600+1590+1560+1610+1640+1630}{6} = 1605.$$

- Now consider the following frequency distribution and find its arithmetic mean.

X	1	2	3	4	5	6	7
F	5	9	12	17	14	10	6

X	F	fx
1	5	5
2	9	18
3	12	36
4	17	68
5	14	70
6	10	60
7	6	42
Total	73	299

Arithmetic Mean = $\bar{x} = \frac{299}{73} = 4.09$

- Let us consider another example:

Marks (x)	0-10	10-20	20-30	30-40	40-50	50-60
No. of students (f)	12	18	27	20	17	6

Marks	No. of students (f)	Mid-Point (x)	fx
0-10	12	5	60
10-20	18	15	270
20-30	27	25	675
30-40	20	35	700
40-50	17	45	765
50-60	6	55	330
Total	100		2800

Arithmetic Mean = $\bar{x} = \frac{2800}{100} = 28$

It may be noted that if the values of x or/and f are large, the calculation of mean is quite time consuming and tedious. The arithmetic is reduced to a great extent, by taking the deviations of the given values from any arbitrary point 'A', as explained below.

Let $d_i = x_i - A$, then $f_i d_i = f_i(x_i - A) = f_i x_i - f_i A$.

Applying summation both sides, we get

$$\sum_{i=1}^{n} f_i d_i = \sum_{i=1}^{n} f_i x_i - A \sum_{i=1}^{n} f_i = \sum_{i=1}^{n} f_i x_i - AN$$

$$\frac{1}{N}\sum_{i=1} f_i d_i = \frac{1}{N}\sum_{i=1} f_i x_i - A = \bar{x} - A$$

$$\bar{x} = A + \frac{1}{N}\sum_{i=1}^{n} f_i d_i$$

This formula is easy to handle with as compared to the earlier formula.

Any number can serve the purpose of arbitrary point 'A' but. usually, the value of x corresponding to the middle part of the distribution will be much more convenient.

In case of grouped or continuous frequency distribution, the arithmetic is reduced to a still greater extent by taking

$d_i = \frac{x_i - A}{h}$, where A is an arbitrary point and h is the common magnitude of class interval. In this case, we have $hd_i = x_i - A$, and proceeding exactly similarly as above, we get

$$\bar{x} = A + \frac{h}{N}\sum_{i=1}^{n} f_i d_i$$

Let us understand by an example:

- Consider the following distribution

Class Interval	0-8	8-16	16-24	24-32	32-40	40-48
Frequency	8	7	16	24	15	7

Let h = 8, A = 28

Class Interval	Mid-point (x)	Frequency (f)	D = (x – A) / h	fd
0-8	4	8	-3	-24
8-16	12	7	-2	-14
16-24	20	16	-1	-16

24-32	28	24	0	0
32-40	36	15	1	15
40-48	44	7	2	14
Total		77		-25

$$\bar{x} = A + \frac{h}{N}\sum_{i=1}^{n} f_i d_i = 28 + \frac{8}{28}(-25) = 25.404$$

Properties of Arithmetic Mean:

Property 1. Algebraic sum of the deviations of a set of values from their arithmetic mean is zero. If $x_i|f_i$, i = 1, 2, ... ,n is the frequency distribution, then

$\sum_{i=1}^{n} f_i (x_i - \bar{x}) = 0$, $\bar{x}$ being the mean of distribution.

Property 2. The sum of the squares of the deviations of a set of values is minimum when taken about mean.

Property 3. (Mean of the composite series). If $\bar{x}_i$, (i = 1, 2, ... , k) are the means of k-component series of sizes n_i, (i = 1, 2, ... , k) respectively, then the mean $\bar{x}$ of this composite series obtained on combining the component series given by the formula:

$$\bar{x} = \frac{n_1\bar{x}_1 + n_2\bar{x}_2 + \cdots + n_k\bar{x}_k}{n_1 + n_2 + \cdots + n_k} = \frac{\sum_{i=1}^{k} n_i\bar{x}_i}{\sum_{i=1}^{k} n_i}$$

Merits and Demerits of Arithmetic Mean

Merits. (i) It is rigidly defined. (ii) It is easy to understand and easy to calculate. (iii) It is based upon all the observations. (iv) Of all the averages, arithmetic mean is affected least by fluctuations of sampling. This property is sometimes described by saying that arithmetic mean is, a stable average.

Demerits. (i) It cannot be determined by inspection nor it can be located graphically. (ii) Arithmetic mean cannot be used if we are dealing with qualitative characteristics which cannot be measured quantitate; such as, intelligence, honesty, beauty, etc. In such cases median is the only average to be used. (iii) Arithmetic mean cannot be obtained if a single observation is missing or lost or is illegible unless we drop it out and compute the arithmetic mean of the remaining values. (iv) Arithmetic mean is affected very much by extreme values.In case of extreme items, arithmetic mean gives a distorted picture of the distribution and no longer remains representative of the distribution. (v) Arithmetic mean may lead to wrong conclusions if the details of the data from which it is computed are not given. (vi) Arithmetic mean cannot be calculated if the extreme class is open. Moreover, even if a single observation is missing mean cannot be calculated. (vii) In extremely asymmetrical (skewed) distribution, usually arithmetic mean is not a suitable measure of location.

Median

Median of a distribution is the value of the variable which divides it into two equal parts. It is the va1ue which exceeds and is exceeded by the same number of observations, i.e., it is the value such that the number of observations above it is equal to the number of observations below it. The median is thus a positional average. In case of ungrouped data, if the number of observations is odd then median is the middle value after the values have been arranged in ascending or descending order of magnitude. In case of even number of observations, there are two middle terms and median is obtained by taking the arithmetic mean of the middle terms. For example, the median of the values 25, 20, 15, 35, 18, i.e., 15, 18, 20, 25, 35 is 20 and the median of 8, 20, 50, 25, 15, 30, i.e., of 8, 15, 20, 25, 30, 50 is (20 + 25)/2 = 22·5.

In case of discrete frequency distribution median is obtained by considering the cumulative frequencies. The steps for calculating median are given below:

(i) Find N/2, where N = $\sum_{i=1}^{n} \mathbf{f}_i$.

(ii) See the (less than) cumulative frequency (cf.) just greater than N/2.

(iii) The corresponding value of x is median.

Let us understand it via an example: Obtain the median for the following frequency distribution:

X	1	2	3	4	5	6	7	8	9
F	8	10	11	16	20	25	15	9	6

X	F	c.f.
1	8	8
2	10	18
3	11	29
4	16	45
5	20	65
6	25	90
7	15	105
8	9	114
9	6	120

Here N = 120, Therefore N/2 = 60 and cumulative frequency just greater than 60 is 65. The value corresponding to 65 is 5. Hence median is 5.

In the case of continuous frequency distribution, the class corresponding to the cf. just greater than N/2 is called the median class and the value of median is obtained by the following formula:

Median = l + $\frac{h}{\mathbf{f}}(\frac{N}{2} - c)$

where l is the lower limit of the median class,

f is the frequency of the median class,

h is the magnitude of the median class,

'c' is the c.f. of the class preceding the median class,

And N = $\sum_{i=1}^{n} \mathbf{f}_i$.

Let us consider another example using above formula.

- Find the median wage of the following distribution:

Wages	20-30	30-40	40-50	50-60	60-70
No. of Labourer	3	5	20	10	5

Wages	No. of Labourer (f)	C.F.
20-30	3	3
30-40	5	8
40-50	20	28
50-60	10	38
60-70	5	43

N/2 = 21.5 and the value just greater than this is 28 in the c.f. column. Therefore, according the above-said formula:

Median = l + $\frac{h}{f}(\frac{N}{2} - c)$ = 40 + $\frac{10}{20}(\frac{43}{2} - 8)$ = 40 + 6.75 = 46.75.

Merits and Demerits of Median

Merits. (i) It is rigidly defined. (ii) It is easily understood and is easy to calculate. In some cases, it can be located merely by inspection. (iii) It is not at all affected by extreme values. (iv) It can be calculated for distributions with open-end classes.

Demerits. (i) In case of even number of observations median cannot be determined exactly. We merely estimate it by taking the mean of two middle terms. (ii) It is not based on all the observations. (iii) It is not amenable to algebraic treatment. (iv) As compared with mean, it is affected much by fluctuations of sampling.

Mode

In statistics, the **mode** is the value which is repeatedly occurring in a given set. We can also say that the value or number in a data set, which has a high frequency or appears more frequently is called mode or modal value.

For example, mode of the set {3, 7, 8, 8, 9}, is 8. Therefore, for a finite number of observations, we can easily find the mode. A set of values may have one mode or more than one mode or no mode at all.

Bimodal, Trimodal & Multimodal (More than one mode)

- When there are two modes in a data set, then the set is called **bimodal**

For example, the mode of Set A = {2,2,2,3,4,4,5,5,5} is 2 and 5, because both 2 and 5 is repeated three times in the given set.

- When there are three modes in a data set, then the set is called **trimodal**

For example, the mode of set A = {2,2,2,3,4,4,5,5,5,7,8,8,8} is 2, 5 and 8

- When there are four or more modes in a data set, then the set is called **multimodal**

In the case of discrete frequency distribution mode is the value of x corresponding tomaximum frequency. For example, in the following frequency distribution:

X	1	2	3	4	5	6	7	8
f	4	9	16	25	22	15	7	3

The value of X corresponding to the maximum frequency, viz., 25 is 4. Hence mode is 4.

In the case of grouped frequency distribution, calculation of mode just by looking into the frequency is not possible. To determine the mode of data in such cases we calculate the modal class. Mode lies inside the modal class. The mode of data is given by the formula:

$$Mode = l + \left(\frac{f_1 - f_0}{2f_1 - f_0 - f_2}\right) \times h$$

Where,

l = lower limit of the modal class

h = size of the class interval

f_1 = frequency of the modal class

f_0 = frequency of the class preceding the modal class

f_2 = frequency of the class succeeding the modal class

Let us take an example to understand this clearly.

- Find the mode for the following distribution:

Class Interval	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80
Frequency	5	8	7	12	28	20	10	10

Here maximum frequency is 28. Thus the class 40-50 is the modal class. The value of mode is given by

Mode = 40 + $\frac{10(28-12)}{(2*28-12-20)}$ = 40 + 6.66 = 46.66

Sometimes mode is estimated from the mean and the median. For a symmetrical distribution, mean, median and mode coincide. If the distribution is moderately asymmetrical, the mean, median and mode obey the following empirical relationship (due to Karl Pearson):

Mean - Median = $\frac{1}{3}$ (Mean - Mode)

Mode = 3 Median - 2 Mean

Merits and Demerits or Mode

Merits. (i) Mode is readily comprehensible and easy to calculate. Like median, mode can be located in some cases merely by inspection. (ii) Mode is not at all affected by extreme values. (iii) Mode can be conveniently located even if the frequency distribution has class-intervals of unequal magnitude provided the modal class and the classes preceding and succeeding it are of the same magnitude. Open end classes also do not pose any problem in the location of mode.

Demerits. (i) Mode is ill-defined. It is not always possible to find a clearly defined mode. (ii) It is not based upon all the observations. (iii) It is not capable of further mathematical treatment. (iv) As compared with mean, mode is affected to a greater extent by fluctuations ofsampling.

Measures of Central Tendencies using Python:

We can find mean of given data using following function:

mean([data-set])

Parameters :[data-set] : List or tuple of a set of numbers.

Returns: Simple arithmetic mean of the provided data-set.

```
# Program 3.1: Python program to demonstrate mean() function from the statistics module
# Importing the statistics module
import statistics
# list of positive integer numbers
data1 = [1, 3, 4, 5, 7, 9, 2]
x = statistics.mean(data1)
# Printing the mean
print("Mean is :", x)


Output:
Mean is : 4.428571428571429
```

We can find median of given data using following function:

median([data-set])

Parameters: [data-set]: List or tuple or an iterable with a set of numeric values

Returns: Return the median (middle value) of the iterable containing the data

```
# Program 3.2: Python code to demonstrate the working of median() function.
# importing statistics module
import statistics
# unsorted list of random integers
data1 = [2, -2, 3, 6, 9, 4, 5, -1]
```

```
# Printing median of the
# random data-set
print("Median of data-set is : % s " % (statistics.median(data1)))


Output:
Median of data-set is : 3.5
```

We can find mode of given data using following function:

mode([data-set])

Parameters : [data-set] *which is a tuple, list or a iterator of real valued numbers as well as Strings.*

Return type : *Returns the most-common data point from discrete or nominal data.*

```
# Program 3.3: Python code to demonstrate the use of mode() function
import statistics


# declaring a simple data-set consisting of real valued positive integers.
set1 =[1, 2, 3, 3, 4, 4, 4, 5, 5, 6]
# In the given data-set we can infer that 4 has the highest population distribution
# So mode of set1 is 4
# Printing out mode of given data-set
print("Mode of given data set is % s" % (statistics.mode(set1)))


Output:
Mode of given data set is 4
```

3.2.2 Measures of Dispersion

Literal meaning of dispersion is scatteredness. We study dispersion to have an idea about the homogeneity or heterogeneity of the distribution. The measures of central tendency serve to locate the center of the distribution, but they do not reveal how the items are spread out on either side of the center. This characteristic of a frequency distribution is commonly referred to as dispersion. In a series all the items are not equal. There is difference or variation among the values. The degree of variation is evaluated by various measures of dispersion. Small dispersion indicates the high uniformity of the items, while large dispersion indicates less uniformity.

Characteristics of a good measure of Dispersion

An ideal measure of dispersion is expected to possess the following properties

- It should be rigidly defined
- It should be based on all the items.
- It should not be unduly affected by extreme items.
- It should lend itself for algebraic manipulation.
- It should be simple to understand and easy to calculate.

The following are the measures of dispersion:

(i) Range, (ii) Mean deviation, (iii) Standard deviation, (iv) Variance, and (v) Coefficient of Variation.

Range

The range is the difference between two extreme observations, or the distribution. If A and B are the greatest and smallest observations respectively in a distribution, then its range is A-B. Range is the simplest but a crude measure of dispersion. Since it is based on two extreme observations which themselves are subject to chance fluctuations, it is not at all a reliable measure of dispersion.

In individual observations and discrete series, A and B are easily identified. In continuous series, the following method is followed. A = Upper boundary of the highest class B = Lower boundary of the lowest class.

Find the value of range for the following data: 8,10, 5,9,12,11

A = 12, B = 5

Range = A – B = 7

Calculate the range from the following distribution:

Class Interval	0-10	10-20	20-30	30-40	40-50	50-60	60-70	70-80
Frequency	5	8	7	12	28	20	10	10

Here the range = 80 – 0 = 80

Merits: 1. It is simple to understand. 2. It is easy to calculate. 3. In certain types of problems like quality control, weather forecasts, share price analysis, etc., range is most widely used.

Demerits: 1. It is very much affected by the extreme items. 2. It is based on only two extreme observations. 3. It cannot be calculated from open-end class intervals. 4. It is not suitable for mathematical treatment. 5. It is a very rarely used measure.

Mean Deviation

If $x_i|f_i$, i = 1, 2, …, n is the frequency distribution, then mean deviation from the average A, (either mean, median or mode), is given by

Mean deviation = $\frac{1}{N}\sum_i f_i|x_i - A|, \sum_i f_i = N$

where |Xi – A| represents the modulus or the absolute value of the deviation (Xi - A).

Since mean deviation is based on all the observations. it is a better measure of dispersion than range. But the step of ignoring the signs of the deviations |Xi – A| creates artificiality and, renders it useless for further mathematical treatment. It may be pointed out here that mean deviation is least when taken from median.

Let us consider few examples:

- Calculate Mean Deviation about Mean for the numbers given below: 1,2,3,4,5.

Here, Mean = $\bar{x} = \frac{\sum_i x_i}{n} = \frac{15}{5} = 3$

x	$\|x - \bar{x}\|$
1	2
2	1
3	0
4	1
5	2
$\sum_i x_i$=15	$\sum_i \|x - \bar{x}\| = 6$

M.D. about Mean = $\frac{\sum_i |x - \bar{x}|}{N} = \frac{6}{5} = 1.2$

Consider another examples based of discrete frequency distribution.

- Calculate the M.D. from Mean for the following data:

X	F	Fx	$\|x - \bar{x}\| = \|x - 6\|$	$f\|x - \bar{x}\|$
2	1	2	4	4
4	4	16	2	8
6	6	36	0	0
8	4	32	2	8
10	1	10	4	4
Total	16	96	14	24

Mean = $\bar{x} = \frac{\sum_i f_i x_i}{\sum_i f_i} = \frac{96}{16} = 6$

M.D. about Mean = $\frac{\sum_i f_i |x - \bar{x}|}{\sum_i f_i} = \frac{24}{16} = 1.5$

- Consider another examples based of continuous frequency distribution.

Marks	No. of Students(f)	Middle-Point(x)	fx	$\|x - \bar{x}\| = \|x - 33.4\|$	$f\|x - \bar{x}\|$
0-10	6	5	30	28.4	170.4
10-20	5	15	75	18.4	92
20-30	8	25	200	8.4	67.2
30-40	15	35	525	1.6	24
40-50	7	45	315	11.6	81.2
50-60	6	55	330	21.6	129.6
60-70	3	65	195	31.6	94.8
Total	50		1670		659.2

Mean = $\bar{x} = \frac{\sum_i f_i x_i}{\sum_i f_i} = \frac{1670}{50} = 33.4$

M. D. about Mean = $\frac{\sum_i f_i |x - \bar{x}|}{\sum_i f_i} = \frac{659.2}{50} = 13.18$

Merits: 1. It is simple to understand and easy to compute. 2. It is rigidly defined. 3. It is based on all items of the series. 4. It is not much affected by the fluctuations of sampling. 5. It is less affected by the extreme items. 6. It is flexible, because it can be calculated from any average. 7. It is a better measure of comparison.

Demerits: 1. It is not a very accurate measure of dispersion. 2. It is not suitable for further mathematical calculation. 3. It is rarely used. It is not as popular as standard deviation. 4. Algebraic positive and negative signs are ignored. It is mathematically unsound and illogical.

Mean Deviation using Python:

Absolute mean deviation can be calculated in python using following code:

Using Numpy

```
# Program 3.4: Mean Deviation using Numpy
# Importing mean, absolute from numpy
data = [75, 69, 56, 46, 47, 79, 92, 97, 89, 88, 36, 96, 105, 32, 116, 101, 79, 93, 91, 112]
# Absolute mean deviation
mean(absolute(data - mean(data)))
Output:
20.055
```

Using Pandas

```
#Program 3.5: Mean Deviation using Pandas
# Import the pandas library as pd
import pandas as pd
data = [75, 69, 56, 46, 47, 79, 92, 97, 89, 88, 36, 96, 105, 32, 116, 101, 79, 93, 91, 112]
# Creating data frame of the given data
df = pd.DataFrame(data)
# Absolute mean deviation
df.mad() # mad() is mean absolute deviation function
Output:
20.055
```

Standard Deviation and Variance

Standard deviation, usually denoted by the sigma (σ), is the positive square root of the arithmetic mean of the squares of the deviations of the given values from their arithmetic mean. For the frequency distribution $x_i|f_i$, i = 1,2, ... , n.

$\sigma = \sqrt{\frac{1}{N}\sum_i f_i(x_i - \bar{x})^2}$, where $\bar{x}$ is the arithmetic mean of the distribution and $\sum_i f_i = N$.

The step of squaring the deviations $(x_i - \bar{x})$ overcomes the drawback of ignoring the signs in mean deviation. Standard deviation is also suitable for further mathematical treatment. Moreover, of all the measures, standard deviation is affected least by fluctuations of sampling.

The square of standard deviation is called the variance and is given by

$$\sigma^2 = \frac{1}{N}\sum_i f_i(x_i - \bar{x})^2$$

For individual series, standard deviation and variance can be calculated as follows:

$$\sigma = \sqrt{\frac{\sum x^2}{n} - \left(\frac{\sum x}{n}\right)^2}, \qquad \sigma^2 = \frac{\sum x^2}{n} - \left(\frac{\sum x}{n}\right)^2$$

Different formulae for calculating variance:

Now we know that variance for any frequency distribution may be written as:

$$\sigma^2 = \frac{1}{N}\sum_i f_i(x_i - \bar{x})^2$$

However, it can better be written as:

$$\sigma_x^2 = \frac{1}{N}\sum_i f_i(x_i - \bar{x})^2$$

If $\bar{x}$ is not a whole number, the calculation of σ_x^2 is very cumbersome and time consuming. Therefore, formula can be changed to

$$\sigma_x^2 = \frac{1}{N}\sum_i f_i x_i^2 - \left(\frac{1}{N}\sum_i f_i x_i\right)^2$$

If the values of x and f are large, the calculation of fx, fx^2 is quite tedious. In that case we take the deviations from any arbitrary point 'A '. Generally, the point in the middle of the distribution is much convenient and therefore we have

$$\sigma_x^2 = \frac{1}{N}\sum_i f_i d_i^2 - \left(\frac{1}{N}\sum_i f_i d_i\right)^2, \text{ where } d_i = x_i - A$$

We can make the calculations easier by using

$$\sigma_x^2 = h^2\left[\frac{1}{N}\sum_i f_i d_i^2 - \left(\frac{1}{N}\sum_i f_i d_i\right)^2\right], \text{ where } d_i = \frac{x_i - A}{h}$$

Now let us take some examples to understand the standard deviation and variance.

- Calculate S.D. for the data given below.

Sr. No.	Marks (X)	X^2
1	5	25
2	10	100
3	20	400
4	25	625
5	40	1600
6	42	1764
7	45	2025
8	48	2304
9	70	4900
10	80	6400
Total	385	20143

Mean = $\bar{x} = \frac{385}{10} = 38.5$, $\sigma = \sqrt{\frac{\sum x^2}{n} - \left(\frac{\sum x}{n}\right)^2} = \sqrt{\frac{20143}{10} - 38.5^2} = 23.07$

- Calculate S.D. for the following data.

X	F	Fx	X^2	Fx^2
6	7	42	36	252
9	12	108	81	972
12	13	156	144	1872
15	10	150	225	2250
18	8	144	324	2592
Total	50	600		7938

$$\sigma = \sqrt{\frac{\sum fx^2}{\sum f} - \left[\frac{\sum fx}{\sum f}\right]^2} = \sqrt{\frac{7938}{50} - 12^2} = 3.84$$

- Calculate the mean and standard deviation for tile following table giving tile age distribution of 542 members.

Age	20-30	30-40	40-50	50-60	60-70	70-80	80-90
No. of Members	3	61	132	153	140	51	2

In this data let us take d = $\frac{x-A}{h} = \frac{x-55}{10}$

Age	Mid-Value(x)	Members(f)	D=$\frac{x-55}{10}$	Fd	Fd^2
20-30	25	3	-3	-9	27
30-40	35	61	-2	-132	244
40-50	45	132	-1	-132	132
50-60	55	153	0	0	0
60-70	65	140	1	140	140
70-80	75	51	2	102	204
80-90	85	2	3	6	18
Total		542		-15	765

$\bar{x} = A + h\frac{\sum fd}{N} = 55 + 10(\frac{-15}{542}) = 54.72$

$\sigma_x^2 = h^2[\frac{1}{N}\sum_i f_i d_i^2 - (\frac{1}{N}\sum_i f_i d_i)^2] = 100[\frac{765}{542} - (\frac{-15}{542})^2] = 133.3$

σ = 11.55 years.

Standard Deviation and Variance in Python:

It can be calculated in python using many ways. However, we will be considering only two ways i.e. using Numpy and Statistics packages.

Using Numpy:

One can calculate the standard deviation by using **numpy.std()** function in python.

numpy.std(a, axis=None, dtype=None, out=None, ddof=0, keepdims=<no value>)

Parameters:

a: Array containing data to be averaged

axis: Axis or axes along which to average a

dtype: Type to use in computing the variance.

out: Alternate output array in which to place the result.

ddof: Delta Degrees of Freedom

keepdims: If this is set to True, the axes which are reduced are left in the result as dimensions with size one

```
# Program 3.6: Python program to get standard deviation of a list

# Importing the NumPy module

import numpy as np

# Taking a list of elements

list = [2, 4, 4, 4, 5, 5, 7, 9]

# Calculating standard deviation using std()

print(np.std(list))

Output:

2.0
```

One can calculate the variance by using **numpy.var()** function in python.

numpy.var(a, axis=None, dtype=None, out=None, ddof=0, keepdims=<no value>)

Parameters:

a: Array containing data to be averaged *axis: Axis or axes along which to average a* *dtype: Type to use in computing the variance.*

out: Alternate output array in which to place the result.

ddof: Delta Degrees of Freedom

keepdims: If this is set to True, the axes which are reduced are left in the result as dimensions with size one

```
# Program 3.7: Python program to get variance of a list

# Importing the NumPy module

import numpy as np

# Taking a list of elements

list = [2, 4, 4, 4, 5, 5, 7, 9]

# Calculating variance using var()

print(np.var(list))

Output:

4.0
```

Using Statistics:

variance([data], xbar)

Parameters :
[data] : An iterable with real valued numbers.
xbar (Optional) : Takes actual mean of data-set as value.

Returnype : Returns the actual variance of the values passed as parameter.

```
# Program 3.8: Python code to demonstrate the working of variance() function of Statistics

# Importing Statistics module

import statistics

# Creating a sample of data

sample = [2.74, 1.23, 2.63, 2.22, 3, 1.98]

# Prints variance of the sample set

# Function will automatically calculate it's mean and set it as xbar

print("Variance of sample set is % s" %(statistics.variance(sample)))


Output:

Variance of sample set is 0.40924
```

Another example of coding for finding the variance is:

```
# Program 3.9: Python code to demonstrate the use of xbar parameter for variance
# Importing statistics module
import statistics
# creating a sample list
sample = (1, 1.3, 1.2, 1.9, 2.5, 2.2)
# calculating the mean of sample set
m = statistics.mean(sample)
# calculating the variance of sample set
print("Variance of Sample set is % s" %(statistics.variance(sample, xbar = m)))


Output:
Variance of Sample set is 0.3656666666666667
```

Standard Deviation in Python using Statistics package can be coded as:

stdev([data-set], xbar)

Parameters :

[data] : An iterable with real valued numbers.

xbar (Optional): Takes actual mean of data-set as value.

Returnype : Returns the actual standard deviation of the values passed as parameter.

```
# Program 3.10: Python code to demonstrate stdev() function
# importing Statistics module
import statistics
# creating a simple data - set
sample = [1, 2, 3, 4, 5]
# Prints standard deviation
# xbar is set to default value of 1
print("Standard Deviation of sample is % s " % (statistics.stdev(sample)))


Output:
Standard Deviation of the sample is 1.5811388300841898
```

Another example of coding for finding the standard deviation is:

```
# Program 3.11: Python code to demonstrate use of xbar parameter while using stdev() function

# Importing statistics module

import statistics

# creating a sample list

sample = (1, 1.3, 1.2, 1.9, 2.5, 2.2)

# calculating the mean of sample set

m = statistics.mean(sample)

# xbar is nothing but stores the mean of the sample set

# calculating the standard deviation of sample set

print("Standard Deviation of Sample set is % s" %(statistics.stdev(sample, xbar = m)))



Output:

Standard Deviation of Sample set is 0.6047037842337906
```

Coefficient of Variation

Coefficient of variation can be found using the following formula:

Coefficient of Variation (C.V.) $= \frac{Standard\ Deviation}{Arithmetic\ Mean} * 100 = \frac{\sigma}{\bar{x}} * 100.$

Let us consider an example for this:

- The means and standard deviation values for the number of runs of two players A and B are 55; 65 and 4.2; 7.8 respectively. Who is the more consistent player?

Coefficient of variation of Player A $= \frac{\sigma}{x} * 100 = \frac{4.2}{55} * 100 = 7.64$

Coefficient of variation of Player B $= \frac{\sigma}{x} * 100 = \frac{7.8}{65} * 100 = 12$

Coefficient of variation of player A is less. Therefore, Player A is the more consistent player.

Coefficient of variation can be calculated using python as follows:

scipy.stats.variation(arr, axis = None) *function computes the coefficient of variation. It is defined as the ratio of standard deviation to mean.*

Parameters:

arr: *[array_like] input array.*

axis: *[int or tuples of int] axis along which we want to calculate the coefficient of variation.*

-> *axis = 0 coefficient of variation along the column.*

-> *axis = 1 coefficient of variation working along the row.*

Results: *Coefficient of variation of the array with values along specified axis.*

```
# Program 3.12: Coefficient of Variation
from scipy.stats import variation
import numpy as np
arr = np.random.randn(5, 5)
print ("array : \n", arr)
# rows: axis = 0, cols: axis = 1
print ("\nVariation at axis = 0: \n", variation(arr, axis = 0))
print ("\nVariation at axis = 1: \n", variation(arr, axis = 1))



Output:
array :
 [[-1.16536706 -1.29744691 -0.39964651  2.14909277 -1.00669835]
 [ 0.79979681  0.91566149 -0.823054    0.9189682  -0.01061181]
 [ 0.9532622   0.38630077 -0.79026789 -0.70154086  0.79087801]
 [ 0.53553389  1.46409899  1.89903817 -0.35360202 -0.14597738]
 [-1.53582875 -0.50077039 -0.23073327  0.32457064 -0.43269088]]

Variation at axis = 0:
 [-12.73042404   5.10272979 -14.6476392    2.15882202  -3.64031032]

Variation at axis = 1:
 [-3.73200773  1.90419038  5.77300406  1.29451485 -1.27228112]
```

3.2.3 Skewness

Literally skewness means 'lack of symmetry‟. We study skewness to have an idea about the shape of the curve which we can draw with the help of the given data. A distribution is said to be skewed if

(i) Mean, median and mode fall at different points. i.e., Mean G Median G Mode

(ii) The curve drawn with the help of the given data is not symmetrical but stretched more to one side than to the other.

Measures of Skewness: Various measures of skewness are

- $S_k = M - M_d$
- $S_k = M - M_o$,

where M is the mean, M_d the median and M_o is the mode of the distribution.

These are the absolute measures of skewness. As in dispersion, for comparing two series we do not calculate these absolute measures but we calculate the relative measures called the co-efficient of skewness which are pure numbers independent of units of measurement. The following is the coefficients of Skewness.

Prof. Karl Pearson's Coefficient of Skewness (S_k) = $\frac{(M-M0)}{\sigma}$, where σ is the standard deviation of the distribution.

If mode is ill-defined, then using the relation, $M_o = 3M_d - 2M$, for a moderately asymmetrical distribution, we get

$S_k = \frac{3(M-M_d)}{\sigma}$

The limits for Karl Pearson's coefficient of skewness are ± 3. In practice, these limits are rarely attained. Skewness is positive if $M > M_o$ or $M > M_d$ and negative if $M < M_o$ or $M < M_d$.

Skewness can be represented graphically as shown in figure 3.1:

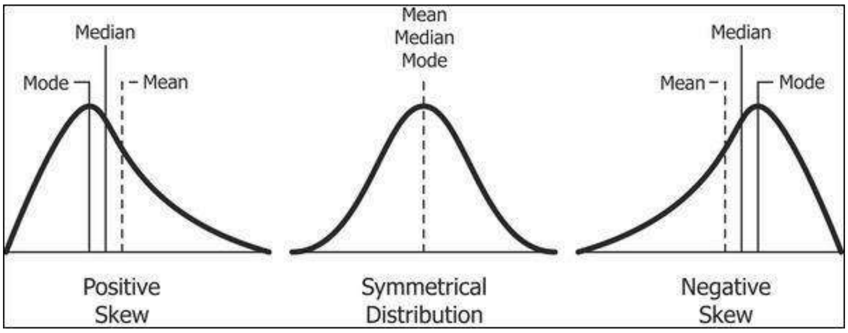


Figure 3.1: Skewness

Skewness can be calculated as per following examples:

- From the marks secured by 120 students in Section A and B of a class, the Following measures are obtained:

Section A: $\bar{X}$= 46.83; S.D = 14.8; Mode = 51.67

Section B: $\bar{X}$= 47.83; S.D = 14.8; Mode = 47.07

Determine which distribution of marks is more skewed.

Karl Pearson's Coefficient of Skewness for Section A = $S_k = \frac{(M-M0)}{\sigma} = \frac{(46.83-51.67)}{14.8}$ = -0.3270

Karl Pearson's Coefficient of Skewness for Section B = $S_k = \frac{(M-M0)}{\sigma} = \frac{(47.83-47.07)}{14.8} = 0.05135$

Marks of Section A is more Skewed. But marks of Section A are negatively Skewed. Marks of Section B are Positively skewed.

- From a moderately skewed distribution of retail prices for men‟s shoes, it is found that the mean price is Rs. 20 and the median price is Rs. 17. If the coefficient of variation is 20%, find the Pearsonian coefficient of skewness of the distribution.

Coefficient of Variation (C.V.) $= \frac{Standard\ Deviation}{Arithmetic\ Mean} * 100 = \frac{\sigma}{\bar{x}} * 100$

$20 = \frac{\sigma}{20} * 100 \implies \sigma = 4$

$S_k = \frac{3(M-M_d)}{\sigma} = \frac{3(20-17)}{4} = 2.25.$

Skewness may be calculated using python as follows:

scipy.stats.skew(array, axis=0, bias=True) *function calculates the skewness of the data set.*

Parameters :
array : *Input array or object having the elements.*
axis : *Axis along which the skewness value is to be measured. By default axis = 0.* ***bias :*** *Bool; calculations are corrected for statistical bias, if set to False.*

Returns: *Skewness value of the data set, along the axis.*

```
#Program 3.13: finding Skewness
from scipy.stats import skew
import numpy as np
# random values based on a normal distribution
x = np.random.normal(0, 2, 10000)
print ("X : \n", x)
print('\nSkewness for data : ', skew(x))


Output:
X :
 [ 0.03255323 -6.18574775 -0.58430139 ... 3.22112446  1.16543279 0.84083317]
Skewness for data: 0.03248837584866293
```

3.2.4 Kurtosis

If we know the measures of central tendency, dispersion and skewness, we still cannot form a complete idea about the distribution as will be clear from the figure 3.2 in which all the three curves A, B and C are symmetrical about the mean and have the same range.

In addition to these measures we should know one more measure which Prof. Karl Pearson calls as the Convexity of curve or Kurtosis. Kurtosis enables us to have an idea about the flatness -or peakedness of the curve, It is measured by the co-efficient β_2 or its derivation γ_2 given by

$\beta_2 = \frac{\mu_4}{\mu_2^2}, \gamma_2 = \beta_2 - 3$

Where, $\mu_2, and\ \mu_4$ are moments about mean and can be calculated using following formulae:

The rth moment of a variable about the mean $\bar{X}$ usually denoted by μ_r is given by:

$\mu_r = \frac{1}{N}\sum_i f_i(x_i - \bar{x})^r$, In particular $\mu_0 = \frac{1}{N}\sum_i f_i(x_i - \bar{x})^0 = \frac{\sum_i f_i}{N} = 1.$

$\mu_1 = \frac{1}{N}\sum_i f_i(x_i - \bar{x})^1 = 0$, being the algebraic sum of deviations from the mean.

Also, $\mu_2 = \frac{1}{N}\sum_i f_i(x_i - \bar{x})^2 = \sigma^2$

The rth moment of a variable x about any arbitrary point x = A, usually denoted by μ'_r is given by:

$\mu'_r = \frac{1}{N}\sum_i f_i(x_i - A)^r = \frac{1}{N}\sum_i f_i d_i$

Where $d_i = x_i - A$.

The relations between moments about an arbitrary point and about mean is represented as:

$\mu_2 = \mu'_2 - \mu'^2_1$,

$\mu_3 = \mu'_3 - 3\mu'_2\mu'_1 + 2\mu'^3_1$,

$\mu_4 = \mu'_4 - 4\mu'_3\mu'_1 + 6\mu'_2\mu'^2_1 - 3\mu'^4_1$.

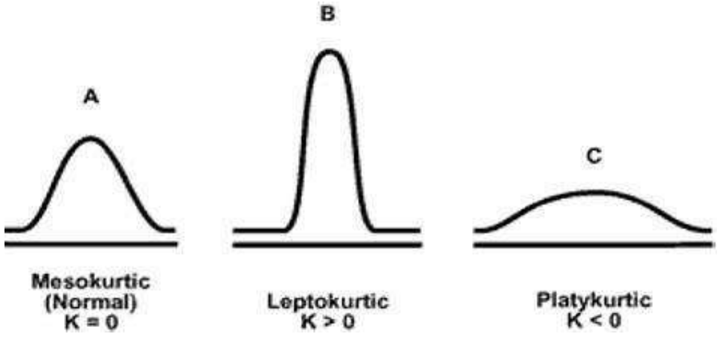


Figure 3.2: Kurtosis

Curve of the type 'A' which is neither flat nor peaked is called the normal curve or mesokurtic curve and for such a curve β_2 = 3, i.e., γ_2 = O. Curve of the type 'C' which is flatter than the normal curve is known as platykurtic curve and for such a curve β_2 < 3, i.e., γ_2 < O. Curve of the type 'B' which is more peaked than the normal curve is called leptokurtic and for such a curve, β_2 > 3 i.e.; γ_2 > 0.

Let us understand the concept by using an example.

- The data on daily wages of 45 workers of a factory are given. Compute skewness and kurtosis using moment about the mean. Comment on the results.

Wages	100-120	120-140	140-160	160-180	180-200	200-220	220-240
No. of Workers	1	2	6	20	11	3	2

Solution:

Wages	No. of Workers(f)	Mid-Point(x)	D=$\frac{x-170}{20}$	Fd	Fd^2	Fd^3	Fd^4
100-120	1	110	-3	-3	9	-27	81
120-140	2	130	-2	-4	8	-16	32
140-160	6	150	-1	-6	6	-6	6
160-180	20	170	0	0	0	0	0
180-200	11	190	1	11	11	11	11
200-220	3	210	2	6	12	24	48
220-240	2	230	3	6	18	54	162
Total	45			10	64	40	330

$\mu'_1 = \frac{\Sigma fd}{N} * \text{h} = \frac{10}{45} * 20 = 4.44$

$\mu'_2 = \frac{\Sigma fd^2}{N} * \text{h}^2 = \frac{64}{45} * 20^2 = 568.88$

$\mu'_3 = \frac{\Sigma fd^3}{N} * \text{h}^3 = \frac{40}{45} * 20^3 = 7111.11$

$\mu'_4 = \frac{\Sigma fd^4}{N} * \text{h}^4 = \frac{330}{45} * 20^4 = 1173333.33$

Moments about mean are:

$\mu_2 = \mu'_2 - \mu'^2_1 = 549.16$

$\mu_3 = \mu'_3 - 3\mu'_2\mu'_1 + 2\mu'^3_1 = -291.32$

$\mu_4 = \mu'_4 - 4\mu'_3\mu'_1 + 6\mu'_2\mu'^2_1 - 3\mu'^4_1 = 1113162.18$

Therefore,

Skewness = $\beta_1 = \mu_3^2$ = 0.00051

Kurtosis = $\beta_2 = \frac{\mu_4}{\mu_2^2}$ = 3.69

From the above calculations, it can be concluded that skewness is almost zero, thereby indicating that the distribution is almost symmetrical. Kurtosis, has a value greater than 3, thus implying that the distribution is leptokurtic.

Kurtosis in python can be calculated as follows:

scipy.stats.kurtosis(array, axis=0, fisher=True, bias=True) *function calculates the kurtosis (Fisher or Pearson) of a data set.*

Parameters:

array: *Input array or object having the elements.*

axis: *Axis along which the kurtosis value is to be measured. By default, axis = 0.*

fisher: *Bool; Fisher's definition is used (normal 0.0) if True; else Pearson's definition is used (normal 3.0) if set to False.*

bias: *Bool; calculations are corrected for statistical bias, if set to False.*

Returns: *Kurtosis value of the normal distribution for the data set.*

3.3 SUMMARY

In this module, various measures of central tendency viz., mean, median and mode have been discussed. Various methods for calculating mean, median and mode have been discussed for grouped, ungrouped, discrete and continuous data. The measure of dispersion such as range, mean deviation, standard deviation, variance and coefficient of variation have been discussed in detail. Different techniques for evaluating these measures have also been elaborated. Following this, skewness and kurtosis along with their measures have also been inspected. All these types of measures have been implemented in python using numpy, pandas, matplotlib and statistics packages.

3.4 PRACTICE QUESTION

Q.1 Compare mean, median and mode as measures of location of a distribution.

Q.2 Describe the different measures of central tendency of a frequency distribution, mentioning their merits and demerits.

Q.3 Given below is the distribution of 140 candidates obtaining marks X or higher in it certain examination (all marks are given in whole numbers):

X	10	20	30	40	50	60	70	80	90	100
c.f.	140	133	118	100	75	45	25	9	2	0

Calculate the mean, median and mode of the distribution.

Q.4 The mean of marks obtained in an examination by a group of 100 students was found to be 49·96. The mean of the marks obtained in the same examination by another group of 200 students was 52·32. Find the mean of the marks obtained by both the groups of students taken together.

Q.5 Which measure of location will be suitable to compare: (i) heights of students in two classes; (ii) size of agricultural holdings; (iii) average sales for various years; (iv) intelligence of students; (v) per capita income in several countries.

Q.6 Explain with suitable examples the term dispersion. State the relative and absolute measures of dispersion and describe the merits and demerits of these.

Q.7 Explain the main difference between mean deviation and standard deviation.

Q.8 Age distribution of hundred life insurance policy holders is as follows:

Age	17-19.5	20-25.5	26-35.5	36-40.5	41-50.5	51-55.5	56-60.5	61-70.5

Number	9	16	12	26	14	12	6	5

Calculate mean deviation from median age.

Q.9 What is standard deviation? Explain its superiority over other measures of dispersion.

Q.10 Calculate the mean and standard deviation of the following distribution:

X	2.5-7.5	7.5-12.5	12.5-17.5	17.5-22.5	22.5-27.5	27.5-32.5	32.5-37.5
f	65	121	175	198	176	120	66

Q. 11 The mean of 5 observations is 4·4 and variance is 8·24. It three of the five observation are 1, 2 and 6; find the other two.

Q.12 Lives of two models of refrigerators turned in for new models in a recent survey are:

Life	Model A	Model B
0-2	5	2
2-4	16	7
4-6	13	12
6-8	7	19
8-10	5	9
10-12	4	1

What is the average life of each model of these refrigerators? Which model shows more uniformity?

Q. 13 What do you understand by skewness? How is it measured? Distinguish clearly, by giving figures, between positive and negative skewness.

Q.14 Explain the methods of measuring skewness and kurtosis of a frequency distribution.

Q.15 Obtain Karl Pearson‟s measure of skewness and kurtosis for the following data:

Values	5-10	10-15	15-20	20-25	25-30	30-35	35-40
Frequency	6	8	17	21	15	11	2

REFERENCES

- A. Abebe, J. Daniels, J.W.Mckean, “Statistics and Data Analysis”.
- A. Martelli, A. Ravenscroft, S. Holden, “Python in a Nutshell”, OREILLY.
- Clarke, G.M. & Cooke, D., “A Basic course in Statistics”, Arnold.
- David M. Lane, “Introduction to Statistics”.
- Eric Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction to Programming”.
- S.C.Gupta and V.K.Kapoor, “Fundamentals of Mathematical Statistics”, Sultan Chand & Sons, New Delhi.

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

UNIT IV: DESCRIPTIVE STATISTICS

STRUCTURE

4.0 OBJECTIVES

In this module, we will try to understand about the descriptive statistics along with its importance and limitations followed by its implementation on python. Then a detailed discussion about exploratory data analysis will be undertaken. Various tools used in exploratory data analysis will also be explored. The implementation of these tools in python will also be deliberated upon. The concept of data visualization will also be taken up in the last part. The various types of data visualization methods will be considered along with their implementation in python.

4.1 INTRODUCTION

This module is designed to know about the overall presentation of data both in a textual manner as well as graphical manner. Descriptive statistics is used to describe the data as a whole, which mean the various measures of data are evaluated using descriptive statistics. It provides an overall description about the data and that"s why known descriptive statistics. The exploratory data analysis presents the various tools used for interpreting and analysingthe data. Various types of graphical methods are used in this type of analysis. Further, data visualization is a concept, which is backbone of data analysis. If we are able to analyse data in a good manner, but not able to present it in an effective manner, it will be no or little use. That"s where comes the need of data visualization. Data visualization provides an effective way of presenting our data.

4.2 MAIN CONTENT

4.2.1 Descriptive Statistics

Descriptive statistics summarize and organize characteristics of a data set. A data set is a collection of responses or observations from a sample or entire population. In quantitative research, after collecting data, the first step of statistical analysis is to describe characteristics of the responses, such as the average of one variable (e.g., age), or the relation between two variables (e.g., age and creativity).

Purpose of Descriptive Statistics

The essential features of the data of a study are defined using descriptive statistics. The sample and measurements are summarized. They form the basis of practically any quantitative analysis of the data in combination with simple graphic analysis. Also, a data set can be summarised and described using descriptive statistics through a variety of tabulated and graphical explanations and discussion of the observed results. Complex quantitative data are summed up in descriptive statistics. Descriptive statistics can be helpful to

- providing essential data on variables in a dataset
- highlighting possible relationships between variables.

Types of descriptive statistics

There are 3 main types of descriptive statistics:

- The distribution concerns the frequency of each value.
- The central tendency concerns the averages of the values.

- The variability or dispersion concerns how spread out the values are.

We have already discussed all three types of descriptive statistics in the previous units.

We can apply these to assess only one variable at a time, in univariate analysis, or to compare two or more, in bivariate and multivariate analysis.

Univariate descriptive statistics: Univariate descriptive statistics focus on only one variable at a time. It"s important to examine data from each variable separately using multiplemeasures of distribution, central tendency and spread.

Bivariate and Multivariate descriptive statistics: If we"ve collected data on more than one variable, we can use bivariate or multivariate descriptive statistics to explore whether there are relationships between them. In bivariate analysis, you simultaneously study the frequency and variability of two variables to see if they vary together. You can also compare the central tendency of the two variables before performing further statistical tests. Similarly, multivariate analysis can be applied for more than two variables.

Importance of Descriptive Statistics

Descriptive statistics allow for the ease of data visualization. It allows for data to be presented in a meaningful and understandable way, which, in turn, allows for a simplified interpretation of the data set in question. Raw data would be difficult to analyse, and trend and pattern determination may be challenging to perform. In addition, raw data makes it challenging to visualize what the data is showing.

Limitations of Descriptive Statistics

Descriptive statistics are so small that only the individuals or items you have calculated are summed up. The data you have obtained cannot be used to generalize to others or objects. When testing a drug to beat cancer in your patients, for example, you cannot say it would operate only based on descriptive statistics in other cancer patients.

Examples for Descriptive Statistics

- Take a simple number to sum up how well a batter does in baseball, the average batting. This figure is just the number of hits divided by the number of times at bat.
- You want to study the popularity of different leisure activities by gender. You distribute a survey and ask participants how many times they did each of the following in the past year: Go to a library, Watch a movie at a theatre, Visit a nationalpark. Your data set is the collection of responses to the survey. Now you can use descriptive statistics to find out the overall frequency of each activity (distribution), the averages for each activity (central tendency), and the spread of responses for each activity (variability).
- There are 100 students enrolled for a particular module. To find the overall performance of the students taking the respective module and the distribution of the marks, descriptive statistics must be used. Getting the marks as raw data would prove the determination of the overall performance and the distribution of the marks to be challenging.

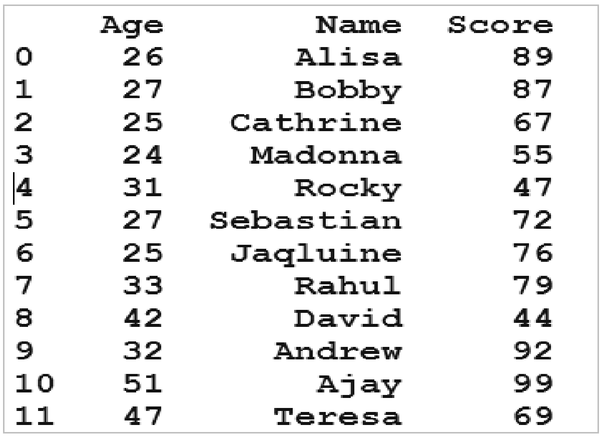
Descriptive Statistics in Python

It can be done in python using Pandas Describe() function. Describe Function gives the mean, std and IQR values. Generally describe() function excludes the character columns and gives summary statistics of numeric columns. We need to add a variable named include=‟all‟ to get the summary statistics or descriptive statistics of both numeric and character column.

Let‟s see with an example

```
# Program 4.1: Creation of Data Frame

Import pandas as pd

Import numpy as np

# Create a Dictionary of Series

d = {'Name':pd.Series(['Alisa', 'Bobby', 'Cathrine', 'Madonna', 'Rocky', 'Sebastian', 'Jaqluine', „Rahul', 'David', 'Andrew', 'Ajay', 'Teresa']), 'Age':pd.Series([26, 27, 25, 24, 31, 27, 25, 33, 42, 32, 51, 47]), 'Score':pd.Series([89,87,67,55,47,72,76,79,44,92,99,69])}

# Create a Data Frame

df = pd.DataFrame(d)

print df
```

Output:



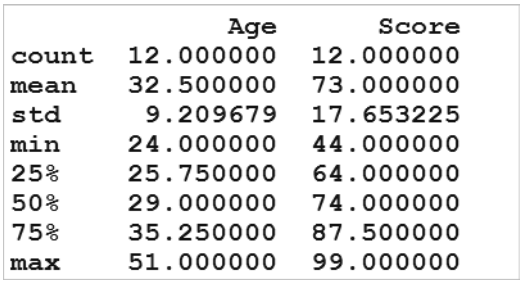
```
     Age        Name  Score
0     26       Alisa     89
1     27       Bobby     87
2     25    Cathrine     67
3     24     Madonna     55
4     31       Rocky     47
5     27   Sebastian     72
6     25    Jaqluine     76
7     33       Rahul     79
8     42       David     44
9     32      Andrew     92
10    51        Ajay     99
11    47      Teresa     69
```

Describe() Function gives the mean, std and IQR values. It excludes character column and calculate summary statistics only for numeric columns.

If we write the following statement:

Print df.describe()

The output will be:



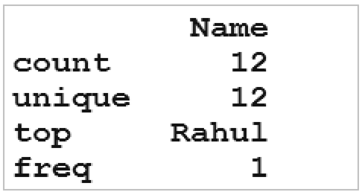
```
             Age      Score
count  12.000000  12.000000
mean   32.500000  73.000000
std     9.209679  17.653225
min    24.000000  44.000000
25%    25.750000  64.000000
50%    29.000000  74.000000
75%    35.250000  87.500000
max    51.000000  99.000000
```

describe() Function with an argument named include along with value object i.e include=‟object‟ gives the summary statistics of the character columns.

If we write the following statement:

Print df.decribe(include=[„object‟])

Then the output will be:



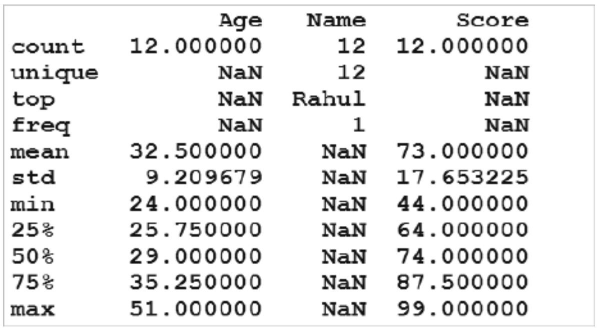
```
         Name
count      12
unique     12
top     Rahul
freq        1
```

describe() Function with include=‟all‟ gives the summary statistics of all the columns.

If we write the following statement:

Print df.decribe(include=‟all‟)

Then the output will be:



```
              Age   Name      Score
count   12.000000     12  12.000000
unique        NaN     12        NaN
top           NaN  Rahul        NaN
freq          NaN      1        NaN
mean    32.500000    NaN  73.000000
std      9.209679    NaN  17.653225
min     24.000000    NaN  44.000000
25%     25.750000    NaN  64.000000
50%     29.000000    NaN  74.000000
75%     35.250000    NaN  87.500000
max     51.000000    NaN  99.000000
```

However, if we don‟t require the output for all the functions then we can use a specific function for the output purpose.

Sr.No.	Function	Description
1	count()	Number of non-null observations
2	sum()	Sum of values
3	mean()	Mean of Values
4	median()	Median of Values

5	mode()	Mode of values
6	std()	Standard Deviation of the Values
7	min()	Minimum Value
8	max()	Maximum Value
9	abs()	Absolute Value
10	prod()	Product of Values
11	cumsum()	Cumulative Sum
12	cumprod()	Cumulative Product

For example, if we wish to find standard deviation, then we can write

```
# Program 4.2: To find standard deviation
import pandas as pd
import numpy as np
#Create a Dictionary of series
d = {'Name':pd.Series(['Tom', 'James', 'Ricky', 'Vin', 'Steve', 'Smith', 'Jack',     'Lee', 'David',
'Gasper',    'Betina',    'Andres']),    'Age':pd.Series([25,26,25,23,30,29,23,34,40,30,51,46]),
'Rating':pd.Series([4.23,3.24,3.98,2.56,3.20,4.6,3.8,3.78,2.98,4.80,4.10,3.65]) }
#Create  a  DataFrame
df = pd.DataFrame(d)
print df.std()


Output :
Age       9.232682
Rating    0.661628
dtype: float64
```

4.2.2 Exploratory Data Analysis

Exploratory data analysis (EDA) is used by data scientists to analyse and investigate data sets and summarize their main characteristics, often employing data visualization methods. It helps determine how best to manipulate data sources to get the answers you need, making it easier for data scientists to discover patterns, spot anomalies, test a hypothesis, or check assumptions.

EDA is primarily used to see what data can reveal beyond the formal modelling or hypothesis testing task and provides a provides a better understanding of data set variables and the relationships between them. It can also help determine if the statistical techniques you are considering for data analysis are appropriate. Originally developed by American mathematician John Tukey in the 1970s, EDA techniques continue to be a widely used method in the data discovery process today.

Importance of Exploratory Data Analysis

The main purpose of EDA is to help look at data before making any assumptions. It can help identify obvious errors, as well as better understand patterns within the data, detect outliers or anomalous events, find interesting relations among the variables.

Data scientists can use exploratory analysis to ensure the results they produce are valid and applicable to any desired business outcomes and goals. EDA also helps stakeholders by confirming they are asking the right questions. EDA can help answer questions about standard deviations, categorical variables, and confidence intervals. Once EDA is complete and insights are drawn, its features can then be used for more sophisticated data analysis or modelling, including machine learning.

Exploratory data analysis tools

Specific statistical functions and techniques you can perform with EDA tools include:

- Clustering and dimension reduction techniques, which help create graphical displays of high-dimensional data containing many variables.
- Univariate visualization of each field in the raw dataset, with summary statistics.
- Bivariate visualizations and summary statistics that allow you to assess the relationship between each variable in the dataset and the target variable you‟re looking at.
- Multivariate visualizations, for mapping and understanding interactions between different fields in the data.
- K-means Clustering is a clustering method in unsupervised learning where data points are assigned into K groups, i.e. the number of clusters, based on the distance from each group‟s centroid. The data points closest to a particular centroid will be clusteredunder the same category. K-means Clustering is commonly used in market segmentation, pattern recognition, and image compression.
- Predictive models, such as linear regression, use statistics and data to predict outcomes.

Types of Exploratory Data Analysis

There are four primary types of EDA:

- Univariate non-graphical. This is simplest form of data analysis, where the data being analysed consists of just one variable. Since it‟s a single variable, it doesn‟t deal with causes or relationships. The main purpose of univariate analysis is to describe the data and find patterns that exist within it.

- Univariate graphical. Non-graphical methods don"t provide a full picture of the data. Graphical methods are therefore required. Common types of univariate graphics include: (i) Stem-and-leaf plots, which show all data values and the shape of the distribution. (ii) Histograms, a bar plot in which each bar represents the frequency (count) or proportion (count/total count) of cases for a range of values. (iii) Box plots, which graphically depict the five-number summary of minimum, first quartile, median, third quartile, and maximum.
- Multivariate non-graphical: Multivariate data arises from more than one variable. Multivariate non-graphical EDA techniques generally show the relationship between two or more variables of the data through cross-tabulation or statistics.
- Multivariate graphical: Multivariate data uses graphics to display relationships between two or more sets of data. The most used graphic is a grouped bar plot or bar chart with each group representing one level of one of the variables and each bar within a group representing the levels of the other variable. Other common types of multivariate graphics include: (i) Scatter plot, which is used to plot data points on a horizontal and a vertical axis to show how much one variable is affected by another.
  (ii) Multivariate chart, which is a graphical representation of the relationships between factors and a response. (iii) Run chart, which is a line graph of data plotted over time. (iv) Bubble chart, which is a data visualization that displays multiple circles (bubbles) in a two-dimensional plot. (v) Heat map, which is a graphical representation of data where values are depicted by color.

Exploratory Data Analysis Tools

Some of the most common data science tools used to create an EDA include:

Python: An interpreted, object-oriented programming language with dynamic semantics. Its high-level, built-in data structures, combined with dynamic typing and dynamic binding, make it very attractive for rapid application development, as well as for use as a scripting or glue language to connect existing components together. Python and EDA can be used together to identify missing values in a data set, which is important so you can decide how to handle missing values for machine learning.

R: An open-source programming language and free software environment for statistical computing and graphics supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians in data science in developing statistical observations and data analysis.

Steps in Exploratory Data Analysis in Python

There are many steps for conducting Exploratory data analysis.

- Description of data
- Handling missing data
- Handling outliers
- Understanding relationships and new insights through plots

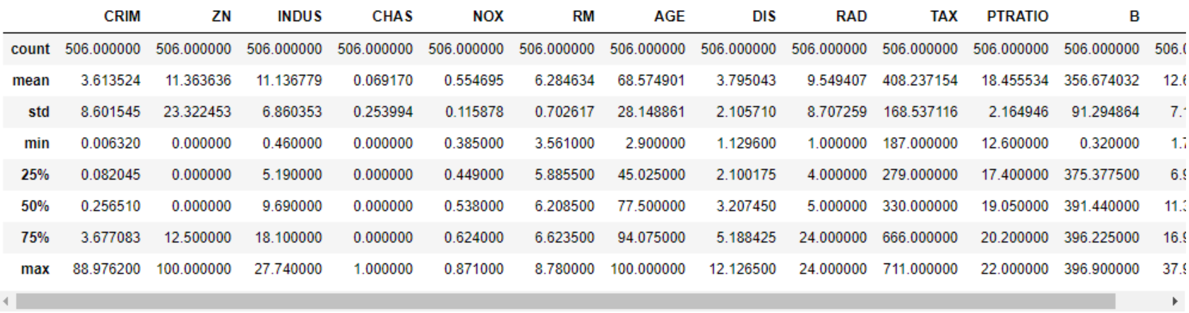
Description of data: We will be using the Boston Data Set for our examples, which can be imported from sklearn.datasets import load_boston.

We need to know the different kinds of data and other statistics of our data before we can move on to the other steps. A good one is to start with the describe() function in python.In Pandas, we can apply describe() on a DataFrame which helps in generating descriptive statistics that summarize the central tendency, dispersion, and shape of a dataset‟s distribution, excluding NaN values.

The result‟s index will include count, mean, std, min, max as well as lower, 50 and upper percentiles. By default, the lower percentile is 25 and the upper percentile is 75. The 50 percentile is the same as the median.

```
# Program 4.3: Loading the Dataset

import pandas as pd

from sklearn.datasets import load_boston

boston = load_boston()

x = boston.data

y = boston.target

columns = boston.feature_names

# creating dataframes

boston_df = pd.DataFrame(boston.data)

boston_df.columns = columns

boston_df.describe()
```

Output:



```
             CRIM          ZN       INDUS        CHAS         NOX          RM         AGE         DIS         RAD         TAX     PTRATIO           B
count  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  506.000000  506.0
 mean    3.613524   11.363636   11.136779    0.069170    0.554695    6.284634   68.574901    3.795043    9.549407  408.237154   18.455534  356.674032   12.6
  std    8.601545   23.322453    6.860353    0.253994    0.115878    0.702617   28.148861    2.105710    8.707259  168.537116    2.164946   91.294864    7.1
  min    0.006320    0.000000    0.460000    0.000000    0.385000    3.561000    2.900000    1.129600    1.000000  187.000000   12.600000    0.320000    1.7
  25%    0.082045    0.000000    5.190000    0.000000    0.449000    5.885500   45.025000    2.100175    4.000000  279.000000   17.400000  375.377500    6.9
  50%    0.256510    0.000000    9.690000    0.000000    0.538000    6.208500   77.500000    3.207450    5.000000  330.000000   19.050000  391.440000   11.3
  75%    3.677083   12.500000   18.100000    0.000000    0.624000    6.623500   94.075000    5.188425   24.000000  666.000000   20.200000  396.225000   16.9
  max   88.976200  100.000000   27.740000    1.000000    0.871000    8.780000  100.000000   12.126500   24.000000  711.000000   22.000000  396.900000   37.9
```

Handling missing data: Data in the real-world are rarely clean and homogeneous. Data can either be missing during data extraction or collection due to several reasons. Missing values need to be handled carefully because they reduce the quality of any of our performance matrix. It can also lead to wrong prediction or classification and can also cause a high bias forany given model being used. There are several options for handling missing values. However,the choice of what should be done is largely dependent on the nature of our data and the missing values. Some of the techniques for this are (i) Drop NULL or missing values, (ii) Fill Missing Values, and (iii) Predict Missing values with an ML Algorithm.

The dropping of NULL values is the fastest and easiest step to handle missing values. However, it is not generally advised. This method reduces the quality of our model as it reduces sample size because it works by deleting all other observations where any of the variables is missing. It can be achieved by using dropna() function as shown below:

```
Boston_df.shpae

Output:

(506,13)
```

```
Boston_df = boston_df.dropna()

Boston_df.shape

Output:

(506,13)
```

After implementing the above code, it will indicate that there are no null values in our data set.

Another way of handling missing values is by filling missing values. This is a process whereby missing values are replaced with a test statistic like mean, median or mode of the particular feature the missing value belongs to. Let‟s suppose we have a missing value of agein the boston data set. Then the below code will fill the missing value with the 30.

```
Boston_df[„AGE‟] = boston_df[„AGE‟].fillna(30)

Boston_df.shpae

Output:

(506,13)
```

The third way is via predicting missing values with an ML Algorithm. This is by far one of the best and most efficient methods for handling missing data. Depending on the class of datathat is missing, one can either use a regression or classification model to predict missing data.

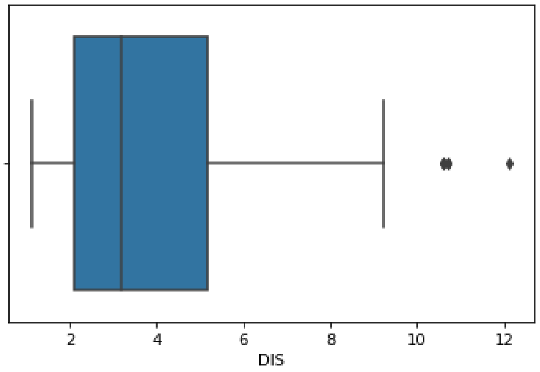
Handling outliers: An outlier is something which is separate or different from the crowd. Outliers can be a result of a mistake during data collection or it can be just an indication of variance in your data. Some of the methods for detecting and handling outliers are (i) BoxPlot (ii) Scatterplot (iii) Z-score and (iv) IQR(Inter-Quartile Range).

BoxPlot: A box plot is a method for graphically depicting groups of numerical data through their quartiles. The box extends from the Q1 to Q3 quartile values of the data, with a line at the median (Q2). The whiskers extend from the edges of the box to show the range of the data. Outlier points are those past the end of the whiskers. Boxplots show robust measures of location and spread as well as providing information about symmetry and outliers.

```
# Program 4.4: BoxPlot
import seaborn as sns
sns.boxplot(x=boston_df['DIS'])
```

Output:

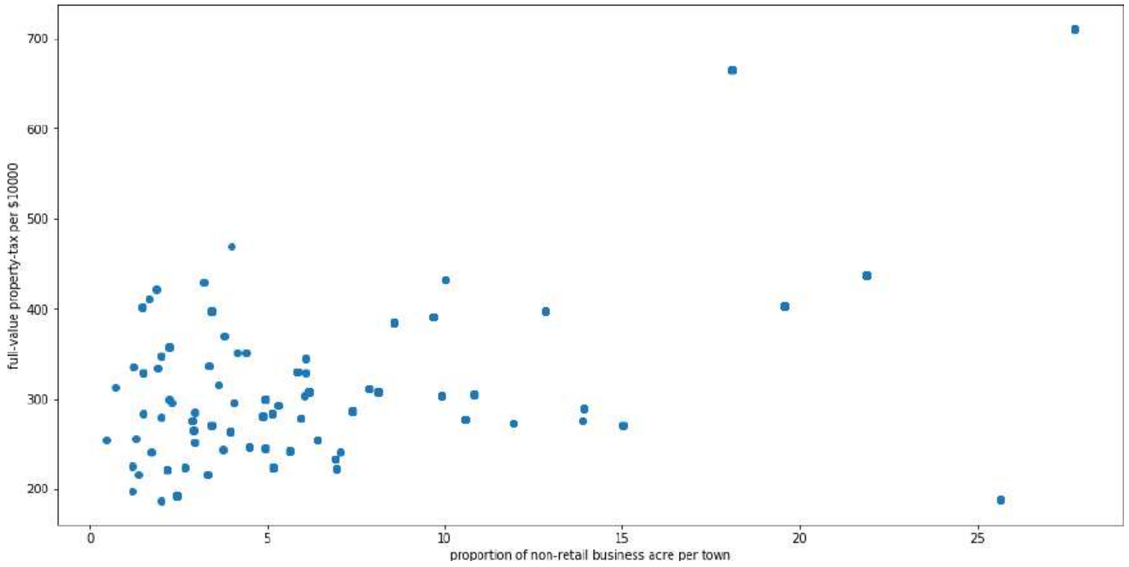
```
Out[11]: <matplotlib.axes._subplots.AxesSubplot at 0x1525a92b240>
```



Scatterplot: A scatter plot is a mathematical diagram using Cartesian coordinates to display values for two variables for a set of data. The data are displayed as a collection of points, each having the value of one variable determining the position on the horizontal axis and the value of the other variable determining the position on the vertical axis. The points that arefar from the population can be termed as an outlier.

```
# Program 4.5: Scatter Plot
import matplotlib.pyplot as plt
fig, ax = plt.subplots(figsize=(16,8))
ax.scatter(boston_df['INDUS'] , boston_df['TAX'])
ax.set_xlabel('proportion of non-retail business acre per town')
ax.set_ylabel('full-value property-tax per $10000')
plt.show()
```

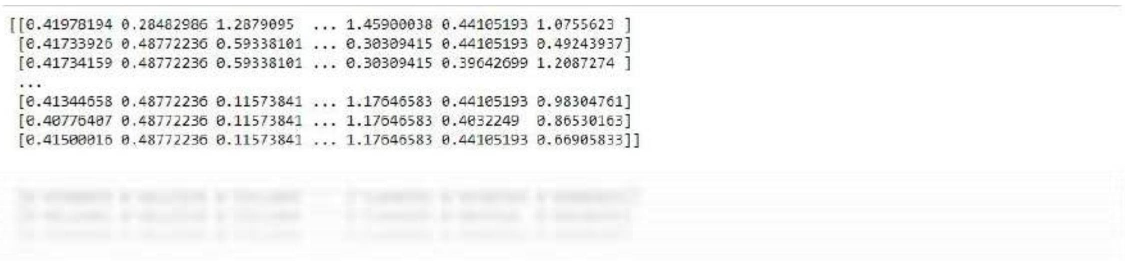
Output:



Z-score: The Z-score is the signed number of standard deviations by which the value of an observation or data point is above the mean value of what is being observed or measured. While calculating the Z-score we re-scale and center the data and look for data points that are too far from zero. These data points which are way too far from zero will be treated as the outliers. In most of the cases a threshold of 3 or -3 is used i.e. if the Z-score value is greater than or less than 3 or -3 respectively, that data point will be identified as outliers.

```
# Program 4.6: Z-Score

From scipy import stats

Import numpy as np

Z = np.abs(stats.zscore(boston_df))

Print(Z)
```

Output:



```
[[0.41978194 0.28482986 1.2879095  ... 1.45900038 0.44105193 1.0755623 ]
 [0.41733926 0.48772236 0.59338101 ... 0.30309415 0.44105193 0.49243937]
 [0.41734159 0.48772236 0.59338101 ... 0.30309415 0.39642699 1.2087274 ]
 ...
 [0.41344658 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.98304761]
 [0.40776407 0.48772236 0.11573841 ... 1.17646583 0.4032249  0.86530163]
 [0.41500016 0.48772236 0.11573841 ... 1.17646583 0.44105193 0.66905833]]
```

For removing outliers, following statements may be used:

```
Boston_df_outlier_Zscore = boston_df[(z<3).all(axis=1)]

Boston_df_outlier_Zscore.shape
```

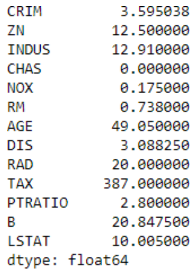
```
Output:
(415,13)
```

We can see from the above code that the shape changes, which indicates that our dataset has some outliers.

IQR: The interquartile range (IQR) is a measure of statistical dispersion, being equal to the difference between 75th and 25th percentiles, or between upper and lower quartiles.

IQR = Q3 − Q1 and can be found using following statements:

```
Q1  = boston_df.quantile(0.25)
Q3 = boston_df_quantile(0.75)
IQR = Q3 – Q1
Print(IQR)


Output:
```



```
CRIM          3.595038
ZN           12.500000
INDUS        12.910000
CHAS          0.000000
NOX           0.175000
RM            0.738000
AGE          49.050000
DIS           3.088250
RAD          20.000000
TAX         387.000000
PTRATIO       2.800000
B            20.847500
LSTAT        10.005000
dtype: float64
```

Once we have IQR scores below code will remove all the outliers in our dataset.

```
Boston_df_outlier-IQR= boston_df[~((boston_df < (Q1 – 1.5 * IQR)) | (boston_df > (Q3 + 1.5 * IQR))).any(axis=1)]
Boston_df_outlier_IQR.shape
Output:
(274, 13)
```

4.2.3 Data Visualization

Data visualization is a graphical representation of quantitative information and data by using visual elements like graphs, charts, and maps. Data visualization convert large and small data sets into visuals, which is easy to understand and process for humans. Data visualization tools provide accessible ways to understand outliers, patterns, and trends in the data.

In the world of Big Data, the data visualization tools and technologies are required to analyse vast amounts of information. Data visualizations are common in our everyday life, but they always appear in the form of graphs and charts.

Data visualizations are used to discover unknown facts and trends. We can see visualizations in the form of line charts to display change over time. Bar and column charts are useful for observing relationships and making comparisons. A pie chart is a great way to show parts-of-a-whole and maps are the best way to share geographical data visually.

Today's data visualization tools go beyond the charts and graphs used in the Microsoft Excel spreadsheet, which displays the data in more sophisticated ways such as dials and gauges, geographic maps, heat maps, pie chart, and fever chart.

Effective Data Visualization

American Statistician and Yale Professor Edward Tufte saya useful data visualizations consist of complex ideas communicated with clarity, precision, and efficiency.

To craft an effective data visualization, we need to start with clean data that is well-sourced and complete. Once the data is ready to visualize, we need to pick the right chart to visualize. After that, we need to design and customize our visualization according to requirements. Simplicity is essential as we don't want to add any elements that distract from the data.

Importance of Data Visualization

- Data visualization is important because of the processing of information in human brains. Using graphs and charts to visualize a large amount of the complex data sets is more comfortable in comparison to studying the spreadsheet and reports.
- Data visualization is an easy and quick way to convey concepts universally. We can experiment with a different outline by making a slight adjustment.
- Data visualization can identify areas that need improvement or modifications.
- Data visualization can clarify which factor influence customer behavior.
- Data visualization helps you to understand which products to place where.
- Data visualization can predict sales volumes.

Uses of Data Visualization

- To make easier in understand and remember.
- To discover unknown facts, outliers, and trends.
- To visualize relationships and patterns quickly.
- To ask a better question and make better decisions.
- To competitive analyse.
- To improve insights.

Types of Data Visualizations

The earliest form of data visualization can be traced back the Egyptians in the pre-17th century, largely used to assist in navigation. As time progressed, people leveraged data visualizations for broader applications, such as in economic, social, health disciplines. Perhaps most notably, Edward Tufte published “The Visual Display of Quantitative Information” , which illustrated that individuals could utilize data visualization to present data in a more effective manner. His book continues to stand the test of time, especially as companies turn to dashboards to report their performance metrics in real-time. Dashboards are effective data visualization tools for tracking and visualizing data from multiple data sources, providing visibility into the effects of specific behaviors by a team or an adjacent one on performance. Dashboards include common visualization techniques, such as:

Tables: This consists of rows and columns used to compare variables. Tables can show a great deal of information in a structured way, but they can also overwhelm users that are simply looking for high-level trends.

Pie charts and Stacked Bar Charts: These graphs are divided into sections that represent parts of a whole. They provide a simple way to organize data and compare the size of each component to one other.

Line graphs and Area charts: These visuals show change in one or more quantities by plotting a series of data points over time. Line graphs utilize lines to demonstrate these changes while area charts connect data points with line segments, stacking variables on top ofone another and using color to distinguish between variables.

Histograms: This graph plots a distribution of numbers using a bar chart (with no spaces between the bars), representing the quantity of data that falls within a particular range. This visual makes it easy for an end user to identify outliers within a given dataset.

Scatter plots: These visuals are beneficial in revealing the relationship between two variables, and they are commonly used within regression data analysis. However, these can sometimes be confused with bubble charts, which are used to visualize three variables via the x-axis, the y-axis, and the size of the bubble.

Heat maps: These graphical displays are helpful in visualizing behavioral data by location. This can be a location on a map, or even a webpage.

Tree maps: These display hierarchical data as a set of nested shapes, typically rectangles. Treemaps are great for comparing the proportions between categories via their area size.

Data Visualization in Python

Data visualization in python is perhaps one of the most utilized features for data science with python in today‟s day and age. The libraries in python come with lots of different features that enable users to make highly customized, elegant, and interactive plots.

Useful packages for visualizations in python are Matplotlib, Seaborn, Statistics, Pandas and Numpy.

Matplotlib: Matplotlib is a visualization library in Python for 2D plots of arrays. Matplotlib is written in Python and makes use of the NumPy library. It can be used in Python and IPython shells, Jupyter notebook, and web application servers. Matplotlib comes with a wide variety of plots like line, bar, scatter, histogram, etc., which can help us, into understanding trends, patterns, correlations.

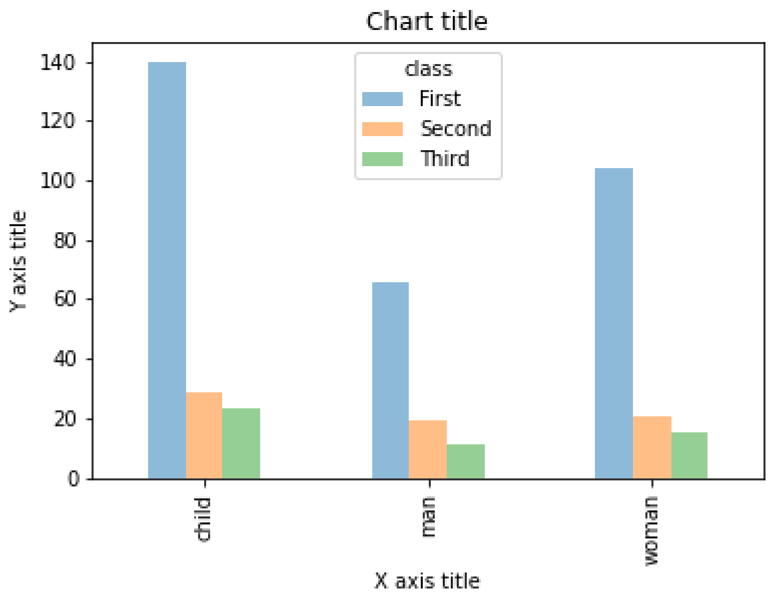
Seaborn: Seaborn is a dataset-oriented library for making statistical representations in Python. It is developed atop matplotlib and to create different visualizations. It is integrated with pandas data structures. The library internally performs the required mapping and aggregation to create informative visuals.

Grouped bar chart: A grouped bar chart is used when we want to compare the values in certain groups and sub-groups

Grouped bar chart using Matplotlib: In the following code a data set named as Titanic has been used which can easily be downloaded from Kaggle.

```
# Program 4.7: Grouped Bar Chart using Matplotlib
#Creating the dataset
import seaborn as sns
import matplotlib.pyplot as plt
df = sns.load_dataset('titanic')
df_pivot = pd.pivot_table(df, values="fare", index="who", columns="class", aggfunc=np.mean)
#Creating a grouped bar chart
ax = df_pivot.plot(kind="bar", alpha=0.5)
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

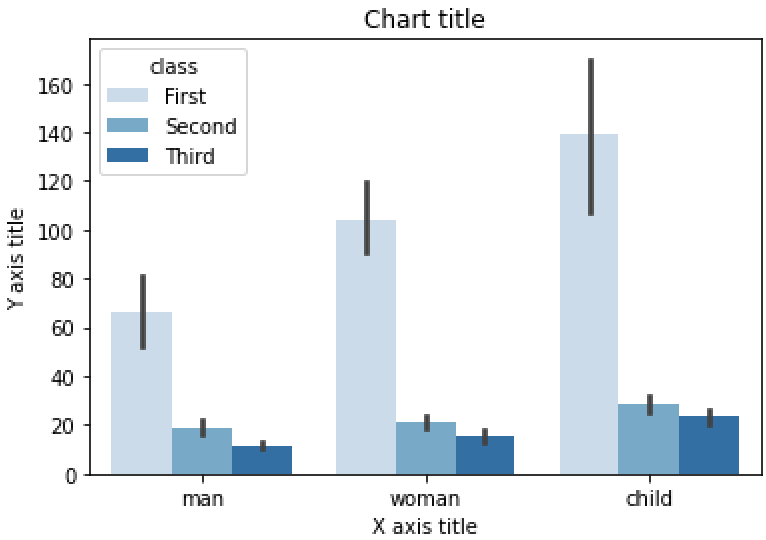
Output:



Grouped bar chart using Seaborn:

```
# Program 4.8: Grouped bar chart using seaborn
#Reading the dataset
import seaborn as sns
import matplotlib.pyplot as plt
titanic_dataset = sns.load_dataset('titanic')
#Creating the bar plot grouped across classes
sns.barplot(x = 'who', y = 'fare', hue = 'class', data = titanic_dataset, palette = "Blues")
#Adding the aesthetics
plt.title('Chart title')
plt.xlabel('X axis title')
plt.ylabel('Y axis title')
# Show the plot
plt.show()
```

Output:



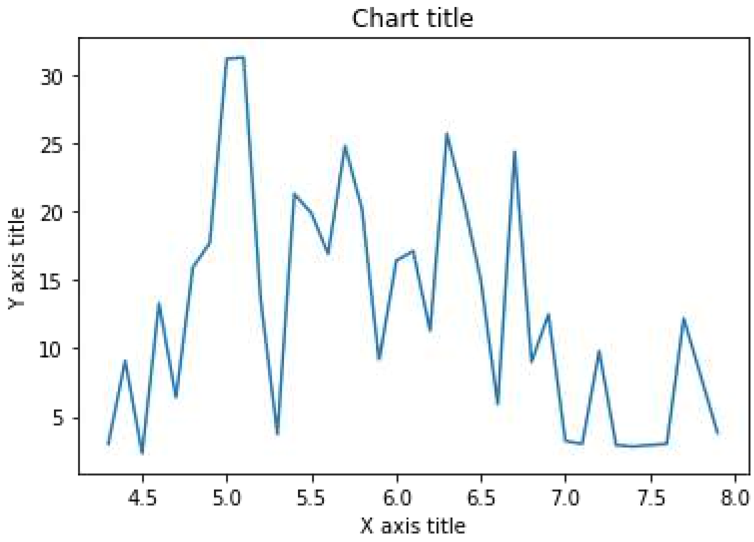
Line chart: A line chart is used for the representation of continuous data points. This visual can be effectively utilized when we want to understand the trend across time.

In the following example “iris” dataset has been used which can be downloaded from Kaggle.

```
#Program 4.9: Line Chart
#Creating the dataset
import seaborn as sns
import matplotlib.pyplot as plt
```

```
df = sns.load_dataset("iris")

df=df.groupby('sepal_length')['sepal_width'].sum().to_frame().reset_index()

#Creating the line chart

plt.plot(df['sepal_length'], df['sepal_width'])

#Adding the aesthetics

plt.title('Chart title')

plt.xlabel('X axis title')

plt.ylabel('Y axis title')

#Show the plot

plt.show()
```

Output:

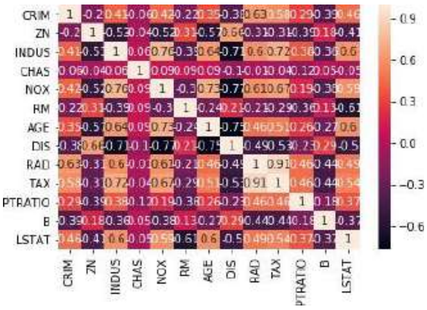


HeatMaps: The Heat Map procedure shows the distribution of a quantitative variable over all combinations of 2 categorical factors. If one of the 2 factors represents time, then the evolution of the variable can be easily viewed using the map. A gradient color scale is used to represent the values of the quantitative variable. The correlation between two random variables is a number that runs from -1 through 0 to +1 and indicates a strong inverse relationship, no relationship, and a strong direct relationship, respectively.

It can be created using following statement using boston data set as used in the earlier section

```
Corr_mat = boston_df.corr().round(2)

Sns.heatmap(data=corr_mat, annot=True)
```

Output:



Other types of graphs have already been discussed in the previous modules.

4.3 SUMMARY

In this module, three important topics have been discussed. The topics discussed in this module are descriptive statistics, exploratory data analysis and data visualization. All thethree topic are related to each other. Starting with descriptive statistics involves the steps to analyse the data using the measures discussed in previous two modules i.e. module-II and module-III. The descriptive statistics explain the measures of central tendency, measure of dispersion along with the frequency distribution used for the given data. Following this exploratory data analysis involves the various steps of data analysis such as identifying data sources, cleaning, presenting and interpreting. When we talk about data presenting, data visualization comes into picture. Data visualization involves the presentation of data using suitable means i.e. graphs so that it becomes easy to understand and interpret. The implementation of these topics have been explained using python.

4.4 PRACTICE QUESTIONS

Q.1 What is meant by descriptive statistics? Describe the purpose of descriptive statistics.

Q.2 Explain various types of descriptive statistics in detail.

Q.3 Describe various functions in python for performing descriptive statistics.

Q.4 What do you mean by exploratory data analysis. Explain its significance.

Q.5 Write and explain the various steps for performing exploratory data analysis.

Q.6 What are the various packages in python that can be used for performing exploratory data analysis? Explain in detail.

Q.7 How can you draw (i) histogram (ii) bar chart (iii) heat map (iv) grouped bar chart (v) pie chart, in python? Explain by writing suitable programs.

Q.8 What is meant by data visualization? Explain the various steps in data visualization.

Q.9 Discuss various tools used for data visualization.

REFERENCES

1. A. Abebe, J. Daniels, J.W.Mckean, “Statistics and Data Analysis”.
2. A. Martelli, A. Ravenscroft, S. Holden, “Python in a Nutshell”, OREILLY.

3. Clarke, G.M. & Cooke, D., “A Basic course in Statistics”, Arnold.
4. David M. Lane, “Introduction to Statistics”.
5. Eric Matthes, “Python Crash Course: A Hands-On, Project-Based Introduction to Programming”.\
6. S.C.Gupta and V.K.Kapoor, “Fundamentals of Mathematical Statistics”, Sultan Chand & Sons, New Delhi.
7. Weiss, N.A., “Introductory Statistics”, Addison Wesley

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

UNIT V: CORRELATION AND REGRESSION

STRUCTURE

5.0 OBJECTIVES

The main goal of this module is to help students learn, understand and practice the basics of statistics which will helpful to do the research in the social sciences. In this module you will learn the basics of statistics which covers two fundamentals concepts correlation and regression. The examples of this module were calculated manual as well as using programming language. Python language is used in this module.

5.1 INTRODUCTION

Data science become a buzzword that everyone talks about the data science. Data science is an interdisciplinary field that combines different domain expertise, computer programming skills, mathematics and statistical knowledge to find or extract the meaningful or unknown patterns from unstructured and structure dataset.

Data science is useful for extraction, preparation, analysis and visualization of data. Various statistical methods can be applied to get insight in the data.

Data science is all about using data to solve problems. Data has become the fuel of industries. It is most demandable field of 21st century. Every industry require data to functioning, searching, marketing, growing, expanding their business.

The application of areas of data science are health care, fraud detection, disease predicting, real time shipping routes, speech recognition, targeting advertising, gaming and many more.

5.2 CORRELATION

Correlation is a statistical measure that expresses the extent to which two or more variables are changes together at a constant rate. It is a relationship between two or more variables. Thedata can be represented by the pairs (x, y) where x is an independent variable and y is a dependent variable.

Correlations are useful for describing simple relationships among the data. It quantifies the degree and direction to which two variables are related. It is a measure of the extent to which two variables are related.

5.3 TYPES of CORRELATION

The different types of correlations on the basis of: the degree of correlation, numbers of variables used and linearity.

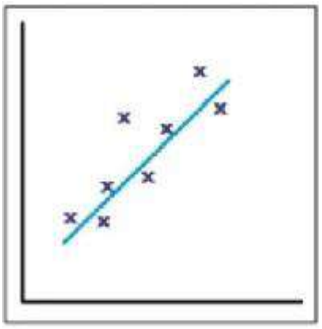
5.3.1 Positive, Negative and Zero Correlation

The positive, negative and zero correlations are basis on the degree of correlation.

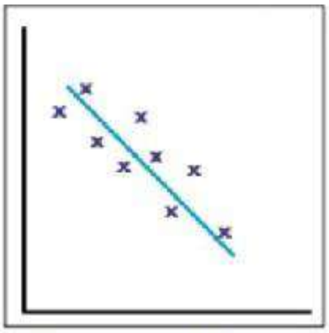
- **Positive Correlation** – A relationship between two variables in which both the variables move in same direction, it means that when one variable is increases as the other variable increases, or one variable is decreases while the other decreases. The examples of positive correlation are: height and weight of person, price and supply of a commodity, etc.
- **Negative Correlation** – A relationship between two variables in which both the variable moves in opposite direction. That means when one variable is increase as the other variable decreases, or one variable decrease while the other increases.

The examples of negative correlation are: no. of absent and grade of student, speed of train and time to reach destination, etc.

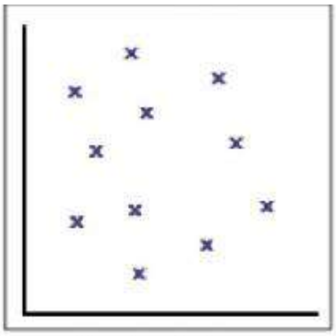
- **No / Zero Correlation** – There is no linear dependence or no relationship between two variables. The example of no / zero correlation is the coffee drunk and level of IQ.



Positive Correlation



Negative Correlation



No Correlation

5.3.2 Simple, Multiple and Partial Correlation

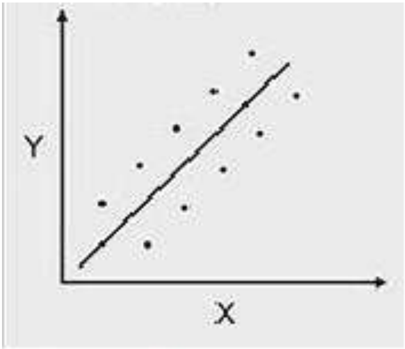
The simple, multiple and partial correlations are depending on the number of variables studied in the analysis.

- **Simple Correlation** – There are only two variables are studied and check the correlation between them is called simple correlation. Examples of simple correlation are age and height of students, price and demand of commodities.
- **Multiple Correlation** – There are three or more variables are studied for correlation simultaneously is called multiple correlation. Example of multiple correlation is to study the relationship between the yield of any crop, amount of fertilizers used and amount of rainfall.
- **Partial Correlation** – There are three or more variables are studied. When one or more variables are kept constant and the relationship is studied between others is called partial correlation. Example of partial correlation is the price of ice-cream, temperature and demand. If we kept the price of ice-cream is constant and studied the correlation between temperature and demand of ice-cream.

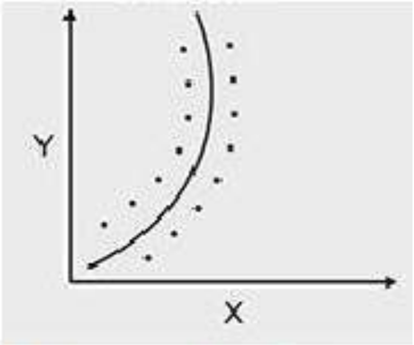
5.3.3 Linear and Non-Linear Correlation

The linear and non-linear correlation are on the basis of linearity of data.

- **Linear Correlation** – The correlation is said to be a linear if the ratio of change of the two variables is constant. If we plot all the points on the scatter diagramtend to lie near a line which like a straight line.
- **Non-Linear Correlation** – The correlation is said to be a non-linear or curvilinear if the ratio of change of the two variables is not constant. If we plot all the points on the scatter diagram tend to lie near a smooth curve which like a curve.



Linear Correlation



Non-Linear Correlation

The **corr()** function is used to find the correlation between two variables in Python. Here we take an example of boys age and weight to calculate the correlation.

Example: Find the correlation of boys age and weight.

Age	10	20	30	40	50	60	70	80
Weight	32	48	59	64	70	67	78	82

The below code will find the correlation of boys age and weight.

```
# Importing library
import pandas as pd

# Data of Age and Weight
Age = pd.Series([10, 20, 30, 40, 50, 60, 70, 80])
Weight = pd.Series([32, 48, 59, 64, 70, 67, 78, 82])

#Calculating Correlation
correlation = Age.corr(Weight)
correlation
```

The above code will calculate the correlation between age and weight of boys as follow:

```
0.9501687384314103
```

Example: Find the correlation of boys age and weight.

Age	10	20	30	40	50	60	70	80
Weight	32	48	59	64	70	67	78	82

The below code will find the correlation of boys age and weight.

```
# Importing library
import pandas as pd

# Data of Age and Weight
data = {
   "Age" : [10, 20, 30, 40, 50, 60, 70, 80],
   "Weight" : [32, 48, 59, 64, 70, 67, 78, 82]
            }

# Creation of Dataframe
df = pd.DataFrame(data)

# Creation of Correlation Matrix
df.corr()
```

The above code will calculate the correlation between age and weight of boys as follow:

	Age	**Weight**
Age	1.000000	0.950169
Weight	0.950169	1.000000

Now we take an example of three subjects mark of science student to calculate the correlation.

Example: Find the correlation between marks of three subjects maths, chemistry and physics of science students.

Maths	100	86	90	80	96	95	92	99
Chemistry	88	90	80	90	90	86	94	88
Physics	92	88	89	94	90	87	93	91

The below code will find the correlation between marks of three subjects such as maths, chemistry and physics of science students.

```
# Importing library
import pandas as pd

data = {
   "Maths" : [100, 86, 90, 80, 96, 95, 92, 99],
   "Chemistry" : [88, 90, 80, 90, 90, 86, 94, 88],
   "Physics" : [92, 88, 89, 94, 90, 87, 93, 91]
             }

df = pd.DataFrame(data)
```

```
df.corr()
```

The above code will calculate the correlation between maths, chemistry and physicssubject marks of science students as follow:

	Maths	Chemistry	Physics
Maths	1.000000	-0.096004	-0.180719
Chemistry	-0.096004	1.000000	0.502519
Physics	-0.180719	0.502519	1.000000

From the above table we can see that, the correlation between maths and chemistry is negative (-0.096004), between maths and physics is also negative (-0.180719) and between chemistry and physics is positive (0.502519).

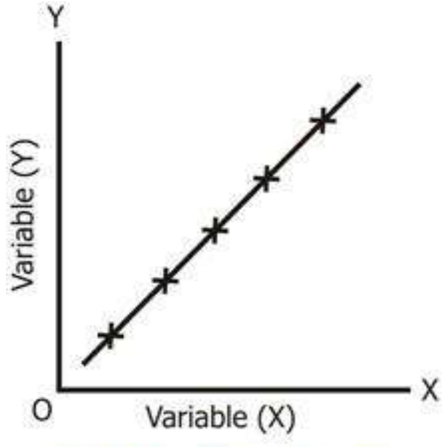
5.4 TECHNIQUES FOR MEASURING CORRELATION

The different techniques for measuring correlation are: graphical method and algebraic method. Scatter diagram is a graphical method. Karl Pearson's correlation coefficient and Spearman's rank correlation coefficient are algebraic methods.

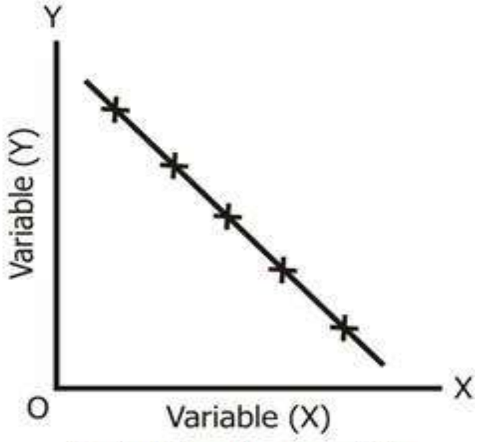
5.4.1 Scatter Plot

Scatter diagram is a graph in which the values of two variables are plotted along with two axes. It is a most basic type of plot that helps you visualize the relationship between two variables. Each value in this plot is represent by a dot. It is a set of dottedpoints to represent the data on both horizontal and vertical axis.

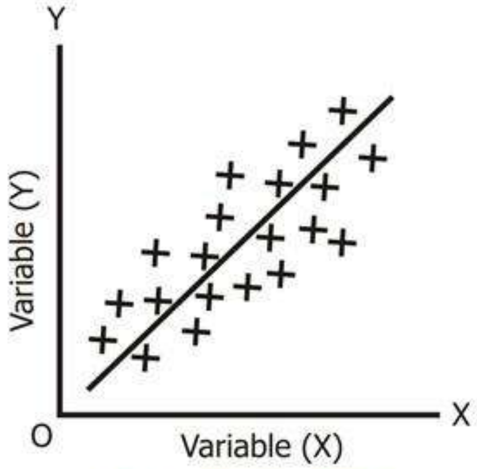
- **Perfect positive correlation :** All the plotted points are on a straight line which is rising from lower left corner to the upper right corner in a scatter diagram.
- **Perfect negative correlation :** All the plotted points are on a straight line which is rising from upper left corner to the lower right corner in a scatter diagram.
- **Strong positive correlation :** All the plotted points are closer to a straight line which is rising from lower left corner to the upper right corner in a scatterdiagram.
- **Strong negative correlation :** All the plotted points are closer to a straight line which is rising from upper left corner to the lower right corner in a scatterdiagram.
- **Weak positive correlation :** All the plotted points are not closer (lie away) to a straight line which is rising from lower left corner to the upper right corner in a scatter diagram.
- **Weak negative correlation :** All the plotted points are not closer (lie away) to a straight line which is rising from upper left corner to the lower right corner in a scatter diagram.
- **No correlation :** All the plotted points are scattered randomly across the graph.



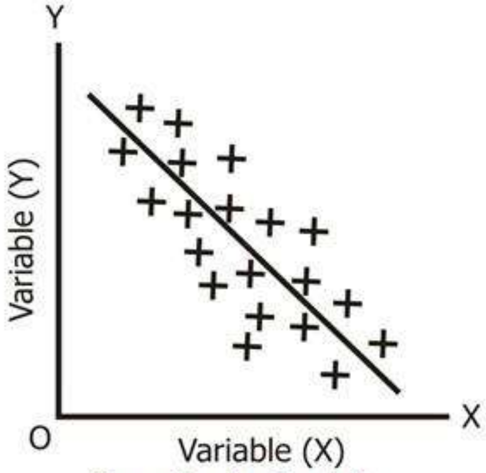
Perfect Positive Correlation



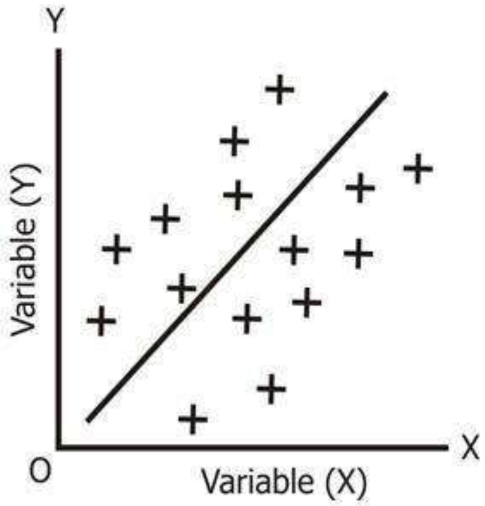
Perfect Negative Correlation



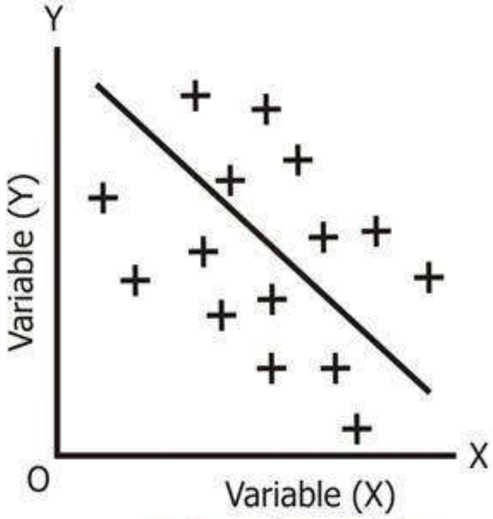
Strong Positive Correlation



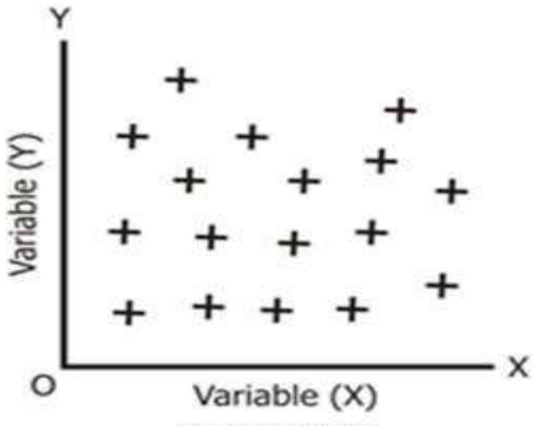
Strong Negative Correlation



Weak Positive Correlation



Weak Negative Correlation



No Correlation

The **scatter()** function is used to draw the scatter plot in Python. It plots one dot for each observation. It required two different set of observation with same length for both the axis.

Here we take an example of boys age and weight. The x-axis represents age and y-axis represents weight of boys.

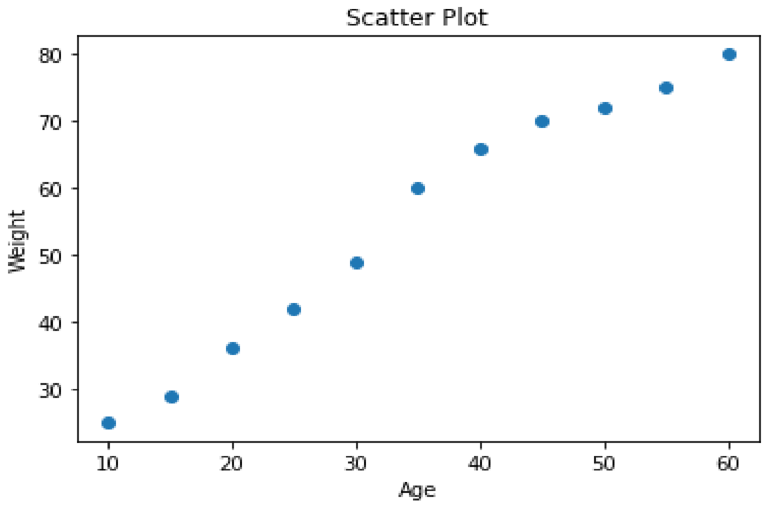
Example: Plot the scatter diagram of age and weight of boys.

Age	10	15	20	25	30	35	40	45	50	55	60
Weight	25	29	36	42	49	60	66	70	72	75	80

The below code will plot the scatter diagram of age and weight of boys.

```
# Importing library
import matplotlib.pyplot as plt

# Data of Age and Weight
Age = [10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60]
Weight = [25, 29, 36, 42, 49, 60, 66, 70, 72, 75, 80]

# Plotting scatter plot with title and label
plt.scatter(Age,  Weight)
plt.title("Scatter Plot")
plt.xlabel("Age")
plt.ylabel("Weight")

# Show plot
plt.show()
```

The above code will create scatter plot as follow:



In the above scatter plot we can see the relationship between two variables age and weight. There is a positive correlation between boys age and weight.

Another method is to draw scatter plot using ***"kind"*** parameter. Here we take an example of two subject marks. The x-axis represents chemistry and y-axis represents physics subject marks.

Example: Plot the scatter diagram of chemistry and physics subject marks.

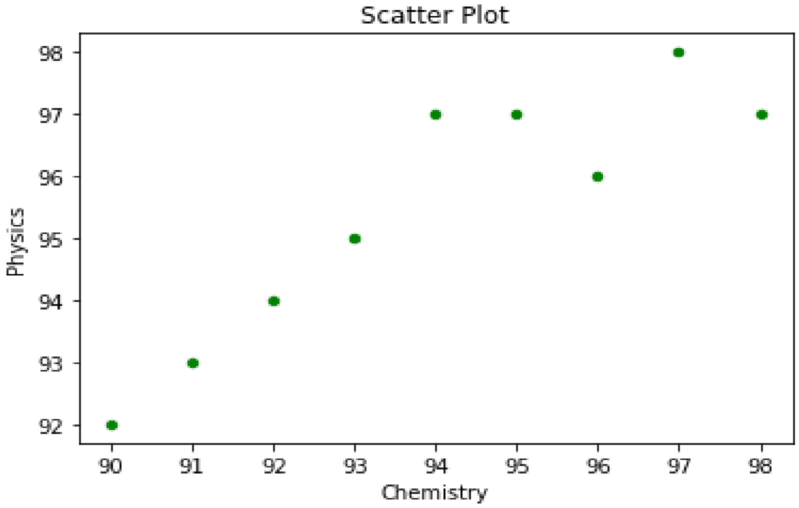
Chemistry	90	92	97	98	96	94	91	93	95
Physics	92	94	98	97	96	97	93	95	97

The below code will plot the scatter diagram of chemistry and physics subject marks.

```
# Importing library
import pandas as pd
import matplotlib.pyplot as plt

# Data of Chemistry and Physics subject marks
data = {
  "Chemistry_Marks" : [90, 92, 97, 98, 96, 94, 91, 93, 95],
  "Physics_Marks" : [92, 94, 98, 97, 96, 97, 93, 95, 97]
             }


# Dataframe Creation
df = pd.DataFrame(data)

# Plotting scatter plot with title and label
df.plot(x='Chemistry_Marks', y='Physics_Marks', kind = 'scatter', color = 'Green')
plt.title("Scatter Plot")
plt.xlabel("Chemistry")
plt.ylabel("Physics")
```

```
# show the plot
plt.show()
```

The above code will create scatter plot as follow:



In the above scatter plot we can see the relationship between chemistry and physics subject marks.

5.4.2 Karl Pearson Coefficient of Correlation

This is the most common measure of correlation. It is also known as Pearson Product Moment Correlation (PPMC). It is used to measure the strength and direction of the linear relationship between two variables in correlation analysis. It is used to measure the degree of association between variables. The correlation coefficient is symbolized by r which lies between -1 and 1.

- ➢ The value of r is 1 indicates perfect positive correlation.
- ➢ The value of r is -1 indicates perfect negative correlation.
- ➢ The value of r is 0 indicates no relationship.

The following formula is used to calculate the Pearson correlation.

$$r = \frac{n\,(\sum xy\,) - (\sum x\,)\,(\sum y\,)}{\sqrt{[\,n\,\sum x^2 - (\sum x\,)^2\,][\,n\,\sum y^2 - (\sum y\,)^2\,]}}$$

Here,

n = No. of values
ΣX = Total of the First Variable Value
ΣY = Total of the Second Variable Value
ΣXY = Sum of the Product of first and Second Value
ΣX^2 = Sum of the Squares of the First Value
ΣY^2 = Sum of the Squares of the Second Value

Example : Calculate the Karl Pearson's Coefficient of correlation from the following data.

X	10	20	30	40	50	60	70	80
Y	32	48	59	64	70	67	78	82

Solution:

X	**Y**	**XY**	**X^2**	**Y^2**
10	32	320	100	1024
20	48	960	400	2304
30	59	1770	900	3481
40	64	2560	1600	4096
50	70	3500	2500	4900
60	67	4020	3600	4489
70	78	5460	4900	6084
80	82	6560	6400	6724
ΣX = 360	**ΣY = 500**	**ΣXY = 25150**	**ΣX^2 = 20400**	**ΣY^2 = 33102**

Now, we calculate

$$r = \frac{n\,(\sum xy) - (\sum x)(\sum y)}{\sqrt{[\,n\sum x^2 - (\sum x)^2\,][\,n\sum y^2 - (\sum y)^2\,]}}$$

$$r = \frac{8\,(25150) - (360)(500)}{\sqrt{[\,8\,(20400) - (360)(360)][\,8\,(33102) - (500)(500)\,]}}$$

$$r = \frac{201200 - 180000}{\sqrt{[\,163200 - 129600\,]\,[\,264816 - 250000\,]}}$$

$$r = \frac{21200}{\sqrt{(\,33600\,)\,(14816)}}$$

$$r = \frac{21200}{\sqrt{497817600}}$$

$$r = \frac{21200}{22311.826}$$

$$r = 0.9501687$$

Now we will calculate the Karl Pearson's Coefficient of correlation using Python.

The below code will calculate the Karl Pearson's Coefficient.

```
# Importing library
import pandas as pd

# Data of Age and Weight
data = {"Age" : [10, 20, 30, 40, 50, 60, 70, 80],
        "Weight" : [32, 48, 59, 64, 70, 67, 78, 82]}

df = pd.DataFrame(data)

#Calculating Correlation
df.corr(method='pearson')
```

The above code will give the following result :

	Age	**Weight**
Age	1.000000	0.950169
Weight	0.950169	1.000000

Here we can see that the value of Karl Pearson’s Coefficient of correlation (r) is 0.950169 which is same as the manual calculation as above.

5.4.3 Spearman’s Rank Correlation Coefficient

In year 1904, Charles Edward Spearman introduced a new method of measuring the correlation between two variables. It is applicable to individual observation. In this method, the rank (or order) of the observation is taken instead of the value of variable. This correlation coefficient is called rank correlation coefficient. This is useful to measure the qualitative characteristics such as beauty, honesty, height etc.

The following formula is used to calculate the Spearman’s Rank Coefficient of correlation.

$$r = 1 - \frac{6\ \sum D^2}{n\ (n^2 - 1)}$$

Here,

r = Spearman rank correlation coefficient
n = No. of pairs of observation
D = Differences in ranks between pair observation

Example : Calculate the Spearman’s Rank Coefficient of correlation from the following data.

X	1	6	5	10	3	2	4	9	7	8
Y	6	4	9	8	1	2	3	10	5	7

Solution:

Rank (X)	**Rank (Y)**	**D = R (X) – (Y)**	**D^2**
1	6	-5	25
6	4	2	4
5	9	-4	16
10	8	2	4
3	1	2	4
2	2	0	0
4	3	1	1
9	10	-1	1
7	5	2	4
8	7	1	1
N = 10			**$\sum$D^2 = 60**

Now, we calculate

$$r = 1 - \frac{6\ \sum D^2}{n\ (n^2 - 1)}$$

$$r = 1 - \frac{6\ (60)}{10\ (100 - 1)}$$

$$r = 1 - \frac{360}{990}$$

$$r = 1 - \ 0.36363$$

$$r = 0.63636$$

Now we will calculate the Spearman’s Rank Coefficient of correlation using Python.

The below code will calculate the Spearman’s Rank Coefficient.

```
# Importing library
import pandas as pd

# Data
data = {"X" : [1,6,5,10,3,2,4,9,7,8],
        "Y" : [6,4,9,8,1,2,3,10,5,7]}

df = pd.DataFrame(data)

#Calculating Correlation
df.corr(method='spearman')
```

The above code will give the following result :

	X	Y
X	1.000000	0.636364
Y	0.636364	1.000000

Here we can see that the value of Spearman’s Rank Coefficient of correlation (r) is 0.636364 which is same as the manual calculation as above.

5.5 REGRESSION ANALYSIS

Regression analysis is a set of various statistical methods for estimating the relationship between a dependent variable (also called outcome variable) and one or more independent variables (also called predictor variable). This is widely used to predict the outputs, forecasting, time series analysis and finding the causal effect dependencies.

There are many different types of regression techniques based on the number of dependent variables, the dimensionality of regression line and the types of dependent variable such as linear regression, polynomial regression, decision tree regression, support vector regression, random forest regression, logistic regression, etc. The most frequently used regression analysis is linear regression.

The following terms are related to regression analysis:

- **Dependent variable or target variable:** Variable to predict.
- **Independent variable or predictor variable:** Variables to estimate the dependent variable.
- **Outlier:** The observation which differs significantly from the other observation.
- **Normality**: The data follows a normal distribution.
- **Multicollinearity:** It is a situation in which two or more independent variables are highly linearly related.
- **Homoscedasticity:** It is a situation in which the error term is the same across all values of the independent variables. It is also called homogeneity of variance.

5.6 APPLICATIONS OF REGRESSION ANALYSIS

Regression analysis refers to a group of techniques for studying the relationships among two or more variables based on a sample. Linear regression is one of the most commonly used techniques in statistics which is used to quantify the relationship between one or more predictor variables and a response variable.

Regression analysis is used for prediction and forecasting. This statistical method is used in various sectors. The most common applications of regression analysis are:

- **Financial Sector** – Regression analysis is used to calculate the Beta (volatility of returns relative to the overall market) which is used as a measure of risk. A company with a higher beta has a greater risk and greater expected returns also. It is also used in stock market to understand the trend of stocks and forecast the prices of different stocks.

- **Marketing Sector** – Regression analysis is used to understand the effectiveness of various marketing campaigns, forecasting the sales and pricing of the products. It is also used to measure the strength of a relationship between different variables such as customer satisfaction with product quality and price of product.

- **Manufacturing Sector** – Regression analysis is used to evaluate the relationship of variables that determine to define a better engine to provide better performance. An important application of regression analysis in manufacturing sector is to estimate the cost of product. In manufacturing industries, an accurate cost prediction during a new product development process is most important factor for manufacturing firms to survive in this competition era.

- **Sales and Promotion Sector** – Regression analysis is used to analyses promotions on sales of product. It is used to find the relationship between the amount spent on advertising on a product and determines the amount of its sales. It finds the return on investment such as a company want to find the amount that have invested in marketing of particular products or brands and sales of that product.

- **Medical Sector** – Regression analysis is an important statistical method for theanalysis of medical data. It enables the identification and characterization of relationships among multiple factors in online public health data. It is used for prediction and clarification can both be appropriate for public health data analysis for better understanding of public health outcomes. It is also used to forecast the different combination of medicines to prepare generic medicines for diseases.

5.7 TYPES OF REGRESSION ANALYSIS

There are three different types of regression analysis as follows:

- Simple Linear Regression
- Multiple Linear Regression
- Non-Linear Regression

5.7.1 Simple Linear Regression

Simple linear regression analysis is a statistical tool for finding the best relationship between one independent (predictor or explanatory) variable and one dependent (response, outcome) variable which is continuous in nature. Linear Regression is a predictive model used for finding the linear relationship between a dependent variable and one or more independent variables. This relationship represents how an input variable is related to the output variable and how it is represented by a straight line.

The simple linear regression model is expressed by using the following equation:

Y = a + bX + □

Here,

Y = Dependent variable
a = Intercept
b = Slop
X = Independent variable
□ = Error (residual)

Example : Obtain the equation of the lines of regression from the following data. Also estimate the value of Y for X = 28.

X	10	20	30	40	50	60	70	80
Y	32	48	59	64	70	67	78	82

Solution:

X	**Y**	**XY**	**X^2**	**Y^2**
10	32	320	100	1024
20	48	960	400	2304
30	59	1770	900	3481
40	64	2560	1600	4096
50	70	3500	2500	4900
60	67	4020	3600	4489
70	78	5460	4900	6084
80	82	6560	6400	6724
ΣX = 360	**ΣY = 500**	**ΣXY = 25150**	**ΣX^2 = 20400**	**ΣY^2 = 33102**

Now, we calculate

$$\bar{X} = \frac{\sum x}{n} = \frac{360}{8} = 45$$

$$\bar{Y} = \frac{\sum y}{n} = \frac{500}{8} = 62.5$$

$$SSxy = \sum xy - \frac{(\sum x)(\sum y)}{n} = 25150 - \frac{(360)(500)}{8} = 25150 - 22500 = 2650$$

$$SSxx = \sum x^2 - \frac{(\sum x^2)}{n} = 20400 - \frac{(360)^2}{8} = 20400 - 16200 = 4200$$

$$SSyy = \sum y^2 - \frac{(\sum y^2)}{n} = 33102 - \frac{(500)^2}{8} = 33102 - 31250 = 1852$$

$$b = \frac{SSxy}{SSxx} = \frac{2650}{4200} = 0.63095$$

$$a = \bar{Y} - b\bar{X} = 62.5 - (0.63095)(45) = 62.5 - 28.39 = 34.11$$

So, we get the following equation for regression line.

Y = 34.22 + 0.63095 X

Now, we can use this equation for prediction. So, we can predict the value of Y, when X = 28.

So, Y = 34.22 + 0.63095 X = 34.22 + 0.63095 (28) = 34.22 + 7.66 = 51.77

From the above example, we get the predicted value of Y is 51.77 when the value of X is 28.

Now we will calculate the intercept and slope using Python.

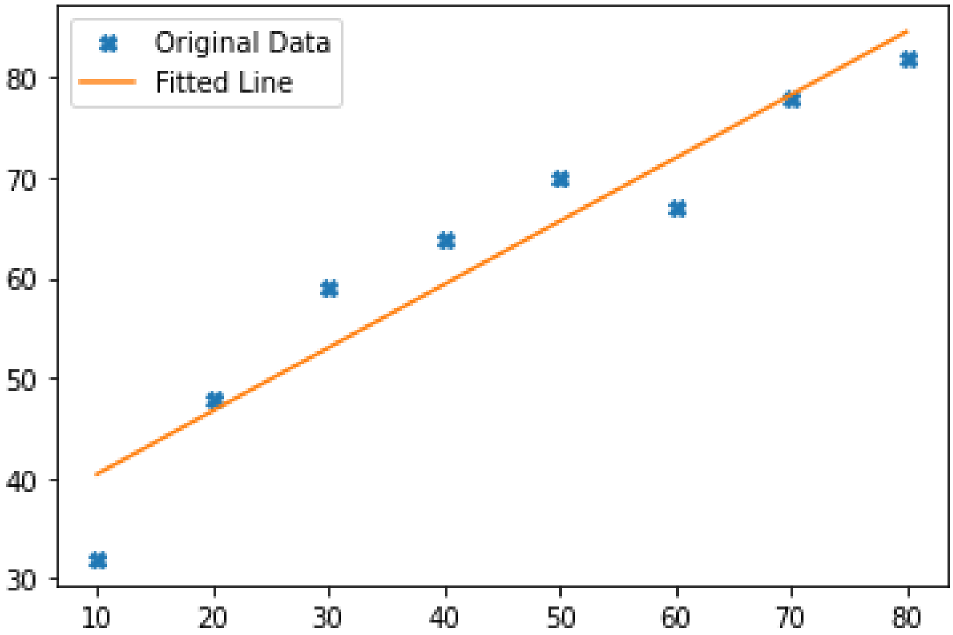
The below code will calculate the intercept and slope and to get the equation of the line of regression. It will also estimate the value of Y for X = 28.

```
# Importing library
import matplotlib.pyplot as plt
from scipy import stats

# Data
x = [10, 20, 30, 40, 50, 60, 70, 80]
y = [32, 48, 59, 64, 70, 67, 78, 82]

# Calculation slop and intercept
slope, intercept, r, p, std_err = stats.linregress(x, y)
print("Slop =", slope)
print("Intercept =", intercept)
print("R value =", r)
print("P value =", p)
print("Standard Error =", std_err)

# Function
def reglinefun(x):
  return intercept + slope * x

# Line
line = list(map(reglinefun, x))

# Plotting
```

```
plt.plot(x, y, 'X', label="Original Data")
plt.plot(x, line, label="Fitted Line")
plt.legend()
plt.show()

# Prediction
predict = reglinefun(28)
print("Predicted value =", predict)
```

The above code will give the following result :

```
Slop = 0.6309523809523809
Intercept = 34.10714285714286
R value = 0.9501687384314103
P value = 0.00029790068633099245
Standard Error = 0.08450983564345486
```



From the above code we can get the estimated value of Y is 51.77 when X = 28.

```
Predicted value = 51.773809523809526
```

5.8 SUMMARY

The students will learn many things related to basic statistics in this module and they will be able to calculate the measures of statistics such as correlation and regression. They will also able to perform the various statistical analysis using Python.

- Ability to understand the correlation in the statistics.

- Ability to understand the various types of correlation such as positive, negative, zero, simple, partial, multiple, linear and non-linear correlation.
- Ability to do understand the methods of studying correlation using scatter diagram.
- Ability to calculate the Karl Pearson correlation coefficient and Spearman rank correlation coefficient.
- Ability to understand the regression analysis and applications of regression analysis.
- Ability to obtain the line of regression and predication using simple linear regression.

5.9 PRACTICE QUESTIONS

Short Answer:

1. What is correlation?
2. Define positive and negative correlation.
3. Define simple correlation and multiple correlation.
4. What is partial correlation?
5. Define linear and non-linear correlation
6. What is regression?
7. List types of regression.

Long Answer:

1. What is correlation? Explain types of correlation.
2. Explain scatter diagram for correlation.
3. Explain Karl Pearson correlation coefficient with example.
4. Explain Spearman Rank correlation coefficient with example.
5. Explain applications of regression analysis.
6. Explain simple linear regression with example.

PRACTICALS

1. Find the correlation between height of father and son (in cm) of the following.

Height of Father	65	66	67	67	68	69	71	73
Height of Son	64	68	65	69	71	70	69	71

2. Find the correlation between marks obtained by eight students in physics, chemistry and biology.

Physics	75	56	70	82	64	78	49	59
Chemistry	80	67	67	75	70	60	55	68
Biology	67	64	68	77	72	73	58	62

3. Draw the scatter plot of age of father and daughter (in year) of the following.

Age of Father	40	45	48	50	51	54	58	60
Age of Daughter	12	13	18	20	22	26	28	32

4. Calculate the Karl Pearson correlation coefficient of the following:

Price	10	20	30	40	50	60	70
Supply	8	7	13	15	11	18	20

5. Calculate the Karl Pearson correlation coefficient of the following:

Age of Husband	25	27	28	30	30	31	34	38
Age of Wife	23	26	27	29	26	28	34	36

6. Calculate the Spearman Rank correlation coefficient of the following:

X	65	66	67	67	68	69	71	73
Y	64	68	65	69	71	70	69	71

7. Calculate the Spearman Rank correlation coefficient of the following:

X	4	3	8	7	1	5	2	6
Y	6	2	7	5	3	4	1	8

8. Obtain the equation of line of regression from the following data.

X	65	66	67	67	68	69	71	73
Y	64	68	65	69	71	70	69	71

9. Obtain the line of regression equation age and blood pressure of the following and predict the value for blood pressure when age is 50 year.

Age	42	55	61	38	68	74	64	46	58	70
Blood Pressure	126	140	148	121	145	160	155	130	142	156

REFERENCES

Books

1. Gupta, S.C. and Kapoor, V.K.: Fundamentals of Mathematical Statistics, Sultan Chand & Sons, New Delhi, 11th Ed
2. Hastie, Trevor, et al. The Elements of Statistical Learning, Springer
3. Ross, S.M.: Introduction to Probability and Statistics for Engineers and Scientists, Academic Press
4. Papoulis, A. and Pillai, S.U.: Probability, Random Variables and Stochastic Processes, McGraw Hill

Web References

1. https://www.geeksforgeeks.org

2. https://www.tutorialspoint.com
3. https://www.w3schools.com
4. https://pandas.pydata.org
5. https://pbpython.com
6. https://www.statisticshowto.com
7. https://realpython.com

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

Unit 6: Mathematical and Statistical Probability

STRUCTURE

6.0 OBJECTIVES

The main goal of this module is to teach you the basics of random experiment and related terms like sample space, sample points, events, type of events and algebra of events which are the building blocks for learning basic concepts of probability and probability theory. By studying this module you should be able to:

- Understand what is random experiment,
- define the sample space of a random experiment
- identify events, types of event,
- Learn about the algebra of events and some algebraic properties of events.

Some examples are given at the end of each definition to understand the basic concept. Few examples are done using python programming language.

6.1 INTRODUCTION

In our day to day life, we came across situations, where we cannot predict the result of our action or outcome of experiment we are conducting. But we may know that the outcome has to one of the several possibilities. For example, (a) gender of newly born baby cannot be predicted before birth but we know that the gender of newly born baby has to be either “Male” or “Female”, (b) result of class 12 student cannot be predicted before the declaration of the result, but we know that the result has to be either “Pass” or “Fail”, (c) when a coin is tossed, we cannot predict the outcome of experiment before the completion of the experiment, but we know that the outcome has to be either “Head” (H) or “Tail” (T). Such experiments, whose outcomes cannot be predicted in advance before the completion of experiment, are called random experiment.

The concepts of such experiments are fundamental to study the theory of probability. Before learning the probability theory, we need to understand some basic terminology like random experiment, sample space, event, types of events and some algebraic operations on events. In subsequent sections, you will learn about all these basic terms with some illustrations.

6.2 RANDOM EXPERIMENT

An experiment is a process that generates well defined outcomes (result). In an experiment, if all possible outcomes of an experiment are known in advance but which particular outcome will occur can be determined with certainty only after the completion of an experiment, then such experiment is called a *random experiment*.

In short, an experiment whose outcome is not predictable with certainty in advance is called a random experiment.

Examples:

a) Tossing of a fair coin,
b) Rolling a fair die,
c) Experiment of tossing two fair coins,
d) Rolling two fair dice,
e) Sex of newly born baby,
f) Recording a person's Blood group,

6.3 Sample Point and Sample Space

Sample Point

Outcome of a random experiment is known as *sample point*. In case of tossing a fair coin, the sample point may be either H or T.

Sample Space

The set of all possible outcomes of a random experiment is called a *sample space* of that random experiment. Usually sample space is denoted by S or U.

Examples:

a) If a coin is tossed once, then the sample space is
S = {H, T}, H : head, T : tail.

b) If a die is thrown once, then the sample space is
S = {1, 2, 3, 4, 5, 6}.

c) If two coins are tossed, then the sample space is
S = {HH, HT, TH, TT}.

d) If two dice are thrown, then the sample space is
S = {(1,1), (1,2), (1,3), (1,4), (1,5), (1,6),
(2,1), (2,2), (2,3), (2,4), (2,5), (2,6),
(3,1), (3,2), (3,3), (3,4), (3,5), (3,6),
(4,1), (4,2), (4,3), (4,4), (4,5), (4,6),
(5,1), (5,2), (5,3), (5,4), (5,5), (5,6),
(6,1), (6,2), (6,3), (6,4), (6,5), (6,6)}.

e) If we record sex of newly born baby, then the sample space is
S = {Male, Female}

f) If we record blood type of a person, then the sample space is
S = {O, A, B, AB}.

6.4 Event

Any subset of the sample space that together has some property we are interested in is called event. Generally, events are denoted by A, B, C,… or A_1, A_2, A_3,….

Examples:

(a) In tossing two coins, the sample space is S = {HH, HT, TH, TT}. Following are some events:

i. A = one head occurs, i.e., A = {HT, TH},
ii. B = two head occurs, i.e., B = {HH},
iii. C = no head occurs, i.e., C = {TT}.

(b) In throwing a die, the sample space is S = {1, 2, 3, 4, 5, 6}. Following are some events:

i. A = even number appear on the top, i.e., A = {2, 4, 6},
ii. B = a number divisible by 3 appear on the top, i.e., B = {3, 6},
iii. C = a number less than 4 appear on the top, i.e., C = {1, 2, 3, 4}.

6.5 TYPES OF EVENTS

6.5.1 Simple and Compound Event

Simple Event

An event consisting of single element of the sample space is called a simple event.

For example, in tossing two coins the sample space is S = {HH, HT, TH, TT}. Let A be the event that two head occurs and B be the event that only one head occurs and it is on the first coin. Then A = {HH} and B = {HT} both are simple events.

Compound Event

An event consisting of more than one element of the sample space is called a compound event.

For example, in tossing two coins the sample space is S = {HH, HT, TH, TT}. Let C be the event that at least one head occurs and B be the event that one head occurs. Then C = {HT, TH, HH} and D = {HT, TH} both are compound events.

6.5.2 Elementary Events

All singleton subsets of sample space of random experiment are called elementary events. That is, events that cannot be broken down into other events are called elementary events. Generally they are denoted by E_1, E_2, E_3, …. All elementary events are simple events.

Examples:

(a) In a random experiment of tossing a coin, the sample space is S = {H, T}. The events E_1 = {H} and E_2 = {T} are elementary events.

(b) In a random experiment of throwing a die, the sample space is S = {1, 2, 3, 4, 5, 6}. The events E_1 = {1}, E_2 = {2}, E_3 = {3}, E_4 = {4}, E_5 = {5} and E_6 ={6} are elementary events.

6.5.3 Equally Likely (equiprobable) Events

Two events are said to be equally likely if one of them cannot be expected to occur in preference to other. Clearly they have equal chance or likelihood of individual occurrence.

Examples:

(a) In tossing a coin, the sample space is S ={H, T}. Here likelihood (probability) of head is same as that of likelihood of tail. So the event A = {H} and B = {T} are equally likely events or equiprobable events.

(b) In throwing a die, the sample space is S = {1, 2, 3, 4, 5, 6}. The likelihood of simple events 1, 2, …, 6 are equally likely.

6.5.4 Impossible and Sure Event

Impossible Event

In a random experiment, the event that is logically impossible is called an impossible event or null event. Event corresponding to null set (empty) is an impossible event.

Examples:

(a) In a random experiment of throwing a die, the sample space is S = {1, 2, 3, 4, 5, 6}. Let A be the event corresponding to the number 8 appears on the top. Clearly the event A = { } =ϕ is an impossible event.

(b) In a random experiment of tossing two coins, the sample space is S = {HH, HT, TH, TT}. Let A be the event that at least three heads occurs. Clearly the event A = { } = ϕ is an impossible event.

Sure Event

In a random experiment, the event that is certain or sure to occur is called certain or sure event. Event corresponding to the sample space is called certain event.

Examples:

(a) In a random experiment of tossing a coin, the sample space is S = {H, T}. If A = event that either head or tail occurs, then clearly event A = {H, T} = S is sure event.

(b) In a random experiment of throwing a die, the sample space is S = {1, 2, 3, 4, 5, 6}. Let A be the event that a natural number less than 7 appears on the top. Clearly the event A = {1, 2, 3, 4, 5, 6} = S is sure event.

6.6 Algebra of Events

6.6.1 Complementary Event

Consider event A associated with a sample space S. The event corresponding to set of points in the sample space S, not belonging to A is called complementary event of A and is denoted by A’ or A^c. In words, A^c means “not A”. In set notation, it can be written as $A^c = \{x \mid x \in S; x \notin A\}$.

Examples:

(a) In tossing of two coins, the sample space is S = {HH, HT, TH, TT}. Let A be the event that two head occurs, i.e., A = {HH}. Then complementary event of A is A^c = {HT, TH, TT}.

(b) In throwing a die, the sample space is S = {1, 2, 3, 4, 5, 6}. Let B be the event that a number divisible by 3 appear on the top, i.e., B = {3, 6}. Then the complementary event of B is B^c = {1, 2, 4, 5}

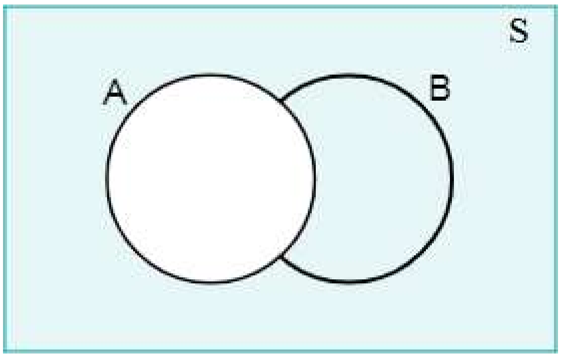


Figure 6.1: Complementary event

In the above Venn-diagram, the shaded region is A^c (Complementary event to A).

Python Code for Complementary event

```
# Create Universal Set S as follows
S = {1,2,3,4,5,6}
# Create Set A as follow
A = {2,5,6}
print("\nComplementary of A using Difference '-' operator")
S - A
```

The above code will give the following result:

```
Complementary of A using Difference '-' operator
{1, 3, 4}
```

6.6.2 Equality of Events

Let A and B be the events of sample space S. Then A and B are equal event if $A \subset B$ and $B \subset A$, i.e., two events are said to be equal if all the sample points of A belongs to B and vice versa.

Example:

In a random experiment of throwing a die, the sample space is S = {1, 2, 3, 4, 5, 6}. Let A be the event that even number appear on the top of the die and B be the event that a number divisible by 2 appear on the top of the die. Then A = {2, 4, 6} and B = {2, 4, 6}. As $A \subset B$ and $B \subset A$, event A and B are equal.

Python Code to check equality of events

```
A = {10, 12, 16, 18}
B = {12, 18, 10, 16}
A == B
print("Set A and B are equal")
C = {12,17,10,18}
A == C
print("Set A and C are not equal")
```

The above code will give the following results:

```
True
Set A and B are equal
False
Set A and C are not equal
```

6.6.3 Union of Events

Consider event A and B as two events associated with a sample space S. The union of two events A and B is denoted by $A \cup B$ and is defined as a set consisting of all those members which belongs to either A or B or both. In words, $A \cup B$ means "A or B". In set notation, it can be written as $A \cup B = \{x \mid x \in A \text{ OR } x \in B\}$.

Examples:

(a) If event A is A = {1, 2, 3, 5, 6} and event B is B = {1, 2, 4, 6}. Then $A \cup B$ = {1, 2, 3, 4, 5, 6}.

(b) Let the sample space be S = {1, 2, 3, 4, 5, 6, 7, 8}. Let A = {1, 2, 3, 4, 5, 6} and B = {2, 3, 7, 8}. Then $A \cup B$ = {1, 2, 3, 4, 5, 6, 7, 8}.

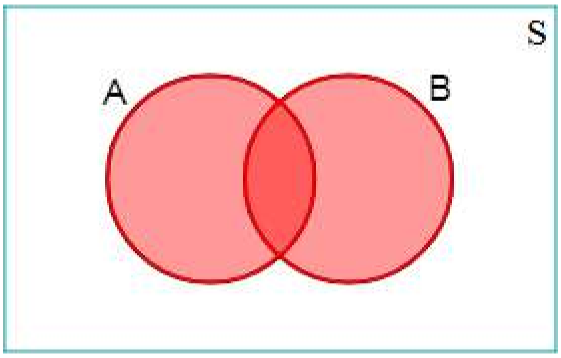


Figure 6.2: Shaded Region is the A ∪ B

In the above Venn-diagram, the shaded region is the union of A and B.

Python Code for Union of two events

```
# Create Set A as follows
A = {1,2,3,4,5,6}
# Create Set B as follow
B = {2,3,7,8}
print("\nUnion using '|' Operator")
A | B
print("\nUnion using 'union()' Function")
A.union(B)
```

The above code will give the following results:

```
Union using '|' Operator
{1, 2, 3, 4, 5, 6, 7, 8}
Union using 'union()' Function
{1, 2, 3, 4, 5, 6, 7, 8}
```

6.6.4 Intersection of Events

Consider event A and B as two events associated with a sample space S. The intersection of two events A and B is denoted by A ∩ B and is defined as a set consisting of all those members which belongs to both A and B. In words, A ∩ B means “A and B”. In set notation, it can be written as A ∩ B = {x | x ∈ A and x ∈ B}.

Examples:

(a) If event A is A = {1, 3, 5, 6} and event B is B = {1, 2, 4, 6}. Then A ∩ B = {1, 6}.

(b) Let the sample space be S = {1, 2, 3, 4, 5, 6, 7, 8}. Let A = {1, 2, 3, 4, 5, 6} and B = {2, 3, 7, 8}. Then A ∩ B = {2, 3}.

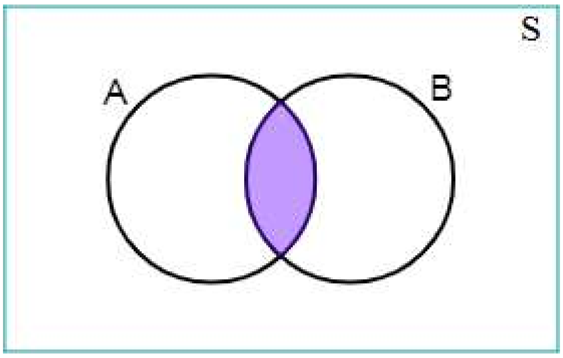


Figure 6.3: Intersection of Two events

In the above Venn-diagram, the shaded region is the intersection of A and B.

Python Code for Intersection of two events

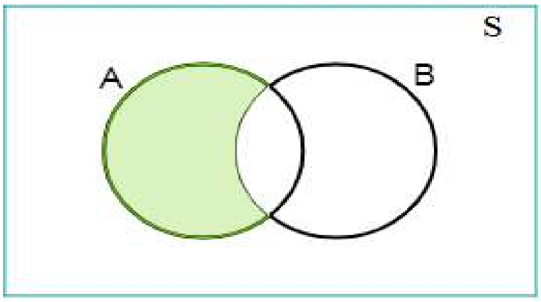
```
# Create Set A as follows
A = {1,2,3,4,5,6}
# Create Set B as follow
B = {2,3,7,8}
print("\nIntersection using '&' Operator")
A & B
print("\nIntersection using 'intersection()' Function")
A.intersection(B)
```

The above code will give the following results:

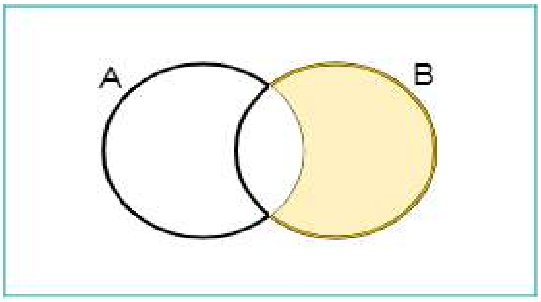
```
Intersection using '&' Operator
{2, 3}
Intersection using 'intersection()' Function
{2, 3}
```

6.6.5 Difference of Events

Let A and B be the events of the sample space S. Then A – B is known as difference of event A from event B. In difference A – B, the event A occur but B does not occur. i.e., A – B = {x | x ∈ A; x ∉ B}. It can also be written as A – B = A ∩ B^c. Similarly, B – A is known as difference event of B from A, i.e., B – A = {x | x ∈ B, x ∉ A}. It can also be written as B – A = B ∩ A^c.



A – B = A ∩ B^c



B – A = B ∩ A^c

Figure 6.4: Difference of Two Events

In the above Venn-diagram, the first figure shows A – B and second shows B – A.

Example:

(a) In a random experiment of tossing two coins, the sample space is S = {HHH, HHT, HTH, THH, HTT, THT, TTH, TTT}. Let A be the event that at least two head occurs, i.e., A = {HHH, HHT, HTH, THH} and B be the event that at least one tail occurs, i.e., B = {HHT, HTH, THH, HTT, THT, TTH, TTT}.
Then A – B = {HHH} = outcomes belongs to A and do not belong to B.

(b) Let the sample space be S = {1, 2, 3, 4, 5, 6, 7, 8}. Let A = {1, 2, 3, 4, 5, 6} and B = {2, 3, 7, 8}. Then A – B = {1, 4, 5, 6}.

Python Code for difference of two events

```
# Create Set A as follows
A = {1,2,3,4,5,6}
# Create Set B as follow
B = {2,3,7,8}
print("Difference using '-' operator\n")
A - B
print("Difference using 'difference()' function\n")
A.difference(B)
```

The above code will give the following result:

```
Difference using '-' operator
{1, 4, 5, 6}
Difference using 'difference()' function
{1, 4, 5, 6}
```

Mutually Exclusive Events

Events are said to be mutually exclusive if occurrence of one of them prevents the occurrence of any of the remaining ones. That is, there is no possibility of A and B both being true, then they are said to be mutually exclusive or disjoint. Symbolically, A and B are mutually exclusive if their intersection is null or empty set, i.e., $A \cap B = \phi$.

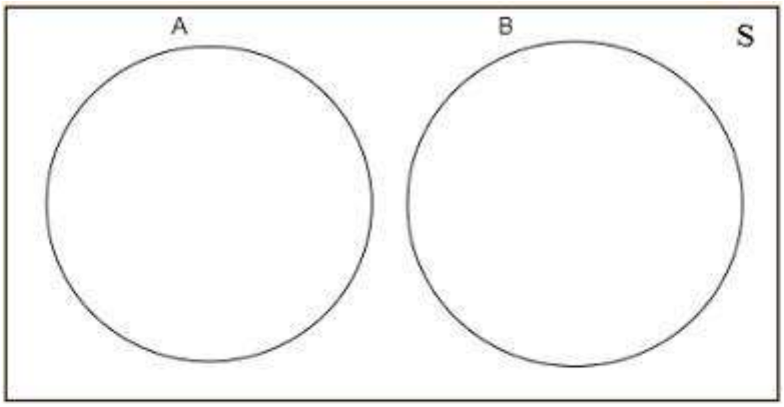


Figure 6.5: Mutually Exclusive Events

In the above Venn-diagram, the two events A and B are mutually exclusive.

Examples:

(a) In tossing of a coin experiment, the sample space is S = {H, T}. Let A = {H}, head occur and B = {T}, tail occur. Then A and B are mutually exclusive events ($A \cap B = \phi$).

(b) In a random experiment of throwing an unbiased die, the sample space is S = {1, 2, 3, 4, 5, 6}. Let A be the event that even number appear on the top, B be the event that odd number appear on the top and C be the event that a multiple of 3 appear on the top. Then A = {2, 4, 6}, B = {1, 3, 5} and C = {3, 6}. The events A and B are mutually exclusive because $A \cap B = \phi$. The events A and C are not mutually exclusive because $A \cap C = \{6\} \neq \phi$. The events B and C are not mutually exclusive because $B \cap C = \{3\} \neq \phi$.

Python Code to check mutually exclusive events

```
# Create Set A as follows
A = {2,4,6}
# Create Set B as follow
B = {1,3,5}
# Create Set C as follow
C = {3,6}
print("Intersection of A and B is ")
A.intersection(B)
print("A and B are mutually exclusive.")
print("Mutually exclusive using 'isdisjoint()' function")
A.isdisjoint(B)
print("Intersection of A and C is ")
A.intersection(C)
print("A and C are not mutually exclusive")
print("Mutually exclusive using 'isdisjoint()' function")
A.isdisjoint(C)
print("Intersection of B and C is ")
B.intersection(C)
print("B and C are not mutually exclusive")
print("Mutually exclusive using 'isdisjoint()' function")
B.isdisjoint(C)
```

The above code will give the following results:

```
Intersection of A and B is
set()
A and B are mutually exclusive.
Mutually exclusive using 'isdisjoint()' function
True
Intersection of A and C is
{6}
A and C are not mutually exclusive
Mutually exclusive using 'isdisjoint()' function
False
Intersection of B and C is
```

```
{3}
B and C are not mutually exclusive
Mutually exclusive using 'isdisjoint()' function
False
```

6.7 Exhaustive Events

If the union of two or more events is the sample space S then they are calledexhaustive events. Simple (elementary) events of random experiments always constitute an exhaustive event.

Examples:

(a) Consider an experiment of throwing die. The sample space is S = {1, 2, 3, 4, 5, 6}. Let A_1, A_2, …, A_6 be the events that 1, 2, … 6 appears on the top respectively. Clearly, at any throw at least one of these events will occur and $A_1 \cup A_2 \cup A_3 \cup A_4 \cup A_5 \cup A_6 = S$. Events A_1, A_2, …, A_6 constitute an exhaustive event.

(b) Consider an experiment of tossing a coin. The sample space is S = {H, T}. Let B_1 = {H} and B_2 = {T}. Clearly, at any toss at least one of these event will occur and $B_1 \cup B_2 = S$. Events A and B are exhaustive events.

6.8 Mutually Exclusive and Exhaustive Events

If $A \cap B = \phi$ and $A \cup B = S$, then A and B are called mutually exclusive and exhaustive events. All the elementary events are always mutually exclusive and exhaustive, so they form partition of the sample space.

Example:

(a) In a random experiment of tossing three coins, the sample space is S = {HHH, HHT, HTH, THH, HTT, THT, TTH, TTT}. Let A be the event that at least two head occurs, B be the event that one or less head occurs. Then A = {HHH, HHT, HTH, THH} and B = {HTT, THT, TTH, TTT}. As $A \cap B = \phi$ and $A \cup B = S$, the events A and B are mutually exclusive and exhaustive.

(b) In a random experiment of throwing an unbiased die, the sample space is S = {1, 2, 3, 4, 5, 6}. Let A be the event that even number appear on the top and B be the event that odd number appear on the top. Then A = {2, 4, 6} and B = {1, 3, 5}. The events A and B are mutually exclusive and exhaustive because $A \cap B = \phi$ and $A \cup B = S$.

6.9 Algebraic Properties of Events

Algebraic properties of events are the rules similar to rules of algebra on the events to perform unions, intersections and complementary of events.

Let A, B, C be the events of sample space S, then following laws holds:

- **Identity Law**
  a) $A \cup \phi = A$
  b) $A \cap \phi = \phi$
  c) $A \cup S = S$
  d) $A \cap S = A$
- **Idempotent Law**
  a) $A \cup A = A$

b) $A \cap A = A$

- **Complement Law**
  a) $A \cup A^c = S$
  b) $A \cap A^c = \phi$
  c) $(A^c)^c = A$
- **Commutative Law**
  a) $A \cup B = B \cup A$
  b) $A \cap B = B \cap A$
- **Associative Law**
  a) $A \cup (B \cup C) = (A \cup B) \cup C$
  b) $A \cap (B \cap C) = (A \cap B) \cap C$
- **Distributive Law**
  a) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$
  b) $A \cap (B \cup C) = (A \cap B) \cup (A \cap C)$
- **De Morgan's Law:**
  The very useful and important relationship between the basic operations of forming unions, intersections and complements of events is known as De Morgan's law. De Morgan's law states that the complement of union (intersection) of two events is the intersection (union) of their complements.
  a) $(A \cup B)^c = A^c \cap B^c$
  The complement of union of two events is the intersection of their complements.
  b) $(A \cap B)^c = A^c \cup B^c$
  The complement of intersection of two events is the union of their complements.

6.10 SUMMARY

The students will learn basic of random experiment and related terms like sample point, sample space, events, type of events, operations (complementary, union, intersection and difference) on events. The basic knowledge of random experiments and related terms is necessary to understand the concept and its use in probability theory. Students will able to perform the various operations on events using Python.

- Learn how to write the sample space related to random experiment.
- Learn how to define events from the sample space based on the characteristics of interest.
- Learn how to perform algebraic operations like complementary, union, intersection and difference.
- Learn how to perform these operations on events using Python programming.

6.11 REFERENCES

Books

1. Gupta, S.C. and Kapoor, V.K. (2014): Fundamentals of Mathematical Statistics, Sultan Chand & Sons, New Delhi, 12th Edition.
2. Hastie, Trevor, et al. (2009): The Elements of Statistical Learning, Springer

3. Ross, S.M. (2004): Introduction to Probability and Statistics for Engineers and Scientists, Academic Press
4. Navidi, W. (2011): Statistics for Engineers and Scientists, McGraw Hill, Third Edition.

6.12 QUESTIONS

Short Answer:

1. Define random experiment. Give some examples of random experiment.
2. Define Sample space and sample points with illustration.
3. What is the difference between sample space and event?
4. Define Simple and Compound events with illustration.
5. Define events and mutually exclusive events with illustration.
6. Define difference of events with illustration.
7. Define impossible event. Give some examples of impossible events.
8. Define sure event. Give some examples of sure events.
9. Define Union and intersection of events.
10. Define mutually exclusive events. Give an example of two events that are mutually exclusive and two events that are not mutually exclusive.
11. Classify the below experiments as random or non-random experiments:
    a. Two cards are drawn from pack of 52 cards and the suits (i.e., Club, Diamond, Heart, Spade) to which they belong are noted.
    b. A bowl contains 3 red and 6 blue balls. Two balls are drawn and their colors are noted.
    c. Water is heated in a bowl to a temperature of 100°C for 5 minutes and we observe that water turns into steam.
    d. Inspecting an item from a production line as defective or non-defective.
    e.

Long Answer:

1. An experiment consists of tossing coins three times. Write the sample space. List the events that corresponds to
    a. At least one head
    b. At most one head
    c. More than one head
    d. Two or more heads
    e. More heads than tails.
2. An experiment consists of tossing coins three times. Write the sample space. Are the outcomes are equally likely?
3. An experiment consists of tossing a coin and a six faced die. What is the sample space of this experiment?
4. A box contains 4 Red, 4 Blue and 4 Yellow Marbles. Construct a sample space for the experiment of drawing two marbles in succession (with replacement).
5. Determine whether these events are mutually exclusive:
    a. Draw a card from pack of 52 cards. Let A be the event of getting spade card and B be the event of getting 6.

   b. Draw a card from pack of 52 cards. Let A be the event of getting club card and B be the event of getting an ace.
   c. Toss coin two times. Let A be the event that only one head occur and B be the event that only one tail occur.
   d. Two dice are rolled. Let A be the event that the number appear on the first die is same as the number appear on the second die and B be the event that the sum of the number appear on the top face is 8.

6. Let A, B, C be three events. Write the expression for the following events:
   a. All the three event occur
   b. At least two of the events occur
   c. None of the event occurs
   d. Exactly two events occur.
7. Two dice are thrown. Write the sample space. Let A be the event that the sum of the numbers on the top of dice is 7, let B be the event that sum of the numbers on the top of dice is at most 7. Determine $A \cap B$, $A \cap B^c$, $A \cup B$, $(A \cup B)^c$.

PRACTICALS

1. Write a python code to define the sample space (S) for tossing two coins. Let A be the event that one head occurs and B be the event that at least one lead occurs. Determine $A \cup B$ and $A \cap B$.
2. Write a python code to define the sample space (S) for an experiment of tossing three coins. Let A be the event that at least two head occurs and B be the event that at least one tail occurs. Determine $A \cup B$ and $A \cap B$.
3. Let the sample space be S = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Let A = {2, 4, 6, 8, 10}, B = {1, 3, 5, 7, 9}, C = {3, 6, 9}, D = {5, 10}. Write python code to
   a. Determine $A \cup B$, $A \cap B$, $A \cap C$, $A \cap D$ and $A – D$.
   b. Find complement of A, complement of C.
   c. Check whether A and B are mutually exclusive and exhaustive.
   d. Check whether C and D are mutually exclusive.
   e. Check whether C and D are equal.

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

UNIT VII: Probability

STRUCTURE

7.0 OBJECTIVES

The main goal of this module is to teach you the basics concept of probability, different definitions of probability and various important theorems on probability. By studying this module you should be able to:

- understand how to calculate probability,
- use different theorems on probability to solve the numerical examples

Some examples are solved to understand the concept of probability and use of various theorems of probability. Few examples are done using python programming language.

7.1. INTRODUCTION

In the sense of mathematical logic, a statement is either true or false, and there is no third possibility in this case. Out of this frame, a third possibility does exist: What is the `extent' or `likelihood' of the statement being true? What is the extent or likelihood of the likelihood being false?, e.g., If a boy informs his father that he will either `pass' or `fail' in an examination, then obviously his father will not feel comfortable with this answer. He will be more satisfied by knowing the likelihood (percentage) of passing or failing in the examination. The numerical measurement of this `extent' or `likelihood' is called `Probability' of the happening or phenomenon under consideration..

In subsequent sections, you will learn about different definitions of probability and some important theorems on probability.

7.2. Classical Definition of Probability

If there are n mutually exclusive, equiprobable and exhaustive elements in a sample space S, and if r of them are favourable to occurrence of some event A, then the probability of event A is given by

$$P(A) = \frac{\text{Number of Favourable Cases to } A}{\text{Total Number of Cases}}$$
$$= \frac{r}{n}$$

This definition was given by Laplace.

The favourable cases to happening of A are always less than or equal to the total exhaustive cases, and also they cannot be negative.

Therefore, $0 \leq \mathrm{r} \leq \mathrm{n}$. Dividing by n, we have

$0 \leq \frac{\mathrm{r}}{\mathrm{n}} \leq 1$, i.e., $0 \leq \mathrm{P(A)} \leq 1$.

If r cases are favourable to the happening of an event A, then $\mathrm{n} - \mathrm{r}$ cases are favourable to not happening of A.

Therefore,

$$P(A^c) = \frac{\text{Number of Favourable Cases to } A^c}{\text{Total Number of Cases}}$$
$$= \frac{n-r}{n}$$
$$= 1 - \frac{r}{n}$$
$$= 1 - P(A)$$

Thus, we have $P(A^c) = 1 - P(A)$.

7.3. Statistical or Empirical Definition of Probability

If the random experiment is repeated under essentially the same conditions for large number of time, then the limit of the ratio of number of times an event happens to the total number of trials is defined as the probability of that event. Here assume that the limit exists.

$$P(A) = \lim_{n \to \infty} \frac{r}{n}$$

7.4. Axiomatic Definition of Probability

Axiomatic approach to the probability was introduced by a Russian Mathematician Kolmogorov with the help of set theory.

Suppose S be the sample space of a random experiment. Let φ(S) be the power set of S and R be the set of all real numbers. Suppose that a set function P : φ(S) → R satisfies following axioms (postulates):

Axiom 1: For all $A \in \varphi(S), P(A) \geq 0$.

Axiom 2: $P(S) = 1$.

Axiom 3: For $A_1, A_2 \in \varphi(S), A_1 \cap A_2 = \phi, P(A_1 \cup A_2) = P(A_1) + P(A_2)$.

Then $P: \varphi(S) \rightarrow R$ is called **probability function** and $P(A)$ is called the **probability of event** A.

- From axiom 1, probability of any event is a non-negative real number. (**Non-negativity**)
- From axiom 2, probability of certain event is 1. (**Certainty**)
- From axiom 3, probability function is additive. (**Additivity**)

Axiom 3, can be generalized as follows :
$A_1, A_2, \ldots, A_n$ are mutually exclusive events from $\varphi(S)$,
$P(A_1 \cup A_2 \cup \ldots \cup A_n) = P(A_1) + P(A_2) + \ldots + P(A_n)$.

Properties of Probability

(a) **Non-negativity:** Probability of any event is always non-negative, i.e., $P(A) \geq 0$.
(b) **Certainty:** Probability of sure event or certain event is always 1, i.e., $P(S) = 1$.
(c) **Additivity:** If $A_1, A_2, \ldots, A_n$ are mutually exclusive events, then

$$P(A_1 \cup A_2 \cup \ldots \cup A_n) = P(A_1) + P(A_2) + \ldots + P(A_n)$$

7.5. Theorems on Probability

7.5.1 Probability of Impossible Event

The probability of impossible event is always zero, i.e., $P(\phi) = 0$.

Proof

Let ϕ be an impossible event of sample space S. Then, we have $\phi \cup S = S$ and $\phi \cap S = \phi$. Thus, ϕ and S are mutually exclusive and exhaustive events. Hence by Axiom 2 and 3,

$$P(\emptyset \cup S) = P(S)$$

$$P(\emptyset) + P(S) = P(S), \text{ (Using Axiom 3)}$$

$$P(\emptyset) + 1 = 1 \text{(Using Axiom 2)}$$

i.e., $P(\emptyset) = 0$.

7.5.2 Probability of Complementary Event

If A and A^c are complementary events of the sample space S, then $P(A) + P(A^c) = 1$.

Proof

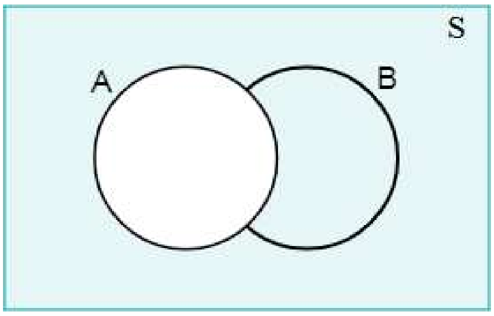

S
A
B

Figure 7.1: Complementary Event of *A*

Since *A* and A^c are complementary events of the sample space *S*, we have $A \cup A^c = S$ and $A \cap A^c = \varphi$. Hence by Axiom 2 and 3,

$$P(A \cup A^c) = P(S)$$

$$P(A) + P(A^c) = P(S), \text{(Using Axiom 3)}$$

$$P(A) + P(A^c) = 1 \text{(Using Axiom 2)}$$

$$\text{i.e., } P(A^c) = 1 - P(A).$$

Thus, we have $P(A) + P(A^c) = 1$.

7.5.3 Addition Law of Probability

If *A* and *B* are two events of non-empty sample space *S*, then

$$P(A \cup B) = P(A) + P(B) - P(A \cap B).$$

Proof

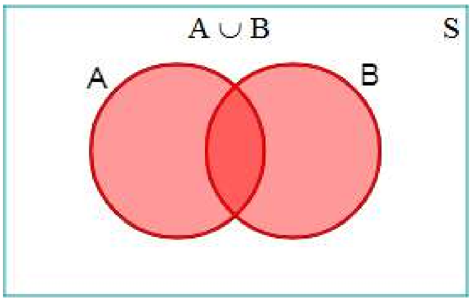


Figure 7.2: Union of *A* and *B*

The above Venn diagram shows $A \cup B$.

From the Venn diagram in Figure 7.3, it is clear that the event $A \cup B$ can be written as $A \cup B = A \cup (A^c \cap B)$ such that *A* and $A^c \cap B$ are mutually exclusive.

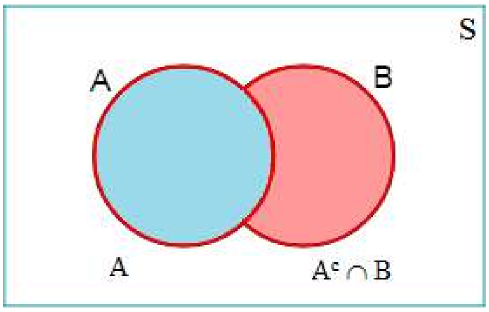


Figure 7.3: $A \cup B = A \cup (A^c \cap B)$

Using Axiom 3,

$$P(A \cup B) = P[A \cup (A^c \cap B)]$$

$$= P(A) + P(A^c \cap B) \qquad \dots\dots (1)$$

Similarly, from the Figure 7.4, the event *B* can be written as union of two mutually exclusive events $A \cap B$ and $A^c \cap B$ such that $B = (A \cap B) \cup (A^c \cap B)$.

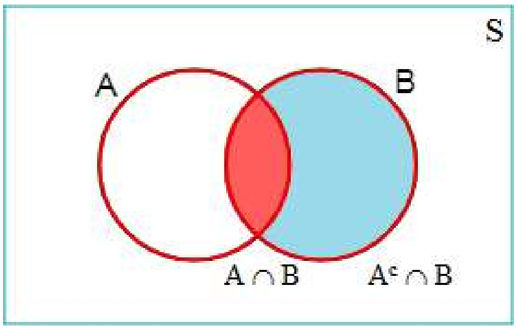


Figure 7.4: $B = (A \cap B) \cup (A^c \cap B)$

Using Axiom 3,

$$P(B) = P[(A \cap B) \cup (A^c \cap B)]$$

$$= P(A \cap B) + P(A^c \cap B)$$

$$\therefore P(A^c \cap B) = P(B) - P(A \cap B)$$

Substituting for $P(A^c \cap B)$ in equation 1, we get

$$P(A \cup B) = P(A) + P(B) - P(A \cap B)$$

7.5.4 Addition Law of Probability for Mutually Exclusive Events

If *A* and *B* are two mutually exclusive events of non-empty sample space *S*, then

$$P(A \cup B) = P(A) + P(B).$$

Proof

Since *A* and *B* are mutually exclusive events, $A \cap B = \phi$.

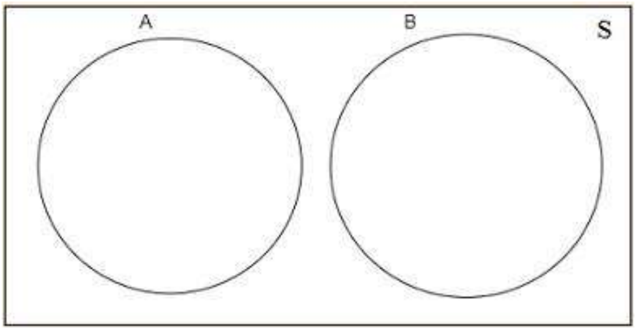


Figure 7.5: Mutually Exclusive Events

Therefore, $P(A \cap B) = P(\phi) = 0$. Hence,

$$P(A \cup B) = P(A) + P(B)$$

7.5.5 Law of Addition for Three Events

For three events *A*, *B* and *C* of non-empty sample space *S*,

$$P(A \cup B \cup C) = P(A) + P(B) + P(C) - P(A \cap B) - P(A \cap C)$$
$$-P(B \cap C) + P(A \cap B \cap C)$$

7.5.6 Some Important Results

Suppose *A* and *B* are the events of the non-empty sample space S.

(a) If $A \subset B$, then $P(A) \leq P(B)$.
(b) $P(A \cap B) \leq P(A)$ or $P(A \cap B) \leq P(B)$.
(c) $P(A \cup B) \geq P(A)$ or $P(A \cup B) \geq P(B)$.
(d) If $A \subseteq B$, then $P(A \cap B) = P(A)$.
(e) If $A \subseteq B$, then $P(A \cup B) = P(B)$.

Example 1:

Consider a random experiment of throwing a fair die. List all the elements of the sample space. Find the probability that

(i) Even number appear on the top
(ii) Odd number appear on the top
(iii) A number greater than or equal to 3 appear on the top.

Solution: The sample space S of the random experiment is S = {1, 2, 3, 4, 5, 6}.

(i) Let A be the event that even number appear on the top. Then A = {2, 4, 6}.

$$P(A) = \frac{\text{Number of favourable cases to A}}{\text{Total number of cases}} = \frac{3}{6} = \frac{1}{2}.$$

(ii) Let B be the event that odd number appear on the top. Then B = {1, 3, 5}.

$$P(B) = \frac{\text{Number of favourable cases to B}}{\text{Total number of cases}} = \frac{3}{6} = \frac{1}{2}.$$

(iii) Let C be the event that a number greater than or equal to 3 appear on the top. Then C = {3, 4, 5, 6}.

$$P(C) = \frac{\text{Number of favourable cases to C}}{\text{Total number of cases}} = \frac{4}{6} = \frac{2}{3}.$$

Python Code to calculate probability

```
# Import a Fraction function from
# fractions library
from fractions import Fraction
# define a user defined function Prob
def Prob(event, space):
   "The probability of an event, given a sample space of equiprobable outcomes."
   return Fraction(len(event & space),
                   len(space))
```

```
## Sample space when a die is thrown
S = {1,2,3,4,5,6}
## Event A is odd number appear on the top
A = {1,3,5}
## Event B is even number appear on the top
```

```
B = {2,4,6}
## Event C is a number greater than or equal to 3
## appear on the top
C = {3,4,5,6}

print("Probability of A is ", Prob(A,S))
print("Probability of B is ", Prob(B,S))
print("Probability of C is ", Prob(C,S))
```

The above code will give the following results:

```
Probability of A is 1/2
Probability of B is 1/2
Probability of C is 2/3
```

Example 2:

Consider a random experiment of tossing three coins. List all the elements of the sample space. Find the probability of getting

(i) at least one head,
(ii) at the most two heads,
(iii) no head.

Solution: Let three coins be tossed. The sample space S consists of 8 elementary events

S = {HHH, HHT, HTH, THH, HTT, THT, TTH, TTT}.

(i) Let A denote the event that at least one head occurs. Then A = {HHH, HHT, HTH, THH, HTT,THT, TTH}.

$$P(A) = \frac{\text{Number of favourable cases to A}}{\text{Total number of cases}} = \frac{7}{8}$$

(ii) Let B denote the event that at the most two head occurs. Then B = {HHT, HTH, THH, HTT,THT,TTH, TTT}

$$P(B) = \frac{\text{Number of favourable cases to B}}{\text{Total number of cases}} = \frac{7}{8}$$

(iii) Let C denote the event that no head occur. Then C ={TTT}

$$P(C) = \frac{\text{Number of favourable cases to C}}{\text{Total number of cases}} = \frac{1}{8}$$

Python Code to calculate probability

```
## import product function from itertools
from itertools import product
## generate sample space for tossing 3 coins
S = set(product(["H","T"],repeat = 3))
S

## Event A at least 1 H
A = {R for R in S if R.count("H") >=1}
A
## Event B at most 2 H
B = {R for R in S if R.count("H")<=2}
B
## Event C no H
C = {R for R in S if R.count("H")==0}
```

```
C
# probability of A
Prob(A,S)
print("Probability of A is ",Prob(A,S))
# probability of B
Prob(B,S)
print("Probability of B is ",Prob(B,S))
# probability of C
Prob(C,S)
print("Probability of C is ",Prob(C,S))
```

The above code will give the following results:

```
Probability of A is 7/8
Probability of B is 7/8
Probability of C is 1/8
```

Example 3:

In a single throw with two uniform dice, what is the probability of getting (a) a total of 9, (b) total different from 9, (c) total is greater than or equal to 8, (d) a total of 7 or 11, (e) maximum of two numbers is greater than 4.

Solution:

Let two uniform dice be thrown. The sample space consists of 36 elementary events.

S = {(1,1), (1,2), (1,3), (1,4), (1,5), (1,6),
(2,1), (2,2), (2,3), (2,4), (2,5), (2,6),
(3,1), (3,2), (3,3), (3,4), (3,5), (3,6),
(4,1), (4,2), (4,3), (4,4), (4,5), (4,6),
(5,1), (5,2), (5,3), (5,4), (5,5), (5,6),
(6,1), (6,2), (6,3), (6,4), (6,5), (6,6)}.

(a) Let A be the event that total of 9 occur. Therefore A = {(3, 6), (4, 5), (5, 4), (6,3)}. So, no. of elements in A is r = 4. Hence, the required probability is

$$P(A) = \frac{\text{Number of favourable cases to A}}{\text{Total number of cases}} = \frac{4}{36} = \frac{1}{9}$$

(b) The event of getting the total different from 9 is the complementary event of A. So, the required probability is

$$P(A^c) = 1 - P(A) = 1 - \frac{1}{9} = \frac{8}{9}$$

(c) Let B be the event that total of greater than or equal to 8 occur. Therefore,

B = { (2,6),
(3,5), (3,6),
(4,4), (4,5), (4,6),
(5,3), (5,4), (5,5), (5,6),
(6,2), (6,3), (6,4), (6,5), (6,6)}.

Number of favorable cases for B = 15. Therefore

$$P(B) = \frac{\text{Number of favourable cases to B}}{\text{Total number of cases}} = \frac{15}{36}$$

(d) Let C be the event that total of 7 occur and D be the event that the total of 11 occur. C ={(1,6), (2,5), (3,4), (4,3), (5,2), (6,1)} and D = {(5,6), (6,5)}. The required probability is C ∪ D. Moreover C and D are mutually exclusive.

$$P(C \cup D) = P(C) + P(D) = \frac{6}{36} + \frac{2}{36} = \frac{8}{36} = \frac{2}{9}$$

(e) Let E = {(x,y) | max (x,y) >4}. Then

E = { (1,5), (1,6)
(2,5), (2,6),
(3,5), (3,6),
(4,5), (4,6),
(5,1),(5,2), (5,3), (5,4), (5,5), (5,6),
(6,1), (6,2), (6,3), (6,4), (6,5), (6,6)}.

No. of cases favourable to E = 20. Therefore, the probability of E is

$$P(E) = \frac{20}{36} = \frac{5}{9}$$

Python Code to calculate probability

```
# Import a Fraction function from
# fractions library
from fractions import Fraction
# define a function Prob
def Prob(event, space):
   "The probability of an event, given a sample space of equiprobable outcomes."
   return Fraction(len(event & space),
                len(space))
## import itertools from product library
from itertools import product
S = set(product([1,2,3,4,5,6],repeat = 2))
print("The sample space is S = ",S)
print("Total cases in S = ",len(S))
```

The above code will give the following results:

```
The sample space is S =  {(1, 3), (6, 6), (5, 6), (2, 1), (6, 2), (1, 6), (5, 1), (2, 5), (1,
2), (3, 3), (5, 5), (4, 4), (6, 3), (1, 5), (3, 6), (2, 2), (4, 1), (1, 1), (6, 4), (3, 2), (2, 6),
(5, 4), (4, 5), (5, 2), (1, 4), (2, 3), (4, 2), (6, 5), (3, 5), (5, 3), (4, 6), (6, 1), (3, 1), (4,
3), (3, 4), (2, 4)}
Total cases in S =  36
```

```
## Event A total of 9 occur
A = {R for R in S if (R[0]+R[1])==9}
print(A)
print("Favourable cases to A",len(A))
# probability of A
Prob(A,S)
print("Probability of A is ",Prob(A,S))
```

The above code will give the following results:

```
{(6, 3), (5, 4), (4, 5), (3, 6)}
Favourable cases to A 4
Probability of A is 1/9
```

```
B= {R for R in S if R[0]+R[1]>=8}
```

```
print(B)
print("Favourable cases to B = ",len(B))
# probability of B
Prob(B,S)
print("Probability of B is ",Prob(B,S))
```

The above code will give the following results:

```
{(6, 4), (5, 4), (2, 6), (5, 5), (4, 6), (6, 6), (5, 6), (4, 5), (4, 4), (6, 3), (6, 2), (3, 6), (5,
3), (6, 5), (3, 5)}
Favourable cases to B = 15
Probability of B is 5/12
```

```
C = {R for R in S if R[0]+R[1]==7}
print(C)
print("Favourable cases to C",len(C))
D = {R for R in S if R[0]+R[1]==11}
print(D)
print("Favourable cases to D",len(D))
# probability of C union D
Prob(C | D,S)
print("Probability of C union D is ",Prob(C | D,S))
```

The above code will give the following results:

```
{(6, 1), (1, 6), (4, 3), (3, 4), (2, 5), (5, 2)}
Favourable cases to C 6
{(5, 6), (6, 5)}
Favourable cases to D 2
Probability of C union D is  2/9
```

```
E= {R for R in S if max(R[0],R[1])>4}
print(E)
print("Favourable cases to E",len(E))
# probability of E
Prob(E,S)
print("Probability of E is ",Prob(E,S))
```

The above code will give the following results:

```
{(6, 6), (5, 6), (6, 2), (1, 6), (5, 1), (2, 5), (5, 5), (6, 3), (1, 5), (3, 6), (6, 4), (5, 4), (2,
6), (4, 5), (6, 5), (5, 3), (3, 5), (4, 6), (6, 1), (5, 2)}
Favourable cases to E 20
Probability of E is 5/9
```

Example 4:

In a lot of 100 electric bulb 10% of them are defective. Five bulbs are selected at random.

a. What is the probability of no defective bulb among the 5 bulbs?
b. What is the probability of 2 bulbs being defective among the 5 bulbs?

Solution:

Total number of bulbs = 100. Out of 100 electric bulb, 10% bulbs are defective, i.e., 10 bulb are defective and remaining 90 are non-defective.

Out of 100 bulbs 5 bulbs can selected in $\binom{100}{5}$ ways. i.e., n = Total number of elements $= \binom{100}{5}$.

And out of 90 non-defective bulbs 5 can be selected in $\binom{90}{5}$ ways. i.e., m = $\binom{90}{5}$.

a. Let A be the event that no defective bulb among selected 5.

$$P(A) = \frac{m}{n} = \frac{\binom{90}{5}}{\binom{100}{5}}$$

b. Let B be the event that two defective bulb among selected 5.

Out of 10 defective bulbs 2 bulbs can be selected in $\binom{10}{2}$ ways and remaining 3 bulb from 90 non-defective can be selected in $\binom{90}{3}$ ways. The events are compound events.

$$P(B) = \frac{m}{n} = \frac{\binom{10}{2}\binom{90}{3}}{\binom{100}{5}}.$$

Example 5:

Two cards are drawn at random simultaneously from a pack of playing cards. Find the probability that

a. both the cards are spade cards,
b. both the cards are of same suit.

Solution:

As two cards are drawn from 52 cards, the sample space consists of $\binom{52}{2}$ elements.

a. Let A be the event that both the cards are space cards. Out of 13 cards of same suit, 2 cards can be selected in $\binom{13}{2}$ ways.

$$P(A) = \frac{\binom{13}{2}}{\binom{52}{2}} = \frac{13 \times 12}{52 \times 51} = \frac{1}{17}.$$

b. Let B be the event that both the cards are of same suit. Two cards of any suit can be selected in $\binom{13}{2}$ ways, but there are 4 suits. So the number of elements favourable to B $= 4 \times \binom{13}{2}$

$$P(B) = \frac{4 \times \binom{13}{2}}{\binom{52}{2}} = \frac{4 \times 13 \times 12}{52 \times 51} = \frac{4}{17}.$$

7.6. Conditional Probability

In a simple language the conditional probability is “What is the chance that something will happen, given that something else has already happened?”.

Let *A* and *B* be two events of non-empty sample space *S*.

The probability of some event *A* when it is known that event *B* has already occurred is called conditional probability of *A* given *B*. It is denoted by *P(A|B)* and is defined by

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, \ \ P(B) > 0.$$

P(A|B) can be read as *probability that A occurs given that B occurs*. The conditional probability of *A* given *B* is the joint probability of *A* and *B* divided by the marginal probability of *B*.

The probability of some event *B* when it is known that event *A* has already occurred is called conditional probability of *B* given *A*. It is denoted by *P(B|A)* and is defined by

$$P(B|A) = \frac{P(B \cap A)}{P(A)}, \ \ P(A) > 0.$$

P(B|A) can be read as *probability that B occurs given that A occurs*. The conditional probability of *B* given *A* is the joint probability of *B* and *A* divided by the marginal probability of *A*.

Result 1:

Let *A* and *B* be two events of non-empty sample space *S* such that *P(A) > 0, P(B) > 0*. Then

$$0 \leq P(A|B) \leq 1.$$

Result 2:

Let *A* and *B* be two events of non-empty sample space *S* such that *P(A) > 0, P(B) > 0*. Then

$$P(A^c|B) = 1 - P(A|B).$$

7.7. Multiplication Law of Probability

Let *A* and *B* be two events of non-empty sample space *S* such that *P(A) > 0, P(B) > 0*, then

$$P(A \cap B) = P(A|B) \cdot P(B)$$

and

$$P(B \cap A) = P(B|A) \cdot P(A).$$

That is,

$$P(A \cap B) = P(A|B) \cdot P(B) = P(B|A) \cdot P(A).$$

Multiplication law of probability is also known as ***Law of Compound Probability***.

Proof:

Suppose there are *n* elements in the sample space and n_1, n_2, n_3 elements in the events $A \cap B^c$, $A \cap B$, $A^c \cap B$ respectively.

Then,

$$P(A) = \frac{n_1 + n_2}{n}, P(B) = \frac{n_2 + n_3}{n} \text{ and } P(A \cap B) = \frac{n_2}{n}.$$

To find the *P(A|B)*, we keep in view *B* as a sample space. In *B* there are $n_1 + n_3$ elements out of which n_2 are in favour of *A*.

$$P(A|B) = \frac{n_2}{n_2 + n_3} \text{ and } P(A \cap B) = \frac{n_2}{n}.$$

Therefore,

$$P(A|B) \cdot P(B) = \frac{n_2}{n_2 + n_3} \cdot \frac{n_2 + n_3}{n} = \frac{n_2}{n} = P(A \cap B).$$

That is,

$$P(A \cap B) = P(A|B) \cdot P(B)$$

Similarly, one can prove

$$P(B \cap A) = P(B|A) \cdot P(A).$$

Result 1:

For three events *A*, *B* and *C* of the sample space *S*,

$$\begin{aligned} P(A \cap B \cap C) &= P(A) \cdot P(B|A) \cdot P(C|(A \cap B)) \\ &= P(A) \cdot P(C|A) \cdot P(B|(A \cap C)) \\ &= P(B) \cdot P(A|B) \cdot P(C|(A \cap B)) \\ &= P(B) \cdot P(C|B) \cdot P(A|(B \cap C)) \\ &= P(C) \cdot P(A|C) \cdot P(B|(A \cap C)) \\ &= P(C) \cdot P(B|C) \cdot P(A|(B \cap C)) \end{aligned}$$

Result 2:

For three events *A*, *B* and *C* of the sample space *S*,

$$P[(A \cup B)|C] = P(A|C) + P(B|C) - P[(A \cap B)|C]$$

Example:

A random sample of 200 students is classified below by sex and experiencing hypertension during examination period. If a student is selected at random from this sample, find the probability that the student is

(a) male given that the student is experiencing hypertension,
(b) experiencing hypertension given that the student is female,
(c) female given that the student is experiencing no hypertension.

Status \Gender	Male	Female	Total
Hypertension	60	45	105
No Hypertension	40	55	95
Total	100	100	200

Solution:

Let *M* be the event that male student is chosen, *F* be the event that female students is chosen, *H* be the event that the one chosen experiences hypertension and *N* be the event that the one experiences no hypertension.

(a) Probability that the student is male given that he is experiencing hypertension is

$$P(M|H) = \frac{P(M \cap H)}{P(H)}$$
$$= \frac{60/200}{105/200} = \frac{60}{105}.$$

(b) Probability that the student is experiencing hypertension given that the student is female

$$P(H|F) = \frac{P(H \cap F)}{P(F)}$$
$$= \frac{45/200}{100/200} = \frac{45}{100}.$$

(c) Probability that the student is female given that she is experiencing no hypertension is

$$P(F|N) = \frac{P(F \cap N)}{P(N)}$$
$$= \frac{55/200}{95/200} = \frac{55}{95}.$$

7.8. Independent Events

Events are said to be independent if occurrence or non-occurrence of any one of them does not depend on that of any of the remaining ones.
In particular, an event *A* is said to be independent of another event *B* if *P(A|B) = P(A)*.
This definition is meaningful only if *P(A|B)* is defined, i.e., *P(B) > 0.*

7.9. Multiplication Law of Probability For Independent Events

If A and B are events such that *P(A) > 0*, *P(B) > 0*, then *A* and *B* are independent if and only if

$$P(A \cap B) = P(A) \cdot P(B).$$

i.e., for independent events *A* and *B*, the probability that both of these occur simultaneously is the product of their respective probabilities.

Note:

Three events *A*, *B* and *C* of the sample space *S*, are **mutually independent** if

a. $P(A \cap B) = P(A)P(B)$
b. $P(B \cap C) = P(B)P(C)$
c. $P(A \cap C) = P(A)P(C)$
d. $P(A \cap B \cap C) = P(A)P(B)P(C)$

If only first three conditions are satisfied, then *A*, *B* and *C* are **pair-wise independent**.

Result 1:

If *A* and *B* are independent events of the sample space *S*, then

a. *A* and B^c are independent
b. A^c and *B* are independent
c. A^c and B^c are independent

Example:

Jaydev and Vijay can solve respectively 60% and 80% problems in a book. They try independently to solve a problem randomly selected from the book. Find the probability that (i) problem is solved, (ii) only Jaydev can solve the problem, (iii) only Vijay can solve the problem, (iv) none of them can solve the problem.

Solution:

Let *A* be the event that Jaydev can solve the problem, and *B* be the event that Vijay can solve the problem. Hence $P(A) = 0.60$ and $P(B) = 0.80$.
As Jaydev and Vijay solve the problem independently, we have

$$P(A \cap B) = P(A) \times P(B) = 0.60 \times 0.80 = 0.48.$$

(i) The problem is solved if both Jaydev and Vijay solve the problem. Thus the required probability is

$$\begin{aligned} P(\text{Problem is solved}) &= P(A \cup B) \\ &= 1 - P(A^c \cap B^c) \\ &= 1 - P(A^c)P(B^c) \\ &= 1 - (0.40)(0.20) \\ &= 1 - 0.08 = 0.92 \end{aligned}$$

(ii) The probability that only Jaydev can solve the problem is $P(A \cap B^c)$.

$$\begin{aligned} P(A \cap B^c) &= P(A \cap B^c) \\ &= P(A)P(B^c) \\ &= (0.60)(0.20) \\ &= 0.12 \end{aligned}$$

(iii) The probability that only Vijay can solve the problem is $P(A^c \cap B)$.

$$\begin{aligned} P(A^c \cap B) &= P(A^c \cap B) \\ &= P(A^c)P(B) \\ &= (0.40)(0.80) \\ &= 0.32 \end{aligned}$$

(iv) The probability that none can solve the problem is $P(A^c \cap B^c)$.

$$\begin{aligned} P(A^c \cap B^c) &= P(A^c \cap B^c) \\ &= P(A^c)P(B^c) \\ &= (0.40)(0.20) \\ &= 0.08 \end{aligned}$$

7.10. SUMMARY

In this unit we have introduced the concept of probability and various definitions of probability, how to compute the probabilities of events using different laws of probability. We have discussed the concept of conditional probability, multiplication law of probability and independence. To understand the theoretical concepts we have cover some numerical examples manually as well as using Python programming language.

Some of the key definitions and properties are introduced in this unit.

- Classical, statistical and axiomatic definitions or probability.
- Various Theorems on probability.
- Some important results on probability
- Examples on probability
- Python code to solve the problems on probability

7.11. REFERENCES

Books

1. Gupta, S.C. and Kapoor, V.K. (2014): Fundamentals of Mathematical Statistics, Sultan Chand & Sons, New Delhi, 12th Edition.
2. Hastie, Trevor, et al. (2009): The Elements of Statistical Learning, Springer
3. Ross, S.M. (2004): Introduction to Probability and Statistics for Engineers and Scientists, Academic Press
4. Navidi, W. (2011): Statistics for Engineers and Scientists, McGraw Hill, Third Edition.

QUESTIONS

Short Answer:

1. Define
   a. Axiomatic Definition of Probability
   b. Statistical or Empirical Definition of Probability
   c. Classical Definition of Probability
2. Given the classical definition of probability. State its limitation.
3. State the addition law of probability for two events.
4. State the addition law of probability for three events.
5. What is the probability of impossible event?
6. What is the probability of certain event?
7. Determine the type of events in each case
   a. $P(A \cap B) = 0$
   b. $P(A \cup B) = P(A) + P(B) =1$
   c. $P(A \cup B) = P(A) + P(B)$.
8. Define conditional probability.
9. State the multiplication law of probability.
10. State the condition for independence of two events.
11. A^c and B^c are independent events. What can you say about the events A and B?
12. State the conditions for pair-wise independence of three events.
13. State the conditions for mutually independence of three events.

Long Answer:

1. Prove the addition law of probability.
2. With the usual notations, prove that
   a. $P(\phi) = 0$
   b. $P(A^c) = 1 – P(A)$
   c. $P(A \cup B) = P(A) + P(B) – P(A \cap B)$
   d. For two mutually exclusive events A and B, $P(A \cup B) = P(A) + P(B)$

   e. Two or more heads
   f. More heads than tails.

3. A die is thrown once. What is the probability that
   a. a number 5 appear on the top,
   b. an even number appear on the top,
   c. an odd number appear on the top,
   d. a number less than 0 appear on the top,
   e. a number greater than or equal to 0 appear on the top.
4. Prove multiplication law of probability.
5. *A* and *B* are independent events. Prove that A^c and B^c are independent.
6. A^c and B^c are independent events. Prove that *A* and *B* are independent.
7. Two cards are drawn without replacement from a pack of 52 cards. What is the probability that (i) Both are drawn are red, (ii) First is king and second is queen, (iii) One is red and other is black?
8. A charted accountant applies for a job in two firms X and Y. He estimates that the probability of his being selected in firm X is 0.7, and being rejected at Y is 0.5, and the probability of at least one of his application being rejected is 0.6. What is the probability that he will be selected in one of the firms?
9. A die is thrown twice. Let *A, B, C* denote the following events: *A* = {(a, b) | a is odd}, *B* = {(a, b)| b is odd} and *C* = {(a, b) | a + b is odd}. Check whether *A, B* and *C* are independent or independent in pairs only.
10. Two dice are thrown. Write the sample space. Let *A* be the event that the sum of the numbers on the top of dice is 7, let *B* be the event that sum of the numbers on the top of dice is at most 7. Determine the probability of $A \cap B$, $A \cap B^c$, $A \cup B$, $(A \cup B)^c$.
11. $P(A) = 2P(B) = 3\ P(B|A) = 0.6$. Find the probability that (i) *B* does not occur, (ii) exactly one of *A* and *B* occurs, (iii) not more than one of *A* and *B* occurs, (iv) *A* and *B* do not occur together and (v) neither of *A* and *B* occurs.

PRACTICALS

1. Write a python code to define solve the example 9 from the above Long Answer questions.
2. Write a python code to define the sample space (*S*) for an experiment of tossing three coins. Let *A* be the event that at least two head occurs and *B* be the event thatat least one tail occurs. Determine $P(A)$, $P(B)$, $P(A \cup B)$ and $P(A \cap B)$.
3. Write a python code to define solve the example 10 from the above Long Answer questions.
4. Let the sample space be *S* = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. Let *A* = {2, 4, 6, 8, 10}, *B* = {1, 3, 5, 7, 9}, *C* = {3, 6, 9}, *D* = {5, 10}. Write python code to
   a. Determine $P(A \cup B)$, $P(A \cap B)$, $P(A \cap C)$, $P(A \cap D)$ and $P(A – D)$.
   b. Find $P(A^c)$, $P(C^c)$.

M.Sc. (Computer Science)

Probability & Statistical Analysis

Semester 1

UNIT VIII: STATISTICAL INFERENCE

STRUCTURE

8.0 OBJECTIVES

a. To elaborate Statistical Inference
b. To discuss Random variables
c. To discuss and Probability and Density Functions
d. To explore Moments in Python

8.6 INTRODUCTION TO STATISTICAL INTERFERENCE

In python, statistical inference is used to draw and infer conclusions from set of values in given dataset. Initially, some random samples are used and extracted from given population which is used to describe and draw relevant inferences (conclusions) for entire population (Vondrejc, 2019). Thereare numerous Inferential Statistics available which can be used along with Python and are named as:

a. Z Test and Z Scores
b. t-Test
c. F-Tests
d. Correlation Coefficients
e. Chi square

8.1.1 Need of Preliminary Libraries

Distinct preliminary set of libraries is required to get imported in Python when one wants to work with Arrays, Data frames and different tools for performing statistical analysis. Numpy and Pandas come under this category. These both are highly important and different packages where Numpy canbe used to perform distinct operations on Arrays whereas another platform Pandas is used to carry different operations on Data frames. These two libraries can be imported as:

```
import numpy as np

import pandas as pd
```

8.1.2 Z Scores and Z-test

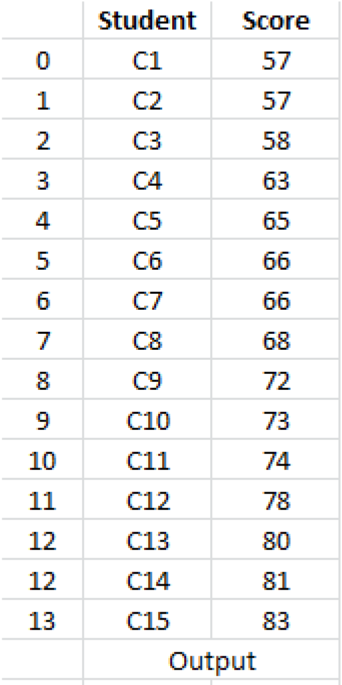
- **Z Scores**

Z Scores computes the probability of score which are calculated from normal distribution. This helps in comparing scores for two or more given normal distributions

- **Z Value**
- **Importing Dataset**

Here, dataset which contains exam scores for some of the students has been imported:

```
Z_scoresdata=pd. Read_excel(C:/Users/user/Desktop/Datasets/marks_Score.xls")

Z_scoresdata
```



	Student	Score
0	C1	57
1	C2	57
2	C3	58
3	C4	63
4	C5	65
5	C6	66
6	C7	66
7	C8	68
8	C9	72
9	C10	73
10	C11	74
11	C12	78
12	C13	80
12	C14	81
13	C15	83

Output

- **Z Score Calculation**

Following is the code by which Z scores can be calculated. Here, Z Scores have been calculated for a column with name „Score‟ column of Z Score dataset. The function „ddof‟ can be used to alter the divisor sum of squares of mean sample. Initially, ddof holds 0 but for std we can use ddof=1

```
Z_scoredata[„scores_zScore‟]=(z_scoresdata[„score‟]-z_scoredata[„score‟].mean())/z
_scoredata[„Score‟].std(ddof=1)
```

- **Z Score calculation in Python—**

```
import numpy as np

from scipy import stats

Arayr1= [[19,3,6,2,35],

[49,11,12,35,5]]

Array2 = [[49,11,11,34,5],

[12,10,10,34,22]]

Print (“array1:”, array1)

Print (“\narray2:”, array2)

Print (“z score for array1:, stats,zscore(array1))
```

Print("\nz score for array1:", stats.zscore(array1,axis=1))

- **Calculating Percentage**

For above considered example, percentage of people scored more than 70 can be calculated. Mean and Standard Deviation is to be taken for calculating area under curve. The code that can be used to find area under curve can be depicted as:

Cutoff= 70

Print(1-(scipy.stats.norm(70.5, 7.06).cdf(70)))

Where 7.06 is standard deviation and 70.5 is mean for finding percentage.

- **Z Test**

This test determines whether the given two datasets are similar or not.

- **Importing dataset**

By taking initial dataset from population and some random samples of this dataset, importing dataset is initial set which can be done in following way:

HghtDataPop=pd.read_csv("C:/User/ABC/Datasets/Heightof100ppl.csv")

Now, by taking random sample of above mentioned dataset:

HghtdataSample= pd.read_excel("C:/User/ABC/Datasets/HeightdataSample.xls")

- **Package to be imported for applying Z Test-**

from statesmodel.stats.weightstats import ztest

- **How to run Z Test**

Ztest(A2,b2=None,value=mean1)

- **Code for implementing Z test in Python**

```
    # considering an array of 55 numbers with mean 105 and std dev 16
import math
import numpy as np
from numpy.random import randn
from statsmodels.stats.weightstats import ztest
    mean = 105
    std_dev = 16/math.sqrt(55)
    alpha =0.01
    nul_mean =101
    data = sd_iq*random(55)+mean
    # printing mean and std_dev
    print('mean=%f std_dev=%f' % (np1.mean(data), np1.std(data)))
    # Z test
    ztest_Score, p1_value= ztest(data1,value = null_mean, alternative='larger')
     if(p1_value < alpha):
    print("Rejecting Null Hypothesis")
    else:
print("Failed to Reject Null Hypothesis")
```

8.1.3 t Test

t-Test is used to evaluate similarity level of groups. This can also be done by using Z test, the difference is Z Test is better to apply for the case sample size greater than 30 whereas t-Test is used for sample size less than 30.

- **Importing package for applying t-Test**

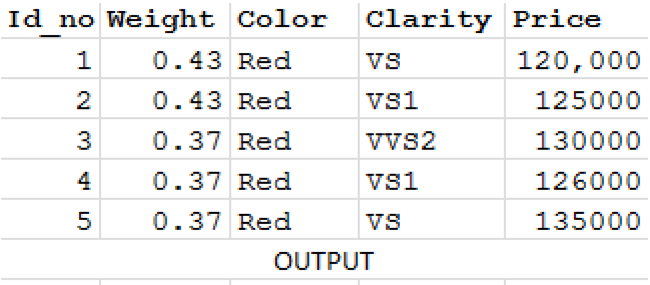
Import scipy.stats as stats

- **Importing Dataset**

For instance, considering hypothetical *Rubyjewellery* dataset which contains all necessary information of Rubyjewellery, which is sold in a store of jewellery.

Rubyjewellery-pd.read_excel("C:/Users/Login/Rubyjewellery.xls")

Rubyjewellery



Id_no	Weight	Color	Clarity	Price
1	0.43	Red	VS	120,000
2	0.43	Red	VS1	125000
3	0.37	Red	VVS2	130000
4	0.37	Red	VS1	126000
5	0.37	Red	VS	135000

OUTPUT

- **Code for t-test in Python**

```
from numpy.random import randn
from scipy.stats import ttest_ind
seed(1)
data_1 = 4 * randn(105) + 51
data_2 =4 * randn(105) + 52
# comparing samples
Stat1, p1 = t_test_ind(data_1, data_2)
print('t=%f, p1=%f' % (sta1t, p1))
```

8.1.4 F –test

F test is a statistical test which has F-distribution under null hypothesis. F test uses F statistics which actually is the ratio of two different variances; hence, they use F Distribution. Such test is applicable in comparing two regression models to check for statistical significance. Moreover, unlike Z and t-Test, where comparison is done on two datasets for inferring results, in F Test compares two variances.

8.2 CONCEPT OF RANDOM VARIABLE

A random variable in Python Statistical Inference is a variable whose values are possibly theoutcomes of random process (Julier, 2004). There are two main categories of random variables such as discrete and continuous variables.

8.2.1 Discrete Random Variable

The variable which takes only one countable number of different numbers and values and also which can be quantified. For instance, ***A*** to be possible number which comes up when rolling a fair dice. ***A*** can take different values: [1, 2, 3, 4, 5, 6]. Hence, is a discrete random variable.

- **Probability Distribution:**

It is associated with list of probabilities along with each possible value. It is also refereed as *probability function.* Mathematically, it can be interpreted as, suppose there is random variable ***Y*** which may take ***n*** different values with probability that $\boldsymbol{Y = y_i}$ can be defined as $\boldsymbol{P(Y = y_i) = p_i}$**.** In this case, probability pi need to satisfy following conditions:

1. $0 \leq pi \leq 1$ for every i

2. $p1+p2+p3+\ldots..+ pn = 1$

Bernoulli distribution, Binomial distribution and Poisson distribution can be considered as best examples of Discrete Probability Distributions.

8.2.2 Continuous Random Variable

The variable which can take infinite number of values is continuous random variable. For example, Y is taken as variable to store the height of players in a group. Such variables are interpreted as continuous random variables over interval of class, such variables are inferred by using area under curve or by integrals. Distribution of these variables is also interpreted as probability distribution functions. It can be demonstrated by using p(x), which must also satisfy following condition:

1. When curve is not having negative values p(x)>0
2. When area under curve =1

A curve that meets such requirements is taken as a density curve. Normal distribution, Exponential distribution and Beta distribution are common examples of Continuous Probability Distributions .

8.2.3 Importing some important distributions in Python

- **Uniform Distribution**

In this distribution all the outcomes of probability distribution are equally like. In Python, this distribution can be visualized by importing *uniform function* from *scipy.stats* module as:

```
    # import uniform distribution
  from scipy.stats import uniform
    # random numbers
    N=1000
    Start=5
    Width=10
Data_uniform= uniform.rvs (size=n, loca=start, scale=width)
```

- **Code for uniform distribution in Python can be written as:**

```
From numpy import random
A=random.uniform(size(3,3))
Print(a)
```

- **Normal Distribution**

It is also considered as bell curve and occurs quite naturally in various situations. It is also called as Gaussian distribution and most commonly applicable in statistical inference. It poses bell shaped density curve and is defined by mean (μ) and standard deviation (σ).Here, the density curve is symmetric and centered about its mean. In Python, it can be imported as:

from scipy.stats import norm

Data1_normal = norm.rvs (size=1000, loca=0, scale=1)

Where *scipy.stats* is a relative module, *norm.rvs ()* method, *loc* is mean of distribution, *scale* is standard deviation and *size* is total number of random variables. Random normal distribution of size (3*3) can be generated by:

From numpy import random

A= random.normal(size=(3,3))

Print(a)

- **Gamma Distribution**

It belongs to two-parametric family under continuous probability distribution. It is used rarely. Most of the times it is used in modeling in engineering areas where the variables are always positive with skewed results. This can be imported in Python as:

from scipy.stats import gamma

Data1_gamma = gamma.rvs (x = 10, size_s= 1000)

- **Exponential Distribution**

This distribution is used to describe the time interval between different events in a process of Poisson point i.e., the process in which any event occurs in a continuous and independent way with constant and average piece of rate. In python it can be imported as:

from sipy.stats import expon

Data1_expon = expon.rvs (scale=1, loca=0, size = 1000)

- **Poisson Distribution**

This distribution is used to show the event occurrence within specific period of time. For instance, users visited a website in a particular interval is a Poisson Process in itself. An event can occur n number of times and average number of events in an particular interval is depicted as lambda λ. Poisson distribution can be imported in Python as:

```
from scipy.stats import poisson
```

```
Data_poisson = poisson.rvs (mu= 2, size, 1000)
```

- **Bernoulli Distribution**

In this distribution, only two outcomes are expected to come such as failure or success and the probability of each success and failure is entirely same for all number of trials and is known as Binomial distribution. In Python, this distribution can be added as:

```
from scipy.stats import binom
```

```
Data1_binom = binom.rvs (n = 20, p = 0.5, size = 1000)
```

8.3 PROBABILITY MASS FUNCTION

This function describes the probability which is associated with given random number y (Vondrejc, 2019). The function is depicted as P(y).

Example, when rolling a die infinitely and looking on proportion 1, then 2 and so on. The random variable can correspond to outcome of a dice roll. So, random variable can take following discrete values i.e., 1, 2, 3, 4, 5 or 6. The main aim of the probability mass function is to describe possibility/probability of every possible value. In this example, there is probability to get 1, 2 and so on. For example in rolling of dice, there is equal probability of getting any number, which can be written as:

$$P(y = 1) = P\ (y = 2)$$

$$=P(y=3)$$

$$=P(y=4)$$

$$=P(y=5)$$

$$=P(y=6)$$

So, 1/6.

In this way, distribution demonstrates similar probability for every possible value, and called as uniform distribution. Probability mass function looks like:

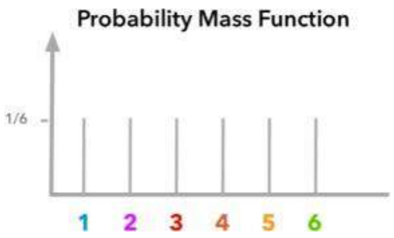


Fig. 1 Probability mass function (for dice)

Here, x-axis depicts outcome and y axis depicts probability. The code that can demonstrate this in more clear way and can be written as:

```
Numb_throws=1000

Outcomes=np.zero(numb_throws)

For I in range(numb_throws);

#when rolling a dice

Outcome=np.random.choiceA(„1‟,‟2‟,‟3‟,‟4‟,‟5‟,‟6‟)

Outcome[i]=outcome

val.cnt=np.uniquee(outcomes, return.count=True)

prop=cnt/len(outcomes)

# after rolling dice 1000 times, plotting of results

Plt.bar(val,propi)

Plt.xlabel("outcome")

Plt.ylabel("probability")

Plt.show()

Plt.close()
```

8.3.1 Properties of Probability Mass Function

Probability Mass function, if and only if,

$$\forall x \in x,\ 0 \leq P(x) \leq 1$$

Here, symbol ∀ depicts for any or for all. This represents, for every x within range of x (in case of rolling dice, set of possible values are 1, 2,3, 4,5 and 6). The probability of each outcome can vary between 0 and 1. Here, 0 represents event has not occurred and 1 represents event has occurred. In particular case of dice, the probability of each value is 1/6 i.e., between 0 and 1.

8.3.2 Probability Density Function

All the discrete variables cannot take infinite values at a certain period of time. Still, it needs to state probability which is associated with all possible outcomes. The equivalence of probability mass function with continuous variable is known as **Probability Density Function.**

- **Important property of Probability Density Function**

$$\forall x \in x, p(x) \geq 0$$

Here, p(x) needs not to be less than 1, because it is not corresponding to probability.

8.4 MATHEMATICAL EXPECTATION

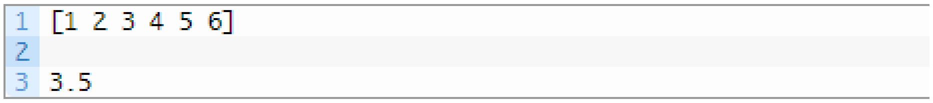
Probability describes happening of certain events and occurrence of particular event depends upon previous experience. The Mathematical Expectation represents all those events which are almost impossible for any experiment. Probability for that particular event remains 0 and Probability of an event remains 1 where both of the numerator and denominator remain equal.

Mathematical Expectation is also referred as expected value, that uses the notation E[X]. This can be computed as probability weighted sum of all possible values which can be drawn by as follows:

E[X]= sum (x1*p1, x2*p2*………*xnpn)

Following is the code with 6-element vector and calculation of mean:

```
From numpy import array

From numpy import mean

M1 = array_example ([1,2,3,4,5,6])

Print (M1)

Result = mean (M1)

Print (result)
```



```
[1 2 3 4 5 6]

3.5
```

- **Properties of mathematical Expectation-**

1. For two variables X and Y, total sum of these two variables is equals to sum of mathemeatical expectationof X and Y respectively i.e., E(X+Y)= E(x)+E(Y).

2. For two independent variables, mathematical expectation would be the product of those considered variables.

3. Mathematical Expectation of constant sum and considered function of any random variable is equals to sum of that constant of function of given random variable.

4. The mathematical expectation of sum and product of a constant and function of any random variable and another constant is all equals to sum of product of that constant and mathematical expectation of given function and random variable and constant.

5. Linear combination of all random variables and a constant is all equals to sum of product of n constants and mathematical expectation of all n numbers.

8.5 MOMENTS

Moments are used to calculate the n moments about mean for given sample i.e., all the array elements with particular axis of that given array (Fan, 2016). In Python, Moments can be calculated as:

```
From scipy import stats

import numpy as np

arry = np.array ([1,27, 31, 21, 13, 9],[12,8,4,8,7,10])

print (“0th moments is :”, stats.moment(arry,moment=0))
```

8.5.1 Moment generating function and characteristic function

```
   from random import choice
   import matplotlib.pyplot as plt
   import numpy as np
   exp_value = lambda values: sum(values) / len(values)
   std_deviation = lambda values, exp_value: np.sqrt(sum([(v - exp_value)**2 for v in values]) /
   len(values))

muu, sigmaa = 40, 1
   population = np.random.normal(muu, sigmaa, 100000)
   mean = expected_value(population)
```

```
    print(
'''population: Expected_value: {0} Standard_deviation: {1}
       '''.format(mean, standard_deviation(population, mean))
    )
    plt.hist(population, 70, density=True)
    plt.show()

    randomly_select_items = [choice(population) for _ in range(63)]
    mean = exp_value(randomly_selected_items)
    s_d = std_deviation(randomly_selected_items, mean)
    print(format(mean, s_d))
  plt.hist(randomly_selected_items, 10, density=True)
    plt.show()
    xsss = np.arange(63, 44, 0.001)
    actual_ys = norm.pdf(xs, muu, sigmaa)
    ys = norm.pdf(xs, mean, s_d)
    plt.plot(xs, actual_ys, label='actual_population_distribution')
    plt.plot(xs, ys, label='sample_distribution')
    plt.legend()
    plt.show()
```

8.6 PRACTICE QUESTIONS

Q1. What is Statistical Inference?

Q2. What are different Inferential Statistics available can be used along with Python? Q3. What are random variable?

Q4. Explain difference between F-test and t-test along with suitable example.

Q5. How Z scores and Z test is calculated. Demonstrate with suitable dataset.

Q6. How moments can be imported and calculated in Python.

Q7. Explain different properties of Mathematical Expectation.

Q8. Explain different distributions. Which distribution is frequently used and why?

REFERENCES

[1] Vondrejc, J. & Matthies, H. G. (2019) “Accurate Computation of Conditional Expectation for Highly Nonlinear Problems”, ,SIAM/ASA Journal on Uncertainty Quantification 7(4), pp. 1349-1368.

[2] Julier, S. J. & Uhlmann, J. K. (2004) “Unscented Filtering and Non linear Estimation”, Proceedings of IEEE, 92, pp. 401-422.

[3] Ale, A. , Kirk, P. & & Stumpf, M.P. (2013) “A General moment expansion method for stochastic kinetic models”, Journal of Chemistry Physics, 138 (17).

[4] Fan, S. , Geissmann, Q. , Lakatos, E. & Lukauskas, S. (2016) “Means: Python package for Moment Expansion Approximation, Inference and Simulation”, Bioinformatics, 32 (18).

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University
(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM
ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-I
Course: FUNDAMENTALS OF DIGITAL MARKETING (MSCS-1-04T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

M.Sc. (Computer Science) Programme Coordinator:
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:
Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Business Management & Commerce:
Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Social Sciences and Liberal Arts:
Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

\

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA
A State University Established by Govt. of Punjab vide Act No. 19
of 2019 and Approved Under section 2(f) of UGC

PROGRAMME COORDINATOR

Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE COORDINATOR & EDITOR

Dr. Pooja Aggarwal (Assistant Professor)
School of Business Management and Commerce
Jagat Guru Nanak Dev Punjab State Open University

Course: Digital Marking	
Course Code: MSCS-1-04T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Identify the core concepts of digital marketing and its role in businesses or organizations.
CO2	Understand digital marketing strategies to reach the target audience.
CO3	Analyze marketing approaches and recognize areas for enhancing performance.
CO4	Resolve digital marketing issues and offer solutions based on the vital examination of digital marketing information.
CO5	Work on social media platforms such as Twitter, Facebook and Instagram

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA
A State University Established by Govt. of Punjab vide Act No. 19
of 2019 and Approved Under section 2(f) of UGC

M.Sc. (Computer Science)
Semester-1
MSCS-1-04T: Fundamentals of Digital Marketing

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

INSTRUCTIONS FOR THE PAPER SETTER/EXAMINER

1. The syllabus prescribed should be strictly adhered to.
2. The question paper will consist of three sections: A, B, and C. Sections A and B will have four questions from the respective sections of the syllabus and will carry 10 marks each. The candidates will attempt two questions from each section.
3. Section C will have fifteen short answer questions covering the entire syllabus. Each question will carry 3 marks. Candidates will attempt any ten questions from this section.
4. The examiner shall give a clear instruction to the candidates to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.
5. The duration of each paper will be three hours.

INSTRUCTIONS FOR THE CANDIDATES

Candidates are required to attempt any two questions each from the sections A and B of the question paper and any ten short q questions from Section C. They have to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.

SECTION-A

Unit-I: Introduction to Digital Marketing: Concepts, Traditional Marketing vs. Digital Marketing, Digital Market Evolution, Career in Digital Marketing

Unit-II: Digital Consumer: Consumer Characteristics and profiles, Information Search Behavior, Factors Influencing Consumption Behavior

Unit III: Purchase: Decision Process, Post Purchase Behavior and Management.

Unit IV: Digital Marketing Strategy: Digital vs Non- Digital Marketing Strategy, Digital Marketing Decisions- Product, Price, Distribution and Promotion

SECTION-B

Unit V: Digital Marketing Strategy: Formulation and Execution, Digital Marketing Mechanisms: Websites- Company and Retail Service Providers, Search.

Unit-VI: Engines: Google, Bing, Ask, Yahoo Video Hosting and Entertainment- You tube, Vimeo, Amazon Prime, Netflix, Hot star.

Unit-VII: Mobile Phones, E-Mails, Blogs

Unit VIII: Social Media: Facebook, Instagram, Twitter, WhatsApp.

Reference Books:

- Seema Gupta, “Digital Marketing”, Tata McGraw Hill Education
- Punit Singh Bhatia, “Fundamentals of Digital Marketing”, Pearson
- Philip Kotler, “Marketing 4.0: Moving from Traditional to Digital”, Wiley Publisher

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT 1- INTRODUCTION TO DIGITAL MARKETING

STRUCTURE

1.0 OBJECTIVES

After studying the Unit, you would be able to

- Understand the Digital Marketing Concept and how it’s different from online marketing
- Explore principles required for effective Digital Marketing
- Find out the origin of digital marketing and its different types

- Identify and manage the problems faced by marketers in Digital market
- Explain the Pros and Cons of Digital marketing
- Highlight the reasons for the growth of digital marketing

1.1 INTRODUCTION

World is changing at a remarkable pace and so does the world of marketing. Marketing is a simple process of creating awareness of products or services through various channels towards identified market. Marketing makes it easier to target certain group wherever and whenever. Earlier it was very difficult for the consumers and producers to buy and sell goods and services but now with the advancement of technology, it has become much easier than before in terms of time, money and efforts involved. Now consumers can purchase any product at any time while sitting at any place or any corner of the world. Consumers can get all details of the product. Digital Marketing has made the efforts of producers easier as they can upload all the details of their product and easily market their product. There is no worry about cost incurred on marketing a product.

Today, businesses operate in a dynamic environment, in which various kinds of competitions determine the rule of the game. Managing a competitive edge in today's business world demands an effective electronic strategy, concentrating on the convenience of electronic buying. Internet has converted traditional marketing into online and digital marketing. These days the world is fully competitive so every business wants to go on for long time. Many new techniques are used by different organizations to advance their brands all over the world. In order to promote, companies advertise their products on various websites. People usually do online shopping from these sites. Say if any consumer wants to buy any product, they can visit the websites of E-Marketers and after making comparison, place the orders.

Digital marketing was first used in 1990s as that was the phase when first search engine got emerged by a company called Hot wired.Then in 1998 another engine company got popped up that is "Google" as it gets more popular search engine in the world. After that the history of Digital marketing had been changed when WEB 2.0 came in the market.

1.2 DIGITAL MARKETING

The rapid adoption of the Internet as a commercial medium has caused marketers to experiment with the innovative ways of marketing thus changing the pattern of marketing strategies. "Marketing consists of individual and organizational activities that facilitate and expedite satisfying exchange relationships in a dynamic environment through the creation, distribution, promotion and pricing of goods, services and ideas". It is said that Digital Marketing first began in the beginning of 1990 with just text-based sites which provided product information. In the beginning, the Internet was cool but with the explosion of internet growth, online marketing has started becoming popular. Digital Marketing creates additional customer value by transforming marketing strategies during effective segmentation, targeting, differentiation and positioning strategies.

The basic purpose of marketing is to satisfy the customers profitable by offering the right product to right customer at right time, the only that has changed is the methods, as Smith and Chaffey (2001) have remarked that "Digital Marketing is *Achieving marketing objectives through applying digital technologies".*

Digital Marketing is promotion of goods and services through the use of online mediums like the use of internet, use of mobile phones or any the use of any other technology which can be used to advertise a product or services, for example; Email Marketing, Web Marketing, Internet Marketing, Social media marketing, etc. Digital Marketing has made work much easier, earlier physical presence of buyer and seller was utmost importance in buying or selling, but, in online marketing such kind of drawback has been overcome.

Digital Marketing is the result of information technology applied to traditional marketing. The rapid growth of the internet and the popularity of the computers and electronic devices all over the world have been making the marketers scramble to determine how best they can use the technology for the business profitability. Marketers need to understand the technology, and how they can use the technology in the best possible way for the promotion and profitability of business. Digital Marketing is the application of a broad range of information technologies in the marketing functions to achieve the followings:

- ✓ Development of the products which has value for the customers and is possible to be marketed through online channels.
- ✓ Transformation of marketing strategies and policies to create more customer value through effective segmentation, targeting, differentiation and positioning strategies.
- ✓ Planning and execution of marketing conception, distribution, promotion, and pricing of goods and services, and ideas.

- ✓ Creating customers and satisfying the individuals and business customer’s needs and wants.

Digital marketing increases the efficiency in all traditional marketing functions. Digital marketing uses the technology for transforming the marketing strategies in new business models that add customer value and increase company profitability.

For example, Swiggy and Zomato created an App as it helps the shopkeepers to reach to their customers so that they can order easily anything at any place as they want according to their needs. They also have their rating page to enhance their market share with online Marketing Apps.

1.3 DIGITAL MARKETING AND ONLINE MARKETING

In reality there is not much difference, both embrace the same methodologies. Primarily Digital Marketing takes the principles of traditional marketing and translates this to the online world. One key benefit being that the nature of ‘online’ removes the geographic boundaries that limit traditional media marketing. Digital Marketing or Online Marketing can take many forms, and continues to develop with new innovations all the time, the most common methodologies include;

SEO

SEM

APP DEVELOPMENT

WEB MARKETING

VIDEO MARKETING

EMAIL MARKETING

CONTENT MARKETING

SOCIAL MEDIA MARKETING

BRANDING

GAME ADVERTISING

PAY PER CLICK

In today's business, Digital Marketing (or online advertising as it is often referred) has become an essential part of most companies' online marketing mix. Year on year companies is allocating increasingly high resources to advertising and marketing online and this is hardly surprising as the shift from offline to online continue to grow. Schemes such as 'laptops for everyone' and increasing end user awareness of the benefits of shopping online have contributed substantially to the rise business online. This in turn has created a shift from traditional marketing to Digital Marketing.

1.4 PRINCIPLES OF DIGITAL MARKETING

Digital Marketing is spreading awareness of products and services with the help of digitalized technologies, specifically online. Growth of digital marketing since 1990's and 2000's has revolutionized the way the companies and brands use technology for advertising. As we know the social media platforms or other digital platforms are increasingly used in the market on daily basis. Now, humans use these technologies inspite of going to physical shops, so these campaigns are more productive, effective and easy. The Basic principles of Digital Marketing are:

- **Build and actively manage a customer database:** In this era of scarce customers, companies need to capture the names of customers and useful information that is possible to collect about potentially valuable prospects and customers. A rich customer database can provide the company with a strong competitive advantage. The company can search and rate different groups and individuals for their probability of responding to a given offer or highly tailored offers. A database permits a company's targeting to be super-efficient.
- **Develop a clear concept on how the company should take advantage of the Internet:** A company can develop presence on the Internet in at least seven ways. The company can use the Internet to do research, provide information, run discussion forums, provide training, carry on-line buying and selling (i.e., e-commerce), provide on-line auctioning or exchanging, and even deliver "bits" to customers.

The company's Web page must be appealing, relevant, and current if it is to attract repeat visits. Companies should consider using state-of-the-art graphics, sound, and video, they should also need to add weekly news or features. The site can be developed to provide valuable help, such as links to a map showing the location of the hotel or restaurant.

- **Be easily accessible and quick in responding to customer calls:** Customers mostly have high and rising expectations about how quickly and adequately they should receive answers to questions and complaints sent in by phone or e-mail. Make sure the Internet user can communicate directly with the company. One advantage of the Internet is that we can communicate automatically. The computer can be programmed to book reservations, select and confirm seat assignments on airlines, and send confirmations of reservations, changes in flight plans and other information to the customer. However, when the user has a question that the computer cannot answer or they have a problem they would like to discuss, they should be given a phone number to call and an automatic e-mail option. Majority of companies' websites have the goal of having 100 per cent electronic communication, and they do not include telephone contact information.

CHECK YOUR PROGRESS

I. Earlier it was very difficult for the consumers and producers to buy and sell goods and services but now with the advancement of technology. (True/False)

II. Today, businesses operate in a dynamic environment, in which various kinds of competitions do not determine the rule of the game. (True/False)

III. Digital Marketing is the result of information technology applied to …………… marketing.

IV. Digital marketing was first used in ………… as that was the phase when first search engine got emerged by acompany called Hot wired.

a. 1991 c. 1981

b. 1990 d. 1980

V. In this era of scarce customers, companies need to capture the names of customers and useful information that is possible to collect about potentially valuable of …………..

a. Advertising Agency c. Prospects and customers

b. Marketing researcher d. Sales men

1.5 ORIGIN OF DIGITAL MARKETING

In this modern era, over 175 million patrons make use of digital media on daily basis, everyone is supposed to be known about digital marketing tools. Soon traditional media platforms will be absorbed and get dominated by digital media. Even more digital media is more cost effective than traditional methods as; large customer base can be reached within a short span of time.

The Digital Marketing came into existence in 1990's. It came over with the introduction of internet and with growth of the web 1.0, whereas web 1.0 is the phrase is known to as beginning of digital media marketing on World Wide Web, that did term as "Websites".

In the year, 1993 clickable banner or web banner came into existence. Clickable banner is basically a type of advertising done with the help of ad server on www. Entrenching an advertisement into a web page is known as web banner so that web traffic could be attracted to the web page by connecting to the web site of the marketer. In the year 1994, new digital tools incorporated in digital marketplace such as Yahoo, it was launched by Jerry Yang. In the very first year, Yahoo, was able to get 1 million thumbs. In year, 1996 various tools such as Alexa, look smart and in year 1998, the most used search engine "Google" came into existence. In year, 2006 customer base was developed up to 6.7 billion in a solitary month.

After then various social sites emerged, Myspace was firstly commenced social networking site, after that Facebook arrived in market followed by Instagram, Snapchat, Twitter etc. which took new opportunities in the market for the businesses, companies and consumers.

1.6 ROLE OF INTERNET

The internet is a global network of interconnected networks. This includes millions of governments, corporate, organizational and private networks forming the World Wide Web (www). The computers in these networks hold files such as web pages that can be accessed by all other networked computers. Computer, mobile phone, and other networked devices like laptops etc., can send and receive data in the form of e-mail or files, over the internet. The data move over phone lines, cables and satellites from sender to receiver. The internet therefore consists of computers with data, users who send and receive the data files and a technology infrastructure to move, create and view or listen to the content. Three important types of networks form part of the internet.

- Intranet: A network that runs internally in a company but uses the internet standards such as HTML and browsers. An intranet is like a mini-internet but only for the internal company use.
- Extranet: An intranet to which value chain partners have admissions for strategic reasons. The access is partial or restricted only.
- Web: The part of internet that supports a graphic user interface for hypertext navigation with a browser such as Netscape or Internet Explorer. The web is what most people think about when they think of the internet.

1.7 TYPES OF E-MARKETS

Digital Marketing is normally carried out in the following types:

- **B2B**: This involves business-to-business marketing. The business organisations make online transactions with other business. They sell the products and services using internet. Also other functions like purchasing, supply chain, services, support and payment systems are managed online.
- **B2C**: This involves business-to-consumer marketing. The goods and services are marketed directly to the consumers by the business organisations. This includes activities like sales, customer service and payment systems.
- **C2C**: This involves consumer-to-consumer marketing, where consumers directly sell products or services to other consumers, using the Internet. Firms like OLX, eBay, Quicker provide such facilities.
- **C2B:** This involves consumer-to-business marketing, where the consumers (individuals) create value and businesses consume that value. For example, when a consumer writes reviews or when a consumer gives a useful idea for new product development then that consumer is creating value for the business if the business adopts the input. Excepted concepts are crowd sourcing and co-creation.

1.8 BARRIERS FOR DIGITAL MARKETING

The problems which were faced earlier were overcome by Digital Marketing but there are some issues which are being there in Online Marketing, which are being faced by both consumers as well as producers.

- Lack of privacy
- More competition
- Online frauds
- Lack of proper knowledge
- Variety of customers
- Internet problems
- Security problems
- Gap between customers and sellers
- Lack of awareness
- Technical problems
- Difficult to engage customers

1.9 ADVANTAGES OF DIGITAL MARKETING

Digital Marketing embraces many of the basic principles of traditional marketing with a few differentiating factors. One key advantage to Digital Marketing is that results can be quantified, allowing the marketers to work more effectively, generate higher ROI and in turn increasing profitability. A well implemented Digital Marketing strategy can achieve a more cost-effective customer acquisition than traditional marketing. The 'reach' of Digital Marketing campaigns means that substantial incremental sales can be achieved online without affecting any other marketing efforts. The Digital Marketing has following advantages over the traditional marketing channels:

- **Digital Marketing can Reach Targeted Customers More Effectively:** Digital Marketing makes it possible to cover the global markets by eliminating the geographical and territorial boundaries. Thus, the location often becomes irrelevant through Digital Marketing. The nature of the internet means a potential customer from anywhere across the globe can make a business transaction and interact with the marketer. Unlike traditional marketing methodologies the power of the web means that prospects and clients can become a part of companies marketing mix 24*7 and 365 days from anywhere in the world.
- **Can Reach Wider or Even International Audience:** Digital Marketing helps in reaching out the customers who can be approached through only online means. Marketing embraces routes to market that simply do not exist with traditional marketing. It's no longer a case of just advertising through search engines and broadcast email shots. The use of new

media such as social networking, viral marketing and other methodologies exclusive to the online world provide access to the groups of customers that would otherwise have not been possible.

- **Cost Effective Marketing:** Utilising online technologies can make a substantial contribution to marketing budgets. By accurately targeting the potential core audience, Digital Marketing often provide the most cost-effective way to increase customer base, raise the brand awareness, increase the demand for product and stay in touch with customers by building long term relations.
- **Full Accountability:** Digital Marketing provides full accountability for its results to the organisations. Online activities are fully tractable and can accurately show ROI (return on investment); this means that marketers can see the value of every Rupee that has been spent on the Digital Marketing activities.
- **Consumer Control:** The internet has already affected a radical shift in the way the media and consumers interact. Traditional media have a one-to-many approach: the media outlet beams its message down to a host of passive consumers. The web, however, relies on many-to-many interaction or one-to-one interaction means: anybody can post content or comment on what they see, and media outlets no longer have complete power over their broadcast message. In the world where everything is social and shared, the consumer has a lot of power and it is likely to further grow in the coming years. Marketing agencies foresee that they will need to hand over even more control to customers, who want to engage on deeper and more significant levels with content. The trend may go so far as letting customers create and mediate Digital Marketing content, with agencies keeping oversight and steering from the side-lines.
- **Marketing can be Personalised as per Customers Need:** Technological devices are shrinking in size, lending themselves more and more to portability. Already, powerful computers can fit inside the wrist watches and cell phones, hide in car dash-boards and meld seamlessly with work and home tabletop surfaces. Consumers will expect to have round-the-clock access to whole of their data, regardless of where they are and what device they are using. Marketing messages will have to be flexible as per the customer requirements. There are many opportunities available here for even more specific and user-relevant content. A device with GPS will know exactly where it is and where the nearest restaurants, cinemas, shops, utilities and user’s friends are; it will store user preferences and recommend nearby locations. Large billboards can gather user data from the people in the area, average out the

demographic statistics and display adverts that are most relevant to the largest number of customers who are passing by.

- **The Rise of Web 3.0:** Nobody is quite sure what web 3.0 will entail, or even whether it has not already started. General consensus is that it will involve the advent of the truly semantic web. The semantic web is a concept that means the exponentially growing amount of data will be categorised in personal and human-usable ways, enabling even better, more meaningful and more accessible content than has been ever possible. In other words, raw data will become real information. Web 3.0 also involved the concepts described above, especially portability, control over marketing and customisation of everything. In the far future, experts even predict the invention of web-based artificial intelligence that will think ahead and source information before a user even knows it is needed. Since web 3.0 is all about "me", marketing must take personalisation and engagement to a new level.
- **Delivers Immediate Results:** One of the most important advantages is the fast availability of the information. The clients/users can easily get information, by navigating the internet, about the products that they wish to purchase, and besides that, they can check the information at any time of the day.
- **Low Investment required:** Digital Marketing allows the companies to save money, an aspect that is really taken into account by the companies since the online marketing campaigns do not require a large amount of investment. The traditional marketing gives less importance to the differences between large and small companies. The Digital Marketing techniques helps in cost savings and thus significant for the companies which cannot incur high expenditure on promotion and marketing of the products and services. The benefits are further passed to the customers.
- **Global Markets:** Presence on the Internet can help the expansion of the company from a local market to national and international markets at the same time, offering almost infinite expanding possibilities.

1.10 DISADVANTAGES OF DIGITAL MARKETING

Digital Marketing may not be useful for all varieties of products and services. Many of the companies offer only a few selected items online, which are suitable to be sold using the online channels. Other products may not have the features to be promoted and sold online. Also, there are many costs associated with the Digital Marketing; these include costs for setting up the websites, maintaining and updating the websites and managing the online channels. Many of the e-marketers are still struggling to make their business visible, viable and profitable using the Digital Marketing techniques.

- **Security:** Providing a secure website for web visitors should be the number one priority for online business owners, but not every business owner feels it necessary to provide maximum security for their web visitors. There are thousands of internet scammers and hackers consistently searching for gaps within the website of an internet business. Online business owners should respect their customers by providing security and making sure to add a badge or statement to ward off potential threats. This security concern further results into misuse of the personal information of the customers and thus they hesitate while making online transactions with the company.
- **Updates:** The Internet is a constant provider of fresh, up-to-the-second information. Yesterday's news is of no use and importance to the web surfer, so it is imperative to always keep the website updated with the latest information. It is one of the common mistakes of budding entrepreneurs, they get the website up and running and then simply leave it that way because they did not include the time needed to maintain the website into their marketing plan, and now the website is neglected and customers can easily notice these things.
- **Costs:** Businesses of all kinds should have a marketing strategy to map out how much money is needed for marketing and where the money will go. Many business owners who are just starting business are not aware of these hidden costs associated with online marketing. Sending a quick email to someone may not cost much more than internet access, but the costs of hardware resources, launching a website and running online marketing campaign make the online marketing an expensive affair.
- **Engagement:** Digital Marketing requires constant tweaking to keep the subscribers engaged. A person will often sign up to receive information as a part of the promotion. For example, if a user provides his email address, he may receive a discount on a product. A user might also unintentionally sign up, such as if subscribing was the default choice on a web page. In such cases, marketing may be perceived as a nuisance. Even in the case that someone deliberately signed up to receive marketing emails and promotion material, like an industry newsletter, marketers must still work to maintain a level of engagement to keep the reader from unsubscribing.

- **Appearance:** Marketers may be working for making the digital promotions to look exactly the way they have envisioned it. Unfortunately, due to the varying servers and computer settings, the design may not come through the way marketers hoped, which can diminish the impact of promotion message. Text may be moved from one line to the next on a person's screen. The email setting on another person's computer may only allow text, blocking images from coming through. If the main focus of online marketing was a logo, it could be lost during the process its open on the screen of the users.
- **Spam:** "Spam" is unsolicited email, if marketers want to email their customer base, they need to receive permission from them. Even after receiving permission, some people still consider any commercial email spam. This adds to difficulty for the online marketing and puts a question mark on Digital Marketing techniques.
- **Deliverability Issues:** The communication and promotion material may not reach its intended audience. For one thing, certain terms, such as "free" or "money" could trigger the spam filter, where the recipient may never see it. Other times, the headline may not be enticing enough for the customer to open the web page and check the site.
- **Undelivered E-mail:** These days many ISPs use complex junk-mail filters. So, there is no guarantee that E-mails are getting into audience inbox. Also, it is possible that individuals delete an E-mail from someone who is not in their contact list. This is increasingly becoming a disadvantage of e-mail marketing.
- **E-mail response decay:** It's not easy to keep subscribers actively engaged with company in the long-term through the online media.
- **Renderability:** Difficulty of displaying the creative as intended within the inbox of different E-mail reading systems. The subscribers may want a message with "unsecured" items such as color, graphics and links that not all browsers will support them. Finally, the recipient will instantly close the window or marketers will have to just settle for the drab all-text E-mail.
- **Privacy Issues:** When an E-mail gets through to the consumer, there is so much e-mail that needs to be looked at sometimes it is difficult for the individual to distinguish between solicited and unsolicited e-mail, as well as have time to read through the E-mail. Another disadvantage of E-mail marketing is to transmit many software viruses and it makes customers to become suspicious even to the trusted sources.

- **Dependability on technology:** Slow internet connections can cause difficulties. If the companies build too complex or too large websites, it will take too long for users to check them or download them and they will get bored eventually.
- **No Physical Contacts:** The Digital Marketing doesn't allow the user "to touch" the product before purchasing it. Because of this, the customers may not be able to become sure about the real look of the products. They have to just rely on what the companies and websites talk about the products. This may result into customer complaints and poor marketer-customer relationship.
- **Lack of Trust:** One of the major disadvantages may be the lack of trust of the users because of the constant virtual promotions that appear to be frauds. This is an aspect that deteriorates the image and reputation of honest companies.
- **Payment Terms:** The cash on delivery payment option may be another disadvantage of the online marketing systems. COD does not guarantee the 100% purchase of the product. This is also the case of thousands of users who dedicate themselves to daily mock big companies by ordering on the internet using fake identities.
- **Worldwide competition through globalisation:** Due to internet and global networks there has been great transparency about the pricing of all competitors resulting into cut throat competition among the players who are present in the online marketing and promotions. This also results into worldwide and global competitions.

CHECK YOUR PROGRESS

VI. In the year, ……….. clickable banner or web banner came into existence.

a. 1991　　c. 1989

b. 1990　　d. 1993

VII. The part of internet that supports a graphic user interface for hypertext navigation with a browser such as Netscape or Internet Explorer.

a. Web　　c. Extranet

b. Intranet　　d. None of these

VIII. An intranet is like a mini-internet but only for the internal company use. (True/False)

IX. The business organisations make online transactions with other business, in ………. market.

a. B2C　　c. C2C

b. B2B　　d. C2B

X. This involves ………………. marketing, where consumers directly sell products or services to other consumers, using the Internet.

XI. The ………….. of Digital Marketing campaigns means that substantial incremental sales can be achieved online without affecting any other marketing efforts.

XII. Digital Marketing does not make it possible to cover the global markets by eliminating the geographical and territorial boundaries. (True/False)

XIII. The ………………… is a concept that means the exponentially growing amount of data will be categorised in personal and human-usable ways, enabling even better, more meaningful and more accessible content than has been ever possible.

1.11 REASONS FOR THE GROWTH OF DIGITAL MARKETING

Earlier days, when human first started trading whatever it is, they traded and marketing that time was all about the stories that used to convince the customer to make trade. The method of trades are changed or improved. They are more efficient to convey the message and faster to communicate, spared the message to mass or niche, but the old principles are always applied. Now business organisations have started using Digital Marketing for the same due to:

- **Low Investment:** When Digital Marketing is implemented properly, the investments required are less and the return on investment is faster than traditional marketing.
- **Reach:** We are living in the era of global product and global brands. Through the WWW (World Wide Web), product would be in the reach of billions of customers in the world. Every day or every month there is increase in the customers by millions.
- **Scope:** In Digital Marketing scope of business is not limited or restricted. The marketers can provide other services also like information management, public relations, customer service and sales.
- **Interactivity:** The internet has made it possible to interact with the customers easily at any time and also responding to the customers instantly. The marketers now can interact better than traditional marketing.
- **Immediacy:** Digital Marketing makes the immediate impact on customer’s mind, for example if we publish an advertisement in a newspaper of an apple’s I-phone. Reader or customer is very eager to purchase the product but he is not interested to go to the show room and buy it because it is away. So customer might change his idea of

purchasing the product. But through Digital Marketing, customer can purchase the product immediately.

The following facts simplifies the reasons for the growth of Digital Marketing.

Top three marketing priorities for the business are:

- ✓ Driving increased conversion rates (47%)
- ✓ Increasing and improving brand awareness (46%)
- ✓ Collecting, measuring, and using behavior (29%)

Top three metrics used to track the success of Digital Marketing:

- ✓ Conversion Rates (67%)
- ✓ Engagement Rates (opens, clicks) (64%)
- ✓ Return on Investment (61%)

Four key areas where marketers plan to increase spending by 2023:

- ✓ Data & analytics (61% plan to increase)
- ✓ Marketing automation (60% plan to increase)
- ✓ Email marketing (58% plan to increase)
- ✓ Social media marketing (57% plan to increase

According to 2019 State of Marketing report, 98% of marketers plan to increase or maintain their future marketing spend so if your brand's marketing budget expanded considerably compared with last year, you are not alone. Data and analytics, marketing automation, email, social media, and content management ranked as the top five areas for increased spending and marketing budget, so expect to see more businesses tailor their content one-to-one as data investments grow alongside speedy, automated lifecycles.

CHECK YOUR PROGRESS

XIV. When Digital Marketing is implemented properly, the investments required are less and the return on investment is faster than traditional marketing. (True/False)

XV. Through the WWW (World Wide Web), product would not be in the reach of billions of customers in the world. (True/False)

XVI. Through Digital Marketing, customer can purchase the product immediately. (True/False)

1.12 LET US SOME UP

Today, businesses operate in a dynamic environment, in which various kinds of competitions determine the rule of the game. Managing a competitive edge in today's business world demands an effective electronic strategy, concentrating on the convenience of electronic buying. Internet Marketing has converted traditional marketing into digital marketing. These days the world is fully competitive so every business wants to go on for long time. The rapid adoption of the Internet as a commercial medium has caused firms/marketers to experiment with the innovative ways of marketing thus changing the pattern of marketing strategies. Marketers need to understand the technology, and how they can use the technology in the best possible way for the promotion and profitability of business. Digital Marketing is spreading awareness of products and services with the help of digitalized technologies, specifically online. Growth of digital marketing since 1990's and 2000's has revolutionized the way the companies and brands use technology for advertising. Digital Marketing embraces many of the basic principles of traditional marketing with a few differentiating factors. One key advantage to Digital Marketing is that results can be quantified, allowing the marketers to work more effectively, generate higher ROI's and in turn increasing profitability. A well implemented Digital Marketing strategy can achieve a more cost-effective customer acquisition than traditional marketing. Digital Marketing may not be useful for all varieties of products and services. Many of the e-marketers are still struggling to make their business visible, viable and profitable using the Digital Marketing techniques.

KEYWORDS

MARKETING Marketing is a simple process of creating awareness of products or services through various channels towards desired market.

DIGITAL MARKETING Digital Marketing is Achieving marketing objectives through applying digital technologies

CLICKABLE BANNER A type of advertising done with the help of ad server on www

INTRANET A network that runs internally in a company but uses the internet standards such as HTML and browsers

EXTRANET An intranet to which value chain partners have admissions for strategic reasons.

SECURITY The security of consumer information on the internet.

SPAM The unsolicited email sends by marketers with customers' consent.

CONVERSION RATES To convert the potential customers in to actual customers.

ANSWER FOR CHECK YOUR PROGRESS

I. True
II. False
III. Traditional
IV. 1990
V. Prospects and customers
VI. 1993
VII. Web
VIII. True
IX. B2B
X. Consumer-to-Consumer
XI. Reach
XII. False
XIII. Semantic web
XIV. True
XV. False
XVI. True

TERMINAL QUESTIONS

- What is Digital Marketing? How it will contribute to the growth of company and nation?
- Describe the concept of digital marketing and how it is different from online marketing?
- "Business organisations cannot have an effective digital marketing without following the principles of digital marketing" comments on this statement.
- Discuss the origin of digital marketing. What is the influence of internet for digital marketing?
- Explain the different types of digital marketing and highlight what are the barriers for the digital marketing?
- How would you define the digital marketing? What are the advantages and disadvantages of digital marketing?
- Highlight the factors contributing for the growth of digital marketing.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-2 DIGITAL MARKET EVOLUTION, CAREER IN DIGITAL MARKETING

STRUCTURE

2.0 OBJECTIVES

After studying the Unit, you would be able to

- Have through knowledge how digital marketing came into existence
- Evaluate the SWOT analysis of digital marketing
- Explore the future trend of the digital marketing
- Make prediction for the digital marketing
- Understand how traditional marketing is different from digital marketing
- Explore the Careers scope in Digital marketing

2.1 INTRODUCTION

Even before the Internet there were many different ways to advertise, in different media such as radio, TV, newspapers, magazines, as well as via telemarketing or pamphlets. Usually, the goal was to get a company or product name and to communicate as many people as possible for the smallest price possible. When the Internet arose, a number of search options became available. Companies had the option to advertise themselves on a larger scale. Due to the advertising perception at the time, many businesses were assumed to have great value, and thus traded on the stock exchange at extraordinary high rates. This collapsed in 2001 at what is commonly known as the dot-com bubble.

While the timeline of Digital Marketing has been short, the cumulative events leading up to where we are now having impacted the entire globe faster than any marketing revolution in history. In 1994, spending for Digital Marketing totaled nearly nothing, but increased to over Rs.300 million in 1995. According to Forrester Research now, little more than a decade later, marketing spending and Digital Marketing business has exploded to nearly Rs.200 billion. Today, it’s hard to believe in having an organization which doesn’t have some kind of online presence.

2.2 THE MARKETING REVOLUTION

When the internet was first introduced in the early 1990s, it was not considered to be an advertising medium at all. Instead, the internet was treated as a tool for exchanging emails and digital information, but was not yet considered valuable for reaching customers. However, it was not long before marketing pioneers began to see the potential for Digital Marketing business as millions of web surfers logging on each day to find valuable and relevant information. Within just a few years, informative and educational marketing, as well as graphically enticing banner ads began to be shown up. It was not long before the results began to flood in, which proved the value of the internet marketplace to even the most skeptical advertisers.

Most importantly, companies which had been spending huge chunks of their marketing budget on offline list building, begin to realize that they could accomplish the same thing via email and for much less. It was not long before everyone from industry giants such as Microsoft Corp. to small businesses began to build company sites and spend marketing dollars to attract qualified traffic. Next, search engine companies like Yahoo, began to create significant profits from advertising alone.

Then came the great Digital Marketing business bust around the year of 2000, which marked the beginning of the end for interruptive marketing such as flashing banner ads. What was happening? As interactive features were added to web pages, consumers were given the option of turning off marketing messages at will and they did. Then entered the age of education based invitational marketing, which crystallized with the creation of web 2.0 technologies. Suddenly, billions of “voices” began to rise all over the world, as the internet marketplace became as much a global community as it was an advertising medium. This led to a relational based marketing approach which has led to one of the most lucrative opportunities for solo entrepreneurs and small startups alike to make a small fortune working from their spare bedroom.

2.3 DIGITAL MARKETING ISSUES, CHALLENGES AND OPPORTUNITIES

There are number of challenges as well as opportunities before the Digital Marketing which are discussed as follows: -

- **Internet Usage:** If computer usage itself is insignificant, Internet usage is almost non-existent. Digital Marketing happens over the Internet and with low Internet usage. The government’s role also seems limited in the promotion of Digital Marketing infrastructure. But maybe the country internet usage will speed up due to the fact that industries, services i.e. banking, investment, electronics, entertainment etc are being net based. Also, the introduction of wireless application protocol and generalized radio services internet and other forms of electronic exchange will come onto the scene.
- **Bandwidth:** The lack of bandwidth is a major cause for concern. India has very low bandwidth speed as compared to other countries i.e., USA, UK and China etc. But there have been have several applications with the DoT by the large and small ISPs to set up their own gateways, which would definitely result the introduction of newer and faster technology.
- **Cyber Laws:** Poor cyber laws may be considered a big issue in the way of Digital Marketing promotion in the country. Sometimes the consumers face problems with their electronic purchases due to of non-delivery of orders placed online. And it takes only one scam to shake the confidence from the system. Legal shelter is extremely imperative for confidence to build up. Then there is the issue of insurance against fraud. The lack of a secure online payment mechanism and the continued debate on passing the Digital Signature Act are all major stumbling blocks.

Business-to-business sites are comparatively better off, as they normally have negotiated-contracts, with governing laws and jurisdiction set out clearly. Also, the transactions are normally between known entities that have, or are actively looking for long business relationships.

- **Digital Identity:** How can customers trust the site they are doing business with? Similarly, how does the site ensure that they are not using a stolen credit-card number? The answer to this is digital certificates. But, to issue a digital certificate, we need an extremely trustworthy agency issuing the certificate of identity.
- **Inter-State Goods Movement Regulations:** Anyone involved in selling and shipping goods across the state borders will testify what a nightmare this can be. No two states have the same sales tax and excise-duty rules. The rates differ, the forms to be filled in differ and then there’s octroi.
- **Setup and Infrastructure Costs:** In spite of what many vendors would want us to believe, Digital Marketing is not a one-man operation that can run from a couple of servers stashed away in a corner in a basement. Equally big is the cost of software, Web-hosting services, bandwidth, and proper security. All these could add up to quite a big amount, both in initial investments and in running costs.
- **Advertising Costs:** The Internet is often likened to a high-speed digital highway with millions passing by every minute. No way, It’s more like a mess of alleyways that no one has ever mapped out. Having a cool Website will not bring in people. People come to a Website either by typing the URL directly in their browser, or by clicking on an interesting link seen on some other site. Either of these costs a lot of money.

2.4 FUTURE TRENDS OF DIGITAL MARKETING

The Internet has actually modified the method in which details are shared, and has actually had an extensive influence on marketing. Over the previous couple of years, there has actually been even more of a change towards incoming strategies, while lots of outgoing methods have actually become old. More companies are discovering success publishing original content instead of embedding ads within external content, because of the extra perks these strategies provide, such as branding and audience growth. Today, over 75% of website traffic comes through organic search results and 85% of people trust search engines for fulfilling their primary business needs. Let us find out what will be the future Digital Marketing trends.

- **Search engine optimization**: It is an essential strategy in digital marketing. It is the process in which the website's visibility is affected or making a web page visible in a search engine's unpaid results, it is often referred to as "natural results". This process is done by increasing the volume of keywords that a site ranks for.
  Search engine optimization targets the various kinds of search such as video search, image search etc. it also covers the factors like how the search engine works, about what the people search for and the kind of keywords typed into search engines. The kind of search engines are preferred by the company's targeted customers also finds out by the search engine optimization.
- **Pinterest:** Now Pinterest being the major platform to advertise the products in socially and publicly. Pinterest is a web and mobile application. It operates a software systems designed to discover information on the World wide web, mainly using images and on the smaller scale, GIFs and videos. It is improving day by day and providing better result to subscribers.
- **Adware:** It is software that once installed it automatically displays advertisements on the user's machine. All the ads may appear in the software itself, and integrated into the web pages visited by the user, or in pop-ups.
- **Social Conversion:** Social networking have already play an important role in Digital Marketing, expect their importance to increase. Social conversions are important for e-commerce and lead generation sites.
- **Automation:** With the passage of time marketing automation has been improve. Now most of the technical products have innovated like washing machines, fridges, coolers etc. by getting innovative ideas technical products become automatically.
- **Search engine marketing**: It is a type of internet marketing uses the search engines for advertising the company's website or business to customers using internet. The more traffic to company's website is generated via advertising or paid links. It includes all the paid listings, search engine optimization and other related services that will increase traffic to the company's web site.
- **Exploring New Paying Methods:** Generally speaking that individual is looking for secure, simple and easy way for payment. Now many online payment methods are

available in the market. By improving the payment methods international marketing will improve and become successful.

- **Email marketing:** It is one of the earliest ways of digital marketing. Email marketing is composed of sending the commercial message regarding the company's product to a group of targeted customers using email and delivering the personalized emails and the targeted messages at the right time. With the help of email marketing software the company can maintain separate email lists on the bases of customer's likes and dislikes.
- **Social media marketing**: It is a form of marketing that utilizes the networking sites as a tool of marketing. The main goal of social media marketing is to produce that kind of content that the customers share with their social network groups. Many customers try to finds their favourite brands on social media so proper social media strategy is now become necessary for every organization to reach to their targeted audience. With the increased popularity of Google, face book, twitter and YouTube social media marketing became more popular. Posting the right kind of content is necessary so that the people will share company's content, it helps to reach to the new customers. These kinds of sites also are used as an effective form of communication to engage with customers.
- **Mobile advertising:** It is also one of the biggest growth areas in the digital marketing. The advertising is done via mobile phones. The mobile phones which are used for advertising are: smart phones, wireless phones or tablet computers.
- **Online advertising:** It is also known as internet advertising or web advertising, is a form of marketing and advertising that uses the internet services to deliver the promotional marketing messages to the consumers. Despite of many benefits of online marketing there are some reasons the consumers finds this mode disruptive and blocking the ads. Online advertising includes: Social media marketing, search engine marketing and Mobile marketing.
- **Affiliate marketing:** Affiliate marketing occurs when the business firm organize third parties for bring in customers for them. The payment is given to the third parties on the bases of sales generated through their advertising. Affiliate generates the traffic to the company through their affiliate networks; once the purchase is made by the visitors they receive the payment. The desired actions can be: Email submission, Phone call, Filling out an online registration form.

- **Display advertising:** It is done through advertising on websites. The main purpose behind display advertising is to deliver brand messages to site visitors and deliver general advertisements. Display advertising includes different formats and contains many items such as: Flash, Images, Text, Video, Audio.

CHECK YOUR PROGRESS

I. In 1994, spending for Digital Marketing totalled nearly nothing, but increased to over Rs.300 million in 1995. (True/False)

II. When the internet was first introduced in the early ………s, it was not considered to be an advertising medium at all.

a) 1995 c) 1985

b) 1990 d) 1981

III. ………… has very low bandwidth speed as compared to other countries i.e., USA, UK and China etc.

IV. Poor cyber laws is not considered a big issue in the way of Digital Marketing promotion in the country. (True/False)

V. ………………. of website traffic comes through organic search results.

a) 50% c) 75%

b) 90% d) 80%

VI. The motive behind paid search advertising is to get instant traffic to the ………..

VII. Google buzz which is an addition to social features in Gmail is creating buzz everywhere. (True/False)

VIII. Social conversions are not important for e-commerce and lead generation sites. (True/False)

2.5 PREDICTIONS REGARDING MARKETING

Digital Marketing has changed time to time with the passage of time. Digital Marketing start in 1993 with the simplest form of online marketing. Today, 25 years later, online marketing has changed in many significant ways, involve search engines, social media platforms, Gmail cover all other electronic post platforms, You-tube etc. Some predictions are measured by specialists, they are given below.

- **Reality optimization will become a thing:** The concept of search optimization is all about getting found, but there can be a new, physical application of this inbound discovery efforts once augmented reality device become popular. In future the reality optimization will become only a thing.
- **Unbound ads will die:** Now internet is almost is everywhere, with no concept of connection, traditional advertising is going to finally die. In future we predict that only digital ads will available like television ads.
- **Competition will be reduced:** By increasing the technology and machines there will be less work available for the humans. With less work, there will be fewer businesses. All this results in a complicated evolution of economy system. We will have more distance to resources we have every time desire, but there will be some brands and businesses providing them. So the competition will reduce.
- **Strange new social interactions will be introduced:** Today are column collections of content; each person and company has an indifferent profile that submits and advertise content. In future digital profiles may be more closely bind to a public and individual can be able to produce content and exchange with others more immediate.
- **More Integrated Global:** Now inbound and outbound marketing are combined. We all know that after the innovation of digital or Digital Marketing the internet services is increase immediately and with this innovation the global marketing will also increase. In future more new technology will innovate and the global relation and businesses are there.

2.6 TRADITIONAL MARKETING V/S DIGITAL MARKETING

Traditional marketing is a broad concept that encompasses several different categories of advertising and marketing. Traditional marketing, rather than the rise of digital technology in online marketing, is critical for the growth of any company. Digital marketing refers to the promotion of goods or services through the use of digital media, mostly the internet, but mobile phones, display advertisements, and other digital terminology is also included.

The key distinction between traditional and digital marketing is how the marketing message is communicated to an audience.

Area	Traditional Marketing	Digital Marketing
Target Audience	Traditional marketing strategies make it simple to connect with local audiences.	You can meet your target audience from all over the world using digital marketing.
Marketing Approach	Traditional marketing takes a more intimate approach so it is very straightforward for marketers to have a one-on-one interaction with the public while educating them or advertising their brand's name.	The marketers' tangible presence is not needed in digital marketing; it is a bonus, but it is not required, because digital marketing enables the marketer to reach a limited number of customers quickly.
Documentation	Traditional marketing strategies include a physical copy of the products/services definition that can be read several times.	Digital channels offer descriptions in softcopy via the website, or via images, you tube, and other media, which can be accessed at any time and from any location to fulfil the wants of the user, and they eliminate the need for paper documentation.
Consumer Interaction	Traditional marketing involves very little interaction because the medium of publicity are not versatile enough to incorporate consumer interaction.	Digital marketing provides consumers with a variety of digital ways like- social networking sites, e-commerce websites, and various apps for gathering reviews where they can express their opinions about the products/services.
Marketing Cost	Traditional marketing is costly because it involves printing and radio/TV commercials, all of which add to the company's costs.	Since everything is online and social media platforms is free, marketing is less costlier than traditional marketing. If a company's marketing needs necessitate it, they may choose to use paid advertisements.
Marketing Analysis	Traditional marketing makes result analysis more difficult because companies must rely on surveys and market findings.	In digital marketing, data and facts are collected and stored digitally, making it easier for marketers to evaluate marketing results.
Real-Time Results	Traditional marketing strategies require the organisation to wait weeks or months for a response.	Digital marketing produces instant results, making it simple to obtain real-time marketing results.
Interruptions	Advertisements and other items that causes interruptions are not something that consumers may opt to skip.	Consumers can miss the interruptive element of digital marketing strategies and continue to engage with the products/services.

Strategy Refinement	Since traditional marketing strategies do not produce real-time results, developing a marketing strategy that is based on marketing results takes time.	Outcomes of the real-time results provided by digital marketing, strategy refining becomes much easier, and the marketing team may choose to modify or adjust their business strategy in response to market results.
Communication	Traditional marketing relies on one-way contact due to the rigidity of marketing mediums.	Online marketing encourages two-way contact, which helps to satisfy customers and give them the feeling that they are being heard and served.

The secret to a successful marketing strategy is understanding the relationship between digital and traditional marketing methods. Will they be able to collaborate? Yes, absolutely. Digital marketing will be a valuable addition to traditional marketing. Both are essential parts of a marketing campaign. However, combining the two will yield better results for business.

2.7 THE SCOPE OF CAREERS IN DIGITAL MARKETING

As we know that huge growth has taken place in the area of online business and it has been growing day and night. We people have huge number of career opportunities in the Digital marketing.

- We can start our career as Professional Blogger
- We can start the Affiliate Marketing & AdSense
- There are opportunities for Freelancing Services
- Group of persons have expertise can start their own organisation
- A person can become a Youtuber
- A Person can start Drop transport business
- Person can start training and coaching classes for untrained persons in online business

Having expertise in the digital business person can occupy the following role in Digital Marketing:

- Digital Marketing Manager or Digital Director
- Web Developer & Web Designer
- Analytics Manager
- CRM Manager

- Email Marketing Manager
- E-Commerce Manager
- Digital Agency Account Director
- Social Media Executive and Social Media Manager
- Search engine optimization Executive/Expert
- PPC/SEM Expert
- Content Developer

CHECK YOUR PROGRESS

IX. In future we predict that only digital ads will available like television ads. (True/False)

X. In future digital profiles may be more closely bind to a public and individual can be able to produce content and exchange with others more immediate. (True/False)

XI. …………. is a broad concept that encompasses several different categories of advertising and marketing.

a) Traditional marketing
c) Social Media Marketing
b) Niche Marketing
d) E-Mail Marketing

XII. Digital marketing refers to the promotion of goods or services through the use of digital media, mostly the………………...

XIII. Traditional marketing involves very huge interaction because the medium of publicity is not versatile enough to incorporate consumer interaction. (True/False)

XIV. Digital marketing, produces instant results, making it simple to obtain ……… marketing results.

XV. Traditional marketing relies on ……………. contact due to the rigidity of marketing mediums.

a) one-way
c) Both
b) Two-way
d) None of these

2.8 LET US SUM UP

Digital Marketing business bust around the year of 2000, which marked the beginning of the end for interruptive marketing such as flashing banner ads. When the internet was first introduced in the early 1990s, it was not considered to be an advertising medium at all.

Instead, the internet was treated as a tool for exchanging emails and digital information, but wasn't yet considered valuable for reaching customers. However, it was not long before marketing pioneers began to see the potential for Digital Marketing business as millions of web surfers logging on each day to find valuable and relevant information. The Internet has actually modified the method in which details are shared, and has actually had an extensive influence on marketing. Over the previous couple of years, there has actually been even more of a change towards incoming strategies, while lots of outgoing methods have actually become old. There are number of challenges as well as opportunities before the Digital Marketing like Internet usage is almost non-existent, lack of bandwidth is a major cause for concern and many more. Even after problems, online marketing has changed in many significant ways, involve search engines, social media platforms, Gmail cover all other electronic post platforms, You-tube etc. Traditional marketing is a broad concept that encompasses several different categories of advertising and marketing. Traditional marketing, rather than the rise of digital technology in online marketing, is critical for the growth of any company.

2.9 KEY WORDS

INTERRUPTIVE MARKETING	Innovative marketing such as flashing banner ads.
CYBER LAW	Law to protect the interest of online users
DIGITAL IDENTITY	Identification of company and user in the online market.
SEARCH ENGINE OPTIMIZATION	Technique to get high ranking for the company web site.
MOBILE MARKETING	Set of practices that enables organizations to communicate an interactive manner through any mobile device
BLOGGING	Blogging is a way to update website periodically with fresh and unique content.

2.10 ANSWERS TO CHECK THE PROGRESS

I. True

II. 1990

III. India

IV. False

V. 75%

VI. Website

VII. True

VIII. False

IX. True

X. True

XI. Traditional Marketing

XII. Internet

XIII. False

XIV. Real-time

XV. One-Way

2.11 TERMINAL QUESTIONS

- Describe in details how the digital marketing came into existence. What were the stages through which it has crossed to become a promotional tool?
- How digital marketing is useful for business organisations? What are the challenges which marketing managers facing in Indian Environment?
- What are digital marketing issues faced by the marketing managers and how would you predict the future trend for digital marketing under same scenario?
- Digital marketing has grown in India through difficult stages, how would you see the future of digital marketing in India?
- Digital marketing has grown from the traditional marketing but id has been doing great for the promotion of business organisation, how would you see which one is better?
- Explore the prediction for the digital marketing. How traditional marketing is different from the digital marketing?
- What is digital marketing evolution? What are the career opportunities in the digital marketing?

UNIT-3 DIGITAL CONSUMER: CONSUMER CHARACTERISTICS AND PROFILE

STRUCTURE

3.0 OBJECTIVES

After studying the Unit, you would be able to

- Basics understanding for the consumer in online market
- Understand the customer centric online marketing
- Explore the adoption process
- Evaluate the features of consumers in online market
- Develop the strategies for segmentation, targeting and positioning
- Analysis the SWOT of E-Marketing
- Describe the Public relationship process

3.1 INTRODCTION

Digital Marketing is a revolution in today's business world. Business organizations have been forced to adopt technological change over the last decade. Digital Marketing utilizes electronic channels to carry through their marketing activities to attain marketing objectives of the organization. In a broad sense, Digital Marketing is the utility of computer and internet technology, or electronic based activities, to improve marketing activities and performance. It improves the online execution of delivering customer benefits and satisfaction. Digital Marketing draws heavily on Internet communications to co-ordinate many marketing activities, such as market research, New product development, market segmentation, positioning, product distribution, customer services, promotion, customer feedback, etc. These internet or electronic based activities should, in turn, be integrated with the overall marketing strategy to support the corporate objectives of the E-commerce.

Digital Marketing is one of the ways through which marketers can develop the link and have good relation with the customer. As business is becoming more and more difficult or rigid, so there is need to make a good relation with the customer because a customer is the backbone of every business. Therefore, with the help of a customer centric online marketing marketer can make a customer happy, as survival of business depends on the happiness and satisfaction of the consumers.

3.2 CUSTOMER CENTRIC ONLINE MARKETING

Digital Marketing has become a vital component of the overall marketing efforts. In a competitive business environment, losing customers is very easy as the competitor is just a click away. Therefore, Implementing Digital Marketing strategy, requires an adequate understanding of changes in the way customers think and act on the World Wide Web. In other words, in a dynamic business environment, Digital Marketing is capable of cutting through the clutter to provide a precious opportunity for businesses to improve accessibility to the world and thrive.

Hence in case marketers want to acquire and to retain more customers, then they have to give more importance to the consumers than their competitors. With the help of a customer centre in online marketing strategy, marketers would be able to attract and satisfy more customers in the E-market; otherwise, E-marketers would not be able to survive in the market. Marketers need to deliver product and services as per the requirement of customers. Companies that are customer centric are 60% more profitable than companies that are not customer centric.

There are Customer-centric best practices marketers need to consider while developing marketing strategies.

- **Believe that customers come first:** In every business they have to believe that the customer will come first. No business can survive in the market without the customers. Customers are the heart and the backbone of the business.

- **Customer Satisfaction: C**ustomer satisfaction is one of the important factors, customer must be satisfied as far as the product or the services are being rendered by the business.
- **Handling time:** in case customers have any problem regarding product, they need to be solved as soon as possible, as this would create trust among customers.
- **Customer Needs:** There should be a proper market research in order to find the utility and value, customer seeking from the product. In case E-marketer is able to match with the expectations of customer, there would be more market share for the product.
- **Customer Relationship:** Marketers need to have good relationship with customers through online marketing channels. When they have a good relationship with customers, they would never sift to another brand. According to Sethgodin, "it is easier to love a brand , when the brands love you back".

3.3 THE ADOPTION PROCESS

The Adoption Process (also known as the Diffusion of Innovation) is more than forty years old. It was first described by Bourne (1959), so it has stood the test of time and remained an important marketing tool ever since. It describes the behavior of consumers as they purchase new products and services. The individual categories of innovator, early adopter, early majority, late majority and laggards are described below:

- **Innovators** are the first one to adopt and display behavior that demonstrates that they likely want to be ahead, and to be the first to own new products, well before the average consumer. They are often not taken seriously by their peers. They often buy products that do not make it through the early stages of the Product Life Cycle (PLC).
- **Early adopters** are also quick-to-buy new products and services, and so are key opinion leaders with their neighbors and friends as they tend to be amongst the first to get hold of items or services.

- **The early majority** looks to the innovators and early majority to see if a new product or idea works and begins to stand the test of time. They stand back and watch the experiences of others. Then there is a surge of mass purchases.
- **The late majority** tends to purchase the product later than the average person. They are slower to catch on to the popularity of new products, services, ideas, or solutions. There is still mass consumption, but it begins to end.
- **Laggards** tend to be very late to take on board new products and include those that never actually adopt at all. Here there is little to be made from these consumers.

There are a number of examples of products that have gone through the adoption process. They include I-pods or DVD players (or even video players and smart watches). Initially only a small group of younger or informed, well off people bought into these products. Opinion leaders or the early adopters then buy the product and tend to be a target for marketing companies wishing to gain an early foot hold. The early majority is slightly ahead of the average, and follow. Then the late majority buys into the product, followed by any laggards. New adoption process or curves begin all the time. Who knows what will happen with solid state technology or Internet purchases of media?

3.4 WHY CUSTOMERS BUY

Marketers spend millions of rupees trying to understand why customer buy products and services. Sometimes it seems that there is no reason for a purchase, but in reality, there is always a reason. Many factors are involved in a customers' buying decision, any one of which can become the deciding factor, such as:

- Practical-consumers purchase products because they need them to survive, such as food, clothes and medicine.
- Impractical-consumers are the opposite of practical, purchasing products that may be necessary.
- Rational-Purchases are made with logical.
- Irrational-products are purchased for foolish or absurd reasons.
- Factual-Purchasing products based on researched reports.
- Emotional-purchasing is based on feelings.

3.4.1 Consumers Also Buy:

- **To Increase:** Sales, Profit, Satisfaction, Confidence, Convenience, Pleasure, and Production.
- **To Protect:** Investment, Self, Employees, Property, Money, and Family
- **To Make:** Money, Satisfied customers, and Good impressions.
- **To Improve:** Customer relations, Employee relations, Brand image, Status, Earnings, and Performance.
- **To Reduce:** Risk, Investment, Expenses, Competition, Worry, and Trouble
- **To Save:** Time, Money, Energy, and Space.

CHECK YOUR PROGRESS

I. Digital Marketing is the utility of computer and internet technology, or electronic based activities, to improve marketing activities and performance. (True/False)

II. In a competitive business environment, losing customers is very difficult as the competitor is just a click away. (True/False)

III. Companies that are customer centric are ……. more profitable than companies that are not customer centric.

a) 70% c) 60%
b) 50% d) 80%

IV. The Adoption Process is also known as the …………………..

V. …………. tend to be very late to take on board new products and include those that never actually adopt at all.

a) Laggards c) Early adopters
b) Innovators d) The early majority

VI. Marketers spend millions of rupees trying to understand why ……… buy products and services.

3.5 CHARACTERISTICS OF THE CONSUMERS

- **Intelligence:** Unless companies are specifically marketing a product to extremely intelligent individuals, it is best to word marketing messages on a level most people can understand, and do not ever talk to customers in a way that would make them feel inferior.

- **Involvement:** A customer with higher levels of involvement with the product, service and marketing information will have more recall than a consumer with less involvement. Creating more interest in the product and making a website more interactive will help to increase sales. More involvement results in more sales.
- **Familiarity:** Generally, the more familiar a customer is with a product, the more likely they are to purchase it; however, having too much familiarity can lead to adaptation, when customers become tired of their “familiar” purchases and seek out novelty items.
- **Expectations:** If the customer does not know what to expect from the product or service, then they are not going to purchase it. This explains why familiar brands like Coca-Cola, Dabur, Godrej etc. do not change their logos; customers are familiar with them, have positive thoughts about them and know what to expect.
- **Physical limits:** Marketers need to remember that some consumers have limitations such as hearing impairment or colour blindness and these needs to be taken into consideration when creating marketing messages.

3.6 MARKET SEGMENTATION, TARGETING AND POSITIONING

It is very important to learn what market segmentation, why it is important is and the different dimensions used by marketers to segment the population. How marketers evaluate and select potential market segments is explained as is the development of a targeting strategy. We understand how a firm develops and implements a positioning strategy and creates a customer relationship management strategy to increase long-term success and profits. After careful study, we will learn that it is not at all right to treat people differently but is a requirement in successful marketing. As small children, we are often taught to treat everyone alike. But after reading market segmentation, we will learn that this strategy does not work in marketing. The goal of marketing is to create value and satisfy needs. However, everyone’s needs are not the same and understanding needs is a complex task.

3.7 E-MARKET SEGMENTATION

Market segmentation is an important process in conventional marketing. Segmentation is an important step in ensuring successful implementation of marketing strategies. Market segmentation means dividing a market into different groups or categories according to certain criteria such as household income, level of income, level of education, age, gender, hobbies and more. Such data can be collected through questionnaire survey or from statistical department. Once we have those groups ready, we can then use differentiated marketing strategies to target those groups based on their needs and wants. For example, ask yourself what you can sell to the senior citizens? Your answers could be nutritional products, tour packages, medicines, indoor exercising machines and more.

Now, how do we do market segmentation in the Internet marketplace, or rather marketspace? The concept is basically the same, the only difference is companies have a much larger marketplace, and the main concern is language and cultural differences. For example, if marketers wish to promote inbound tour packages, they can target adult travelers from different countries, using website that offers different major languages in the world, such as English, French, Spanish, German, Chinese and Japanese. If companies cannot do it themselves, they can always outsource to companies that provide translation services. Besides, organizations might want to customize the website according to different cultures, particularly those relating to food and things they like to see.

One of the most popular Internet outsourcing companies is **Odesk.com.** In this website, one can outsource jobs to suitable contractors who are mostly individual freelancers. One can bargain with them the price as well as other terms and conditions for handling their jobs. Jobs that can be outsourced are webpage design, contents writing and editing, graphic design, translation, copywriting, customer support and more.

3.7.1 The Need for Market Segmentation

Segmentation is the process of dividing potential markets or consumers into specific groups. Market research analysis using segmentation is a basic component of any marketing effort. It provides a basis upon which business decision makers maximize profitability by focusing their company's efforts and resources on those market segments most favorable to their goals. If all consumers were alike and had the same background, education and experience, mass marketing or undifferentiated marketing would be a logical strategy. The essence of market segmentation was summed up by Henry Ford.

3.7.2 Requirements of Market Segments

In addition to having different needs, for segments to be practical it should be evaluated against the following criteria:

- Identifiable
- Accessible
- Substantial
- Unique needs
- Durable

3.7.3 Bases for Segmentation

The next step in developing a market segmentation strategy is to select the most appropriate bases on which to segment the market. The marketer will have to try different segmentation bases or segmentation variables, alone or in combination, to find the best way to view the market structures. The major bases to segment consumer markets are the following:

- Geographic segmentation
- Demographic segmentation
- Psychographic segmentation
- Behavioural segmentation

3.8 TARGETING

In targeting, the marketers evaluate the attractiveness of each potential segment and decide which of these groups they will invest resources against to try to turn them into customers. The customer group or groups selected are the firm's target market. A target market consists of a set of buyers who share common needs or characteristics that the company decides to serve. Target marketing can be carried out at different levels:

- **Undifferentiated Marketing:** An undifferentiated targeting strategy is one that appeals to a wide-spectrum of people. If successful, this type of operation can be very efficient, especially because production, research, and promotion costs benefit from economies of scale, it's cheaper to develop one product or one advertising campaign than to choose several targets and create separate products or messages for each. The company must be willing to bet that people have similar needs or differences among them that are trivial.
- **Differentiated Marketing:** A company that chooses a differentiated targeting strategy develops one or more products for each of several customer groups with different product needs. A differentiated strategy is called for when consumers are

choosing among brands that are well known in which each has a distinctive image in the marketplace and in which it's possible to identify one or more segments that have distinct needs for different types of products. E.g.: Tata Motors- a car for every 'purse', purpose and personality, Nike shoes- for running, golf, aerobics, cycling and other sports, Procter and Gamble- different brands for laundry detergent.

- **Concentrated Marketing or Niche Marketing:** When a firm focuses its efforts on offering one or more products to a single segment, it is using a concentrated targeting strategy. A concentrated strategy is often useful for smaller firms that do not have the resources or the desire to be all things to all people. Marketers generally identify niches by dividing a segment into sub-segments. E.g.: sports channels like ESPN, Star Sports, and Religious TV Channels.

  Characteristics of niche marketing

  - The customers in the niche have a distinctive set of needs.
  - The customers are ready to pay a premium price to the company that best satisfies their needs.
  - The niche is not likely to attract other competitors.
  - The niche marketer gains some economies of scale through specialisation.
  - The niche has adequate size, profitability and growth potential.

- **Customized Marketing or Micro Marketing:** Ideally, marketers should be able to define segments so precisely that they can offer products and services that exactly meet the unique needs of each individual or firm. A custom marketing strategy is common in industrial contexts in which a manufacturer often works with one or a few large clients and develops products and services that only these clients will use.

  In most cases this level of segmentation is neither practical nor possible when mass-produced products are sold. However, advances in computer technology, coupled with the new emphasis on building solid relationships with customers, have focused managers' attention on devising a new way to tailor specific products and the messages about them to individual customers.

3.9 MARKET POSITIONING

The concept of positioning was propounded by two advertising executives, Al Ries and Jack Trout. They considered positioning as a creative exercise done by marketing people with both existing as well as new products. According to them positioning starts with a product, a piece of merchandise, a service, a company, an institution or even a person. But positioning is not what is done to a product. Positioning is what marketers do to the mind of the prospect. That is, marketers position the product in the mind of the prospect.

Positioning is developing a product and brand image in the minds of consumers. It can also include improving a customer's perception about the experience they will have if they choose to purchase product or service. The business can positively influence the perceptions of its chosen customer base through strategic promotional activities and by carefully defining the business' marketing mix. Effective positioning involves a good understanding of competing products and the benefits that are sought by the target market. It also requires companies to identify a differential advantage with which it will deliver the required benefits to the market effectively against the competition. Business should aim to define themselves in the eyes of their customers in regards to their competition.

3.9.1 Developing a Positioning Strategy

Positioning means developing a marketing strategy aimed at influencing how a particular market segment perceives a product or service in comparison to the competition. Developing a positioning strategy entails gaining a clear understanding of the criteria that the target consumers use to evaluate competing products and then convincing them that your product will meet those needs. Positioning can be done in many ways. Marketers must devise a marketing mix that will effectively target the segment's members by positioning their products to appeal to that segment. A first step is to analyze the competitors' positions in the marketplace. Who are the direct competitors and what products or services are they providing? Indirect competition can also be important. The following is a list of some established product positioning strategies.

- **Against a Competitor:** Positioning product directly against a competitor's typically requires a specific product superiority claim. A memorable example is Dominos 30 minutes delivery guarantee.
- **Product Categories:** Comparing product to a product in a different category can be an effective way to differentiate. In a soap-compares-itself-to-lotion example,

Palmolive dishwashing liquid claims that it softens your hands while you do the dishes.

- **Away from a Competitor:** Positioning as the opposite of competitor can help marketer to get attention in a market dominated by some other product. A famous example is 7-UP calling itself the Uncola.
- **Benefits:** This strategy focuses on a benefit; product provides to target audience. Examples include Volvo's emphasis on safety and Colgate toothpaste's focus on reducing cavities.
- **Product Attributes:** Highlighting a specific attribute of product can also be compelling. For example, Taj hotels focus on luxury; Ginger hotels focus on economy.
- **Usage Occasions:** This kind of positioning stresses when or how product is used by target audience. Jeep's focus on off-road driving is an excellent example.
- **Users:** Focusing on the unique characteristics of specific users can also be effective. For example, Dummies series of instruction books are attractive to people who want to learn about a topic from a source that doesn't assume any prior knowledge on the reader's part.

3.9.2 Product Positioning and Differentiation

Marketers must also develop a positioning strategy that includes offering a product or service with a competitive advantage, providing a reason why consumers will perceive the product as better than the competition. Once a positioning strategy is set, marketers must finalize the marketing mix by putting all the pieces into place. The elements of the marketing mix must match the selected segment. This means that the goods or services must deliver benefits that the segment values, such as convenience or status. Furthermore, marketers must price this offering at a level these consumers will pay, make the offering available at places consumers are likely to go, and correctly communicate the offering's benefits in locations where consumers are likely to take notice. Finally, marketers must evaluate the target market's responses so they can modify strategies as needed. Over time, the firm may find that it needs to change which segments it targets or even redo a product's position to respond to marketplace changes. An example of such a makeover is called repositioning.

3.10 REPOSITIONING

In volatile markets, it can be necessary, even urgent, to reposition an entire company, rather than just a product line or brand. When Goldman Sachs and Morgan Stanley suddenly shifted from investment to commercial banks, for example, the expectations of investors, employees, clients and regulators all needed to shift, and each company needed to influence how these perceptions changed. Doing so involves repositioning the entire firm. This is especially true of small and medium-sized firms, many of which often lack strong brands for individual product lines. In a prolonged recession, business approaches that were effective during healthy economies often become ineffective and it becomes necessary to change a firm's positioning. Repositioning a company involves more than a marketing challenge. It involves making hard decisions about how a market is shifting and how a firm's competitors will react. Often these decisions must be made without the benefit of sufficient information, simply because the definition of "volatility" is that change becomes difficult or impossible to predict.

3.10.1 Product positioning process

Generally, the product positioning process involves:

- Defining the market in which the product or brand will compete (who the relevant and prospect buyers are)
- Identifying the attributes (also called dimensions) that define the product 'space'
- Collecting information from a sample of customers about their perceptions of each product on the relevant attributes
- Determine each product's share of mind
- Determine each product's current location in the product space
- Determine the target market's preferred combination of attributes (referred to as an ideal vector)
- Examine the fit between the product and the market.

3.10.2 Why is positioning important?

Product positioning is a crucial ingredient in the buying process and should never be left to chance. It is company's opportunity to influence the market's perception of their products and services. Failure to proactively address product positioning is unlikely to end well. With or without marketer input, customers will position the product, probably based on information from the competitors, which will not flatter the organisations. Clear, concise, meaningful product positioning also helps companies cut through the relentless advertising and marketing noise of the marketplace. In customer's mind, product positioning gives the messages some context so they can be better heard and accepted.

CHECK YOUR PROGRESS

VII. More involvement of customer in purchase, results in more ………..

a) Promotion c) Production

b) Sales d) Market share

VIII. If all consumers were alike and had the same background, education and experience, mass marketing or undifferentiated marketing would be a logical strategy. (True/False)

IX. ………….. is an important step in ensuring successful implementation of marketing strategies.

X. A……………… consists of a set of buyers who share common needs or characteristics.

XI. An …………… targeting strategy is one that appeals to a wide-spectrum of people.

a) Differentiated c) Undifferentiated

b) Niche d) All above

XII. It can also include improving a customer's perception about the experience they will have if they choose to purchase product or service. (True/False)

XIII. Positioning means developing a marketing strategy aimed at influencing how a particular market segment perceives a product or service in comparison to the competition. (True/False)

XIV. The …………… of the marketing mix must match the selected segment.

3.11 E-MARKETING SITUATION ANALYSIS

The *situation analysis for E-marketing* bridges the internal audit and competitor research. It answers the question where are we now in terms of our electronic marketing (internal v/s external perspective)? The analysis literally considers **electronic marketing** situation by considering the fit between internal and external factors. There are similarities with traditional concepts and techniques, but marketers need to focus upon digital marketing.

Some of the problems that marketer may encounter with SWOT are as a result of one of its key benefits i.e., its flexibility. Since SWOT analysis can be used in a variety of scenarios, it has to be flexible. To overcome these issues, one should employ a Power SWOT. Smith and Chaffey (2006) distil the situation of a business using Internet as part of its business under the following 5S's:

- **Sell**: Grow sales and attract business using digital technologies.
- **Serve:** Add value through the benefits of the Internet such as speed.
- **Speak:** Get closer to customers by making business available to them at home, work or on the go with mobile technologies.
- **Save:** Reduce costs by using information technologies to make business more efficient.
- **Sizzle:** Extend the online brand (or create a new one) remember sell the sizzle not the sausage i.e. the benefits, aesthetics or value of a product or service rather than its features.

3.12 PUBLIC RELATIONS PROCESS

The definition of Public Relations as relations with the general public through publicity, those functions of a corporation, organisation, branch of military service, etc., concerned with informing the public of its activities, policies, etc., attempting to create favourable public opinions. Public Relations are the planned effort to influence opinion through good character and responsible performance, based upon mutually satisfactory two-way communications.

- **Research-listening:** This involves probing the opinions, attitudes and reactions of those concerned with the acts and policies of an organisation, then evaluating the inflow. This task also requires determining facts regarding the organisation: "what's our problem?"
- **Planning-decision making:** This involves bringing these attitudes, opinions, ideas and reactions to bear on the policies and programmes of the organisation. It will enable the organisation to chart a course in the interests of all concerned: "Here's what we can do."
- **Communication-action:** This involves explaining and dramatizing the chosen course to all those who may be affected and whose support is essential: "Here's what we did and why."

- **Evaluation:** This involves evaluating the results of the programme and the effectiveness of techniques used: "How did we do?"

CHECK YOUR PROGRESS

XV. The *situation analysis for E-marketing* does not bridges the internal audit and competitor research. (True/False)

XVI. Public Relations are the planned effort to influence opinion through good character and responsible performance, based upon mutually satisfactory two-way communications. (True/False)

XVII. There are similarities with traditional concepts and techniques, but marketers need to focus upon digital marketing.

3.13 LET US SUM UP

Digital Marketing is one of the ways through which marketers can develop the link and have good relation with the customer. As business is becoming more and more difficult or rigid, so there is need to make a good relation with the customer because a customer is the backbone of every business. Therefore, with the help of a customer centric online marketing marketer can make a customer happy, as survival of business depends on the happiness and satisfaction of the consumers. The Adoption Process was first described by Bourne (1959), so it has stood the test of time and remained an important marketing tool ever since. It describes the behaviour of consumers as they purchase new products and services. Marketers spend millions of rupees trying to understand why people buy products and services. Sometimes it seems that there is no reason for a purchase, but in reality, there is always a reason. The goal of marketing is to create value and satisfy needs. However, everyone's needs are not the same and understanding needs is a complex task. Market segmentation is an important process in conventional marketing. Segmentation is an important step in ensuring successful implementation of marketing strategies. Market segmentation means dividing a market into different groups or categories according to certain criteria such as household income, level of income, level of education, age, gender, hobbies and more. In targeting, the marketers evaluate the attractiveness of each potential segment and decide which of these groups they will invest resources against to try to turn them into customers. Positioning is developing a product and brand image in the minds of consumers. It can also include improving a customer's perception about the experience they will have if they choose to purchase product or service. The *situation analysis for E-marketing* bridges the internal audit and competitor research. It answers the question where are we now in terms of our electronic marketing.

3.14 KEY WORDS

DIGITAL MARKETING	Electronic marketing strategy to carry through their marketing activities to attain marketing objectives.
CUSTOMER CENTRIC	Marketing strategy focused towards satisfying the customers.
ADOPTION PROCESS	It describes the behaviour of consumers as they purchase new products and services.
RATIONAL BUYING	Purchases are made with logical justifying.
SEGMENTATION	Dividing the Mass market into different groups.
TARGETING	Selecting the appropriate group out of all segmentations.
UNDIFFERENTIATED MARKETING	Where only one product satisfies all the customer in mass market.
NICHE MARKET	Small market, more narrowly defined market.
POSITIONING	Positioning is developing a product and brand image in the minds of consumers.
PUBLIC RELATIONSHIP	Public Relations are the planned effort to influence opinion through good character and responsible performance, based upon mutually satisfactory two-way communications.

3.15 ANSWERS TO CHECK THE PROGRESS

I. True
II. False
III. 60%
IV. Diffusion of Innovation
V. Laggards
VI. Customer
VII. Sales
VIII. True
IX. Segmentation
X. Target market
XI. Undifferentiated
XII. True
XIII. True
XIV. 4Ps
XV. False
XVI. True
XVII. Digital marketing

3.16 TERMINAL QUESTIONS

- Why understanding the consumer perception is important for marketers? How the work is done under customer centric marketing strategy?
- What are the different stages for product adoption process? How the marketing strategies will differ for each stage?
- How the identification of characterises of consumers are important for the marketing manager?
- Highlight the stages of public relationship, how each stage contributes for long term relationship with the customer?
- What is market segmentation, and why is it an important strategy in today’s marketplace?
- Explain the major variables used to segment the consumer markets. Give example of each.
- How do we segment the business markets?
- What is target marketing? What are the different levels of target marketing?
- What is market positioning? How the companies do positioning of their products? Discuss with examples.

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-4 INFORMATION SEARCH BEHAVIOUR

STRUCTURE

4.0 OBJECTIVES

After studying the Unit, you would be able to

- Know how marketers sharing the information with customers
- Understanding the behaviour of customer while searching the information
- Understanding how motive will decide the information search
- Have experience how information requirement change with changing consumer role
- Importance of customer value for developing marketing strategies
- Explore how satisfaction will lead to change in consumer behaviour
- Monitor and measure the customer satisfaction
- Find out the role of internet marketers

4.1 INTRODUCTION

In Today’s World “The customer is the King of the Market”, So every company is making efforts to win the customer. All the process of every company is done by considering the demands, nature, expectations of the customer related to products and services.

One of the many critical tasks of marketer is not only to create the new customers but also to sustain its existing customer base. Customer is the most important factor in the internal business environment because customer is the reason why the business exists. The customer market can be broadly categorized into consumer market and industrial/business market.

Consumer market comprises of the individuals and households who purchase the product and services for the final consumption. The consumer market is dominated by the products and services that are specifically designed and manufactured for the general consumers. For example, FMCG products and other retail products are a part of consumer market. Business/Industrial market includes the businesses (both private and public institutions) which buy products and services for their own operations or to use them in their own product and services. For example, when Apple purchases Intel processing chips from Intel for using them in its devices it becomes a part of business market for Intel.

Digital marketing is the method of developing a brand, service, or product on the internet. Digital marketing differs from traditional marketing in that it requires the use of online platforms and strategies that enable companies and organisations to monitor the effectiveness of their marketing campaigns, while in traditional marketing, each marketing campaign is evaluated individually.

Most businesses have established a web presence in the twenty-first century. E-mail was widely used, and technology made it possible for people to do it reasonably easily. For a long time, organisations have used customer relationship management systems to handle their databases. Some businesses used banner ads on their websites in a similar manner to how they advertise in the news. Forward-thinking businesses were developing their search engine strategies and collaborating with affiliates. All of this was online marketing, and online marketing teams and experts will appear in a very short time.

Marketers use digital platforms to direct prospects through their buying process and keep in touch with their current clients, in addition to traditional marketing channels such as television, newspapers, advertisements, and so on.

Marketing teams create content, videos and other assets to answer questions or provide context to consumers throughout the three stages of the buyer’s journey:

- **The stage of awareness:** the buyer recognises that they have a need.
- **The evaluation stage:** Buyer chooses a course of action to satisfy their needs.

- **The decision stage:** The buyer selects a product or service to satisfy the requirement.

For example, a customer can discover that they need new gym shoes. An activewear company's marketing department could write a piece about what features you need in a running shoe versus what you need for strength training. The buyer decides that they require pairs of running shoes that meet the requirements based on material. Another piece of content could include a list of the most common running shoes, as well as their price ranges. They determine after they have been informed on these factors. Content marketing is often less costly than other types of marketing, and it generates almost three times the number of leads.

4.2 CONSUMER BEHAVIOUR AND INFORMATION SEARCH

It Refer to the habits or attitudes of a consumer that influences the buying process of a product or service. There are different types of studies are involved to read out the consumer behaviour. Every business concern has a different policies or methods to study the consumer behaviour. Consumer behaviour is dynamic in nature so they adopt the particular strategy to find these changes. Consumer behaviour impacts virtually every stage of the buying process specifically in relation to digital environments and devices.

In our daily life customer need different types of information to understand the things and make decisions. Consumer before finalize his buying decision use various kinds of information in order to get the answers of what to buy, from whom to buy, how to buy and many more related questions. In the past customers having limited sources of information and difficult for them to find out, but at present with the advancement of technology customer can get any information as per requirement from the ample sources of information. It is not only the consumers, have been searching the information, marketers too need information regarding the consumer to develop the marketing strategies and other business decisions. So, marketers and business organization have been relying on the different online sources to get the related information.

4.3 BUYING MOTIVES DECIDE THE INFORMATION SEARCH

Consumers are going to behave differently to satisfy their needs, consumer behavior basically starts with the needs. The consumer’s needs may differ at various points of times. The hierarchy of need theory has been given by the Abraham Maslow, he has classified the needs in five stages i.e., Basic need, Safety need, Social need, Esteem needs and Self Actualization need. The persons satisfy their needs in sequences; initially they satisfy their basic needs then move to next. In a same way to satisfy his self-actualization need, he has to satisfy rest of the need first. It is all because of these different need's consumers behave differently. D. J. Durian has defined the Buying Motives as "those influences or factors which provide the impulse to buy, induce action or determine choice in the purchase of goods and services." Buying motives are those motives which induce the consumers to select or buy particular product to satisfy needs. In another way buying motives are reasons which are satisfied by purchase of the commodities.

As we are clear that buying motive is the motive or urge to satisfy the needs that make customers to buy goods. There are buying motives behind every purchase. Buying motives are the thoughts, feelings and instincts which create a desire in the customers to buy an article. A person is not buying because the sales men have influence them to buy, they are buying due to desire that have taken place in them.

Identifying buying motives of the consumer is a difficult task for the marketers, as there are number of factors to induce consumers to buy particular product. The different experts have given different classification of buying motives. We have done the classification on the bases of:

1. Product Motives
2. Patronage Motives.

1. **Product motives:** Product motives are those motives which are related to the product that induce or prompt the consumers to choose and buy the product. Product motives are related with the attributes of the product. They include physical attraction of product (design, colour, shape, package, price etc.) and psychological attraction of the product (life style and status associated with the product).

   Product buying motives may be further subdivided in to four groups:

   - Emotional Product motives
   - Rational product motives
   - Operational product motives
   - Socio-psychological motives

   - **Emotional Product Motives:** Emotional product motives are those which induce the persons emotionally to buy a product either they need or not or even buy the product without evaluating its attributes. When buyers choose or buy a product without thinking over the matter logically, they are said to be influence by emotional product buying motives. The emotional product motives include:

- ✓ Pride is the strongest emotional motive used by the sellers to induce the buyer to buy product. Many consumers are proud of possessing some products, for example diamond merchants sell their products by using pride or prestige motives.
- ✓ Emulation is another emotional buying motive used by the sellers to sell the products to customer. For example, a girl may like to have same product for a simple reason that all her friends have the same.
- ✓ Affection for other is one of the important emotional buying motives inducing the customers to choose or buy the product. For example, a father buys a watch for his son out of his affection.
- ✓ Desire for comfort is another emotional motive used by the sellers. Many customers buy the product only because of comfort.
- ✓ Sex appeal is another emotional motive used by the marketers to sell the product. Consumers use the product, as they want to be attractive to the members of the opposite sex.
- ✓ Ambition is also used as emotional motive by the sellers to induce the consumers to buy the product. Out of ambition some consumers buy the products.
- ✓ Sometime customers buy the product, because they want to have the product which is not possessed by others.
- ✓ Many consumers buy a particular product because of habit.

For example, Jealousy is an emotional motive of persons, the larger percent being the ladies. It is feeling of hatred, envy or inferiority complex in matters of beauty, wealth, achievement and possession. That someone else is lucky, successful, deserving impress upon the neighbour, friends, colleagues and relatives, people may be induced to buy sarees, ornaments, refrigerators, cars, buildings, paintings, furniture etc. It will not be out of the place here to mention the advertisement by Onida T.V. Company which says, ‘Neighbour’s Envy, Owner’s Pride’.

- **Rational Product motives:** As compare to the emotional motives there are a rational motive which induces the customers to buy the products. Here the consumers make the purchase decisions with logical analysis of product's attributes. When customers buy the product after careful consideration, they are said to be influence by rational product motives. A rational Product motive includes:
    - ✓ Desire for safety is an important rational buying motive to induce the consumers. For example, we go for safety locker in bank; it is all because of this motive. In a same way we buy Vitamin tablets or medicines due to this reason.
    - ✓ Saving money while buying the product is the rational motive to influence the customers. Mostly consumers buy Honda bikes due to high mileage.
    - ✓ Relatively low price is one of the rational buying motives; customers buy the products which are relatively cheaper.
    - ✓ Suitability of the product for the needs another rational motive as we know the intelligent persons buy the product based on their suitability.
    - ✓ How much the product is durable another rational motive which induces the customers.
    - ✓ The convenience of the product is another strongest rational product motive to influence the person to buy the product.

- **Operational Product Motives:** Operational product motive is concerned with the satisfaction derived from the functions of products and the utility of the product. The products having more utility for the customers or products having more satisfaction power will be more preferred by the customers. Hence the satisfactions from product or product's utility induce the persons to buy the product.

- **Socio-Psychological Motives:** Socio-psychological motives are different from the above three motives. In this customer attracted toward the products because of prestige attached with product. Thus, the produced are evaluated by the consumers on the bases of social status. Customer would prefer to buy product which is socially acceptable.

2. **Patronage buying motives:** These motives refer to those reasons which influence the customers to buy wanted product from a particular seller or shop. Many times, we see most of the times customers are going to buy the product from particular shop, there are number of motives or reasons behind this kind of behaviour. Patronage buying motives are further classifies into two groups:
   - **Emotional Patronage motives**
   - **Rational Patronage motives**

- **Emotional Patronage Motives**: In this motive customer buy the wanted product from a particular shop without evaluating the reason; why that particular shop to buy product. Here the reason for buying from that particular shop is purely a subjective. The emotional patronage motive includes:
  - ✓ Appearance of the shop one of the strongest motives to induce the customer to buy from particular shop. Customers buy goods and services from particular shop because of attractive appearance of the shop.
  - ✓ Seller display the products in the shop in such a way, the customers get attracted toward that shop for buying the goods and services.
  - ✓ Recommendations of the others like friends or family members also induce the customer to go for shopping in that particular shop.
  - ✓ Sometime customers go to a particular shop for shopping because other customers are also going to that particular shop for shopping.
  - ✓ Prestige is one of the strongest patronages buying motives of the buyers. For example, customers prefer to go five-star hotels for coffee all because of prestige.
  - ✓ Sometime customers always go to particular shop for buying product, because they are habitual to buy from that particular shop.

- **Rational Patronage buying Motives:** On the other hand, the customers have the logic for buying from a particular shop in case of rational patronage motives. In this motive the customers are aware what are the advantages attached with a particular shop in the form of wide selection, quality material, after sales services etc. we can include in the rational patronage motives:

- ✓ Convenient location of the shop which induce the buyer to buy product from particular shop.
- ✓ If the seller offers discounts or charge less price for the product, customer will prefer to go there for shopping.
- ✓ The credit services provided by the shop also induce the customers.
- ✓ The before and after sales services provided by the shop also influence the customers to go for a particular shop.
- ✓ The efficiency of the seller's staff to help the customer in buying process also induce the customer to go that shop for buying.
- ✓ The variety of quality products in the shop is one of the motives making the buyer patronise a particular shop.
- ✓ The reputation of the shop in society is also one of the rational motives to induce the customer.

CHECK YOUR PROGRESS

I. In Today's World "The customer is the ……….. of the Market", So every company is making efforts to win the customer.

a) Promoter
b) User
c) Producer
d) King

II. …………………. comprises of the individuals and households who purchase the product and services the service for the final consumption.

a) International market
b) Consumer market
c) Business Market
d) Non-profit organisation market

III. Digital marketing is the method of developing a brand, service, or product on the newspaper. (True /False)

IV. Most businesses have established a web presence in the …………. century.

V. Consumer behaviour impacts virtually every stage of the buying process specifically in relation to digital environments and devices. (True /False)

VI. The consumer's needs may differ at various points of times. (True /False)

VII. Identifying buying motives of the consumer is a easy task for the marketers, as there are number of factors to induce consumers to buy particular product. (True /False)

VIII. ……………….. are those motives which are related to the product that induce or prompt the consumers to choose and buy the product.

IX. ……………… motives are those which induce the persons emotionally to buy a product either they need or not or even buy the product without evaluating its attributes.

X. …………………. motive is concerned with the satisfaction derived from the functions of products and the utility of the product

a) Emotional product
b) Operational product
c) Both
d) None of these

4.4 DIFFERENT INFORMATIONS REQUIRED FOR DIFFERENT BUYING ROLES

Most of the time the purchases are made by the individuals, but some time it may be done one a group like household. The number of individuals may interact to affect or influence the purchase decision. Depending upon the products, the persons involved in the process of purchase have to play different role or some time one person has to play all the roles required in buying process.

The five different roles which are required in buying process are discussed as:

- **Initiator:** The persons who start the process of considering a purchase is the initiator, as they are the persons who wants product to satisfy their needs. For example, a student may act as initiator for a laptop that he needs for education motive, a girl may feel the need of beauty products.
- **Influencer:** The influencers are going to influence the initiator's decision that needs products to satisfy their needs. A marketer needs to focus the influencer because they are going to alter the behavior of person who need product. For example, a student taken the initiative to buy the motorcycle, the elder brother or father in the family may influence him to buy the scoter not the bike.
- **Decider:** Decider is the person who is going to decide whether to buy the product or not, actually he is the person who is going to pay for the product. For example, in a family father is the payer, thus he is having a great influence in the buying process.
- **Buyer:** The person who actually goes to the market and buys the product. Usually, the buyer is the person to whom marketer can see in the market. But evaluating the

behavior of buyer cannot fulfill the purpose of marketers because the decisions regarding product, brand, shop etc. have already been taken at the end of decider.

- **User:** the person who is actually going to use or consume the purchased product. Most of the time it is the initiator who use the product. The product may be consumed by the individual or it can be consumed in a group.

Here it is important to note that who is going to play what role all depends upon the Product's type, its price, buying capacity and many more variables. For example, a student has to buy the note book for study and he has enough pocket money. In this case a student may play all the roles required in buying process. In another example a student need motorcycle, the different persons involved in buying may play different roles, like student himself act as initiator, his friends or bother may play the role of influencer to influence him what kind of bike he should buy, Parents may play the role of decider who are actually going to pay for the bike and at the end student who is going to ride the bike may act as user. In the same way if the student's elder brother has taken the initiative regarding bike for commuting college to home, elder bother will be considered as initiator or student for whom the bike has been purchased would be considered as user. It is not necessary the role of initiator and user played by one person. So, who is going to play what role all depends on the product type, how much expenditure will be done for the product, by whom the product would be used and many more factors.

Few more examples of Different role in buying process:

1. A kindergarten student needs to buy colour crayons to use in class.

- ✓ Initiator: The student
- ✓ Influencer: His teacher or his classmates
- ✓ Decider: Either of the parents
- ✓ Buyer: Either of the parents or a sibling.
- ✓ User: The student himself.

2. A boy enters college and needs a laptop for doing assignments.

- ✓ Initiator: The boy himself
- ✓ Influencer: His friends and classmates
- ✓ Decider: The boy himself
- ✓ Buyer: The boy himself

- ✓ User: The boy himself

4.5 CUSTOMER VALUE

Customer Value relates to any organization's Customer, and what they feel while buying and using their products. It could be defined as "the difference between the value the customer's gain from owning it and using the product and the cost of obtaining it. Woodruff defines customer value as a "customer perceived preference for and evaluation of those products, attributes, attribute performances and consequences arising from use that facilitate (or block) achieving customers goals and purposes in use situations. Customer value is of two types, desired value and perceived value. Desired value means what the customer wants or desires in a product while the perceived value represents the benefits a customer receives after buying the product.

Delivering customer value is a continuous effort as the Value of a product keeps on changing according to the needs and wants of the product. According to *Steve Jobs "You can't just ask customer's what they want and then try to give that to them. By the time you get it built, they'll want something new."*

Classification of customer value: Customer value can be classified into four types:

- **Functional value:** it relates to what solution a product provides to the customer.
- **Monetary value:** it refers to whether the function of the product relates to the price of the product or the product is worth the price paid for it.
- **Social value:** It refers to how the product helps the customer to connect with other people in the society.
- **Psychological value:** the extent to which the product allows consumers to express themselves or feel better in their life.

Customer Perceived Value

The consumers of any product are usually unaware of the production costs of that product. They simply have an internal feeling about the worth of that product, which is based on total customer benefits and total customer cost associated with it. This worth of a product or service in the mind of the customer is known as customer perceived value. This perceived value often affects the price that a customer is willing to pay for a product. In order to pay a higher price for their products various companies apply marketing strategies to create a higher perceived value for their product. For example, perfumes are often associated with big celebrities to create a higher brand value in the minds of the customers.

Customer Perceived Value = Total customer benefit + Total customer cost

Total customer benefit= Product benefit + Service benefit + Personnel benefit + Image benefit.

Total customer cost = Monetary cost + Time cost + Energy cost + psychological cost.

RFM Analysis

It is a marketing analysis tool often used to identify any organizations loyal customers. This analysis is based on three quantitative factors.

- **Recency:** It means "how recently "or the time elapsed since the last purchase was made by a customer or an advertisement was last published or broadcasted.
- **Frequency:** It refers to how often a customer purchases a particular product.
- **Monetary value:** it refers to the amount of money a customer spends on purchasing a product. RFM analysis often supports the marketing adage that "80% of business comes from 20% of the customers."

Customer lifetime Value

In marketing, customer lifetime value (CLV) is a metric that represents the total net profit a company makes from any given customer. CLV is a projection to estimate a customer's monetary worth to a business after factoring in the value of the relationship with a customer over time. CLV is an important metric for determining how much money a company wants to spend on acquiring new customers and how much repeat business a company can expect from certain consumers.

4.6 CUSTOMER SATISFACTION

Custom satisfaction is a measure of how the products and services supplied by a company meet the customer's expectations. It is measured and seen as a key performance indicator for measuring the success or failure of any business enterprise. If we want to retain our customer's then they should be satisfied with our products.

Loyal customers, they don't just come back, they don't simply recommend you, they insist that their friends do business with you.

It is seen as a key performance indicator within business and is often part of a Balanced Scorecard. In a competitive marketplace where businesses compete for customers, customer satisfaction is seen as a key differentiator and increasingly has become a key element of business strategy

Importance of customer satisfaction:

- Key performance indicator
- Key element of business strategy
- Indicator for consumer purchase intentions.
- Level of satisfaction varies from person to person and from time to time.
- Indicator for good quality product and services.
- Higher customer satisfaction can lead to higher economic growth.
- Higher satisfaction is directly proportional to greater customer loyalty.

Know what customers want most and what company does best, so, focus on where these two meets.

Customer satisfaction measures how well the expectations of a customer concerning a product or service provided by company have been met. Customer satisfaction is an abstract concept and involves such factors as the quality of the product, the quality of the service provided, the atmosphere of the location where the product or service is purchased, and the price of the product or service. Businesses often use customer satisfaction surveys to gauge customer satisfaction. These surveys are used to gather information about customer satisfaction. Typical areas addressed in the surveys include:

- Quality of product
- Value of product relative to price - a function of quality and price
- Time issues, such as product availability, availability of sales assistance, time waiting at checkout, and delivery time
- Atmosphere of store, such as cleanliness, organization, and enjoyable shopping environment

Studies carried out by companies like Argos and Cadburys have found very high levels of customer satisfaction. It is not surprising because these companies emphasize market research and marketing as the tools to find out what customers want. Knowing what customer wants then makes it possible to tailor everything do to pleasing the customers e.g., providing the goods that customers want, in the packaging that they want, in retail outlets which are convenient to use and well placed.

There are many factors which lead to high levels of customer satisfaction including:

- Products and services which are customer focused and thence provide high levels of value for money.
- Customer service giving personal attention to the needs of individual customers.
- After sales service, following up the original purchase with after sales support such as maintenance and updating services given with the core product (for example in the updating of computer packages).

What is clear about customer satisfaction is that customers are most likely to appreciate the goods and services that they buy if they are made to feel special. This occurs when they feel that the goods and services that they buy have been specially produced for them or for people like them. This relates to a wide range of products such as razors that are designed for ease of use and good quality finish, petrol products that are environmentally friendly and customized to meet the needs of particular types of engines, etc.

CHECK YOUR PROGRESS

XI. The persons who start the process of considering a purchase is the …………

a) Influencers c) Initiator
b) Deciders d) Buyer

XII. The …………………. are going to influence the initiator's decision that needs products to satisfy their needs.

XIII. ……….. is the person who is going to decide whether to buy the product or not, actually he is the person who is going to pay for the product.

a) Influencers c) Initiator
b) Deciders d) Buyer

XIV. Customer value is of two types, desired value and ………………..

XV. Delivering customer value is a continuous effort as the Value of a product keeps on changing according to the needs and wants of the product. (True /False)

XVI. ………………… relates to what solution a product provides to the customer.

a) Monetary value
b) Functional value
c) Psychological value
d) Social value

XVII. ……………. means "how recently "or the time elapsed since the last purchase was made by a customer or an advertisement was last published

XVIII. ……………….is a marketing analysis tool often used to identify any organizations loyal customers.

XIX. Customer lifetime value (CLV) is a metric that represents the total net profit a company makes from any given customer. (True /False)

XX. Custom satisfaction is a measure of how the products and services supplied by a company meet the customer's expectations. (True /False)

4.7 MONITORING AND MEASURING CUSTOMER SATISFACTION

The increasing market competition and growing customer focus has resulted in measuring the level of customer satisfaction related to a product or service. Higher degree of customer satisfaction is said to be an indicator for customer's future profits. It can be measured in case of both goods and services. Satisfied and delighted customers are believed to be profitable for any business enterprise, it helps to determine companies focus areas for improving service quality and identifying the gaps. It helps us to understand whether the organization is meeting customer expectation and delivering the required standard of quality services.

Customer behaviour is one of the best ways for measuring customer satisfaction. There can be two ways of measuring customer satisfaction.

- **Direct Ways:** It is done by directly approaching the customers for their valuable feedback (personal calls, face to face interviews, questionnaires etc.)
- **Indirect ways:** This can be obtained by a) complaint reports of customers regarding any specific product or service. If the complaint reports under a particular segment are high, it is an indication of low performance by the company. If the complaints are low that means the organization is performing well during that period of time. b) Customer loyalty: if the customer is coming back and buying the same product that means it loyal to the product and gives an indication of positive customer satisfaction.

Direct ways of obtaining customer feedback

- **Event based survey:** This type of survey is done during or after the customer service period on a regular and a continuous basis with short 3-5 questions.
- **One time/One off Survey:** This type of survey is performed for specific reasons example, a change in service provider, etc. It is usually done after a major change has been done in the organization.
- **Focus Groups:** It is a special group od people who meet and discuss on predetermined issues. One person reports all the points and sums up all the important findings.
- **Periodic/Annual survey:** It is a planned and a scheduled procedure on a periodic basis, mostly done quarterly or annually. In each periodic survey the elements to be measured remain the same. This type of survey helps the organization in comparing its own services over a period of time.

The Process of measuring customer satisfaction:

- To ensure the reliability and validity of measures choose a team of researchers.
- Select an appropriate method for obtaining feedback e.g. telephone, interviews, face to face interviews, questionnaires etc.
- Select the sample of customers for obtaining feedback.
- Select the attributes to be measured and a measurement scale.
- Carry out the research.
- Analyse the results.
- Adopt corrective measures.

4.8 DELIVERING CUSTOMER VALUE

One of the most common reasons for the failure of a business venture, is its inability to deliver value to customers. The concept of value is one of those things that is both simple and complex. Simple because it only has three components; complex because it can only be defined by the customer, and can include tangible and intangible concepts such as perceptions and opinions.

Consumer having wants and resources (financial ability), they demand products and services with benefits that add up to the most value and satisfaction.

The four types of value include: *functional value, monetary value, social value,* and *psychological value*. The sources of value are not equally important to all consumers. How important a value is, depends on the consumer and the purchase. Values should always be defined through the "eyes" of the consumer.

- **Functional Value:** This type of value is what an offer does, it is the solution an offer provides to the customer.
- **Monetary Value:** This is where the function of the price paid is relative to an offering perceived worth. This value invites a trade-off between other values and monetary costs.
- **Social Value:** The extent to which owning a product or engaging in a service allows the consumer to connect with others.
- **Psychological Value:** The extent to which a product allows consumers to express themselves or feel better.

For a firm to deliver value to its customers, they must consider what is known as the "total market offering". This includes the reputation of the organization, staff representation, product benefits, and technological characteristics as compared to competitor's market offerings and prices. Value can thus be defined as the relationship of a firm's market offerings to those of its competitors.

The term value may mean, low price, receiving what is desired, receiving quality for what is paid, or receiving something in return for what is given (Zeithaml). Evaluation of customer value can be done at different levels.

At a lower level, customer value can be viewed as the attributes of a product that a customer perceives to receive value from. At a higher level, customer value can be viewed as the emotional payoff and achievement of a goal or desire. When customers derive value from a product, they derive value from the attributes of the product as well as from the attribute performance and the consequence of achieving desired goals from the use of the product. Thus, delivering value to the customers includes all the steps a customer will go through while obtaining the product and while using the product.

Points to be remembered while delivering customer value:

- Understand the value of a product for the customer.
- The value should be provided at both the stages during the buying procedure and after purchasing the product.
- The value should be delivered at all the steps.
- The value for a particular product may vary from time to time.

4.9 ROLES OF INTERNET MARKETERS

Internet marketers, the essentiality of taking online businesses to greater heights. The generating of marketing leads, developing/expanding customer base, tracking sales conversions and evaluating overall web marketing and promotional campaigns are the many roles that an Internet marketer plays. Being in a team of players, the website developer, the programmer and the website administrator, the Internet marketer is often involved in the creation of web content which showcases the overall aesthetics of the website and the business.

Search Engine Optimization (SEO) and Search Engine Marketing (SEM) are terms that Internet marketers are familiar and close with. The difference is that SEO is used to generate organic traffic while SEM involves paying or sponsoring the traffic (such as a Google AdWords & PPC campaign). In order to fulfill the roles of the above, the Internet marketer has to be adept in keyword research, being able to bring out the accuracy and niche keyword and phrases. Analyzing them comes as the next step, to strategies which keywords would help anchor an online web marketing campaign success.

Internet marketers have to grasp the concept of link building, on top of SEO and SEM. The main goal is to constantly compete in page ranking in search engines like Google. Indeed, it is an arduous process of generating link juices which usually involve search engine and directory submission, bookmarking and campaigning for reciprocal link building.

Measuring marketing performance and tracking conversions is another important tool that Internet marketers have to be equipped with. The combination of Google analytics and Google AdWords will be able to achieve that, including the tracking of behavioral patterns of website visitors such as bounce rate, browser capabilities and mobile device accessibility.

Without forgetting the entrance of social media, the Internet marketer has to expand its roles in managing corporate accounts of different social media networks. Social networking sites like Facebook, Twitter, YouTube are fertile grounds available for the expansion of brand awareness, generation of marketing leads and web traffic. In addition, the recent trend to go local, Geo-location-based marketing, has further broadened the scope of web marketers.

What is described above is a complex and highly technical job scope, which does not involve skills and trainings alone. But, with the passion and love for the journey, the Internet marketer is able to excel to his/her best abilities.

CHECK YOUR PROGRESS

XXI. Customer behaviour is one of the best ways for measuring customer satisfaction. (True /False)

XXII. …………… is a special group of people who meet and discuss on predetermined issues.

XXIII. ………………… is a planned and a scheduled procedure on a periodic basis, mostly done quarterly or annually.

XXIV. One of the most common reasons for the failure of a business venture, is its inability to…………………...

4.10 LET US SUM UP

Consumer market comprises of the individuals and households who purchase the product and services the service for the final consumption. The consumer market is dominated by the products and services that are specifically designed and manufactured for the general consumers. Marketers use these digital platforms to direct prospects through their buying process and keep in touch with their current clients, in addition to traditional marketing channels such as television, newspapers, advertisements, and so on. In our daily life customer need different types of information to understand the things and make decisions. Consumer before finalize his buying decision use various kinds of information in order to get the answers of what to buy, from whom to buy, how to buy and many more related questions. In the past customers having limited sources of information and difficult for them to find out, but at present with the advancement of technology customer can get any information as per requirement from the ample sources of information. Consumers are going to behave differently to satisfy their needs, consumer behavior basically starts with the needs. Identifying buying motives of the consumer is a difficult task for the marketers, as there are number of factors to induce consumers to buy particular product. The different experts have given different classification of buying motives. Most of the time the purchases are made by the individuals, but some time it may be done one a group like household. The number of individuals may interact to affect or influence the purchase decision. Depending upon the products, the persons involved in the process of purchase have to play different role or some time one person has to play all the roles required in buying process. Customer Value relates to any organization's Customer's and what they feel while buying and using their products. As far as Custom satisfaction is a measure of how the products and services supplied by a company meet the customer's expectations.

4.11 KEY WORDS

CONSUMER BEHAVIOUR	It Refer to the habits or attitudes of a consumer that influences the buying process of a product or service.
BUYING MOTIVES	Factors which provide the impulse to buy
PRODUCT MOTIVES	Motives which are related to the product that induce or prompt the consumers to choose and buy the product.
PATRONAGE BUYING MOTIVES	Those reasons which influence the customers to buy wanted product from a particular seller or shop.
INITIATOR	The persons who start the process of considering a purchase.
CUSTOMER VALUE	Customers' feel while buying and using their products.
FUNCTIONAL VALUE	It relates to what solution a product provides to the customer.
CUSTOMER SATISFACTION	It's a measure of how the products and services supplied by a company meet the customer's expectations.

4.12 ANSWERS TO CHECK THE PROGRESS

I. King
II. Consumer market
III. False
IV. Twenty-first
V. True
VI. True
VII. False
VIII. Product motives
IX. Emotional product
X. Operational product
XI. Initiator
XII. Influencer
XIII. Deciders

XIV. Perceived value
XV. True
XVI. Functional value
XVII. Recency
XVIII. RFM Analysis
XIX. True
XX. True
XXI. True
XXII. Focus Groups
XXIII. Periodic/Annual survey
XXIV. Deliver value to customers

4.13 TERMINAL QUESTIONS

- Discuss in detail how digital marketing has contributed to make consumer as king of the market?
- What is consumer behaviour? Highlight the behaviour of the consumer while searching for the information.
- What kind of information required by the consumer for finalising product to buy? What is the role of buying motives to decide the information required?
- Do consumers use the same kind of information irrespective of their buying role?
- What is customer value and why it is important for the marketers? Do customer value and customer satisfaction are the same?
- What is RFM analysis and why it is used?
- What is customer satisfaction and how we can measure the satisfaction level of consumer?
- How we can insure the delivery of customer value? What is the role of internet marketers for the same?

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-5 FACTORS INFLUENCING CONSUMPTION BEHAVIOR

STRUCTURE

5.0 OBJECTIVES

After studying the Unit, you would be able to

- Describe the Consumer Behavior
- Find out how the internal influencers impact the consumer behavior
- Explore the role of public relationship in consumer behavior
- Utilizes the public relationship for marketing
- Highlight the dimensions of public relationship
- Evaluate, how environment influence the consumer behavior
- Elaborate the CRM and its best use in marketing
- Identify the reasons for losing customers

5.1 INTRODUCTION

Digital marketing is normally associated with Internet marketing. Internet marketing captures data which feeds into the firm’s database; the database is used to generate profiles and lists, which enable the firm to have effective direct marketing campaigns. Underlying electronic business has two phenomena: digitalization and connectivity. Digitalization consists of converting text, data, sounds, and image into a stream of bits that can be dispatched at incredible speeds from one location to another. Connectivity involves building networks and expresses the fact that much of the World’s business is carried over networks connecting people and companies. These networks are called intranets when they connect people within a company; extranets when they connect a company with its suppliers and customers; and the Internet when they connect users to an amazingly large information superhighway.”

5.2 CONSUMER BEHAVIOUR

Consumer behaviour is described as the action and reaction of the consumer while searching, evaluating, buying and consuming the product to satisfy their needs. Consumer behaviour focuses on how individuals make decisions to spend their resources (time, money and effort) on products and services for consumption and satisfaction purpose. As individuals we all differ in many ways, in likes, dislikes, attitudes, cultural background, income level, education, occupation, family background etc. Despite such differences we are all consumers.

If a marketer can identify online consumer’s buying behaviour, he or she will be in a better position to target products and services at them. Online Buyer behaviour is focused upon the needs of individuals, groups and organisations.

It is important to understand the relevance of human needs to buyer behaviour (remember, marketing is about *satisfying needs*). Let us look at human motivations as introduced by Abraham Maslow in his hierarchy of needs: The hierarchy is triangular. This is because as you move up it, fewer and fewer people satisfy higher level needs. We begin at the bottom level.

Physiological needs such as food, air, water, heat, and the basic necessities of survival need to be satisfied. At the level of safety, man has a place to live that protects him from the elements and predators. At the third level we meet our social and belongingness needs i.e., we marry, or join groups of friends, etc.

The final two levels are esteem and self-actualisation. Fewer people satisfy the higher-level needs. Esteem means that you achieve something that makes you recognised and gives personal satisfaction, for example writing a book. Self-actualisation is achieved by few. Here a person is one of a small number who actually do something. For example, Neil Armstrong self-actualised as the first person to reach the Moon. The model is a little simplistic but introduces the concept a differing consumer need quite well.

To understand consumer buyer behaviour is to understand how the person interacts with the marketing mix. As described by Cohen (1991), the marketing mix inputs (or the four Ps of price, place, promotion, and product) are adapted and focused upon the consumer.

The psychology of each individual considers the product or service on offer in relation to their own culture, attitude, previous learning, and personal perception. The consumer then decides whether or not to purchase, where to purchase, the brand that he or she prefers, and other choices.

5.3 INTERNAL INFLUENCERS TO IMPACT THE CONSUMER'S BEHAVIOUR

- **Personality:** Some marketers believe customers choose products that express their personalities. Personality is defined as the thoughts, emotions, intentions and behaviour that people express as they move through their environment. Personality is unique to individuals, but may be applied to groups, is a combination of characteristics and traits and influences purchasing behaviours. Marketers use interviews and focus groups to understand personality and how it relates to the purchase of certain products in online market.
    - ✓ **Frugality:** consumers restrain themselves and think heavily about purchases
    - ✓ **Impulsiveness:** purchases are made without much thought beforehand
    - ✓ **Anxiety:** a person with lots of anxiety may have more post-purchase dissonance and feel upset about purchases after they get them home
    - ✓ **Bargaining:** some consumers prefer to bargain for purchases, it gives them a sense of control over their spending
    - ✓ **Vanity:** taking excessive pride in one's appearance and accomplishments
    - ✓ **Competence:** being responsible and dependable
    - ✓ **Ruggedness:** craving products that are tough and strong
    - ✓ **Sincerity:** honest and genuine
    - ✓ **Excitement:** craving daring and spirited purchases
    - ✓ **Sophistication:** desiring products that are glamorous and prestigious
- **Motivation:** Motivation is an internal state that drives consumers to satisfy needs. Motivation is the energizing force that could be used by marketers to activate online consumer's behaviour. Once we recognize that we have a need, a state of tension

exists that drives the consumer to the goal of reducing this tension and eliminating the need. Consequently, only unmet needs motivate.

- **Memory:** Marketing messages can be effective only if the consumer correctly understands the online messages, and remembers them when needed. Memory refers to a consumer's ability to understand the online marketing messages and assign them value and meaning. Value and meaning has to come together to make memory an effective influencer.
  - ✓ **Colour:** Colours have an enormous impact on marketing messages, and it affects consumers in a subjective manner, so that most of the time consumers do not even know they are being affected.
  - ✓ **Font:** The presentation of words and how they are shaped online, will also enhance the marketing message and contribute to the value and meaning.
  - ✓ **Simplicity of the message:** A simple message is generally easier to understand. Using short phrases and easy to read terminology, such as "heart healthy" will quickly and easily convey the message that the product is good for your heart.
  - ✓ **Consistency of the message:** The message needs to fit with the surrounding information; style, colour, text, photographs, music and all have to work together.
  - ✓ **Source of the message:** Consumers will be more likely to remember and purchase products endorsed by credible sources, animated characters or celebrities. They have to be likeable, have some expertise or at least pretend that they have expertise, be trustworthy, and attractive.

5.4 ROLE OF ONLINE PUBLIC RELATIONSHIP IN CONSUMER BEHAVIOUR

In the mid-twentieth century, mass production techniques and mass marketing changed the competitive landscape by increasing product availability for consumers. However, the purchasing process that allowed the shopkeeper and customer to spend quality time getting to know each other was also fundamentally changed. Customers lost their uniqueness, as they became an "account number" and shopkeepers lost track of their customers' individual needs as the market became full of product and service options. Many companies today are racing to re-establish their connections to new as well as existing customers to boost long-term customer loyalty. Some companies are competing effectively and winning this race through the implementation of relationship marketing principles using strategic and technology-based public relationship (PR) applications.

Public Relations in its true sense are about human connections and the art of mastering human connections at a deep level. In the early days of PR, it was about relationships with not just the press but communities in various forms; the only difference was that these audiences were not online.

Online PR is very similar to traditional PR in the sense that it's about influencing people rather than buying placement for brand content. The influence could result in a story in a magazine, newspaper or blog. It could also result in other online pick-up, including social media. Typically, when people talk about traditional PR, they are referring to traditional media: newspapers, TV, radio and magazines. With online PR, traditional media brands may still be a target, but often online PR targets online properties, along with an array of other platforms and networks, from search to social.

CHECK YOUR PROGRESS

I. Digital marketing is normally associated with Internet marketing. (True/False)

II. Consumer behaviour does not focus on how individuals make decisions to spend their available resources on products and services for consumption and satisfaction purpose. (True/False)

III. If a marketer can identify online consumer's buying behaviour, he or she will be in a better position to target products and services at them. (True/False)

IV. ………….. desiring products that are glamorous and prestigious.

V. In …………….. consumers restrain themselves and think heavily about purchases.

a) Anxiety c) Frugality

b) Impulsive d) Compulsive

VI. In ……………… a person with lots of anxiety may have more post-purchase dissonance and feel upset about purchases after they get them home.

a) Impulsive c) Frugality

b) Compulsive d) Anxiety

VII. ……………… is the energizing force that could be used by marketers to activate online consumer's behaviour.

VIII. In the mid-twentieth century, mass production techniques and mass marketing changed the competitive landscape by increasing product availability for consumers. (True/False)

IX. ……………. in its true sense are about human connections and the art of mastering human connections at a deep level.

5.5 DISTINCTION BETWEEN MARKETING AND PUBLIC RELATION

- Marketing promotes the transfer of goods and services from the producer and provider to the consumer. Public relations help an organization and its publics adapt mutually to each other.
- Marketing's immediate goal is sales. Public relations' immediate goal is mutual understanding or positioning of the organization with its publics.
- Marketing's implicit goal is profit. Public relations' implicit goal is positive perceptions and predispositions.
- Marketing's measure of success is the number of sales and/or the revenue it generates. Public relations' measure of success is expressed public opinion or other evidence of public support.

5.6 BENEFITS OF ONLINE PR

A successfully executed online public relations strategy offers endless benefits to a business. By listening and researching online conversations and key influencers, creating compelling content and interacting with others online within online communities, a company can effectively use Digital PR to enhance the business and achieve growth. Some of the benefits of Digital PR include the following:

- Increased traffic to a website, thus increased web conversion rates
- Contribution to search engine optimisation by generating links back to a website as well as enhancing website authority
- Enhanced brand awareness
- Position a company as an authoritative voice in their industry
- Provide an avenue for improved customer relations by allowing a company to directly engage with Individuals interested in their brand or product

- Provide a platform to communicate information and/or company insights to target audience

5.7 COMPONENTS AND TOOLS OF PUBLIC RELATIONS

- **Public:** A group of similar individuals; an assortment of persons having the same interests, problems, circumstances, goals; it is from such persons that opinion emanates. Public is a varied creature; it comes in many forms and sizes. Public has a multitude of wants and desires; it has its likes and dislikes, sometimes, strong likes and strong dislikes. Employers make for a public and employees another public; the government is a public and citizens constitute another public, and so on, each of these groups is a public of the sort, tries to attract a different audience with its own tools and techniques.
- **Relations:** Human wants to create the need to establish relations with one another. The representative wants of the individuals will profoundly affect their relationship. To understand any relationship, therefore, one must understand the wants of those involved. 'Relationships are of all possible types. We have relationship by ran-superior to inferior, inferior to superior, and equal to equal. We have relationship by sentiment-benevolent, Friendly, suspicious, jealous, hostile. A relationship may be active, or it may be passive it may be good or it may be bad, or it may be neutral. At any rate, the relationship is there to be accepted, ignored or altered, as desired.
- **Propaganda:** Propaganda is the manipulation of symbols to transmit accepted attitudes and skills. It describes political application of publicity and advertising, also on a large scale, to the end of selling an idea cause or candidate or all three.
- **Campaigns**: These consist of concerted, single-purpose publicity programme, usually on a more or less elaborate scale, employing coordinated publicity through a variety of media, aimed, at a number of targets, but focused on specific objectives. A campaign objective may be the election of a candidate, the promotion of political cause or issue, the reaching of a sales goal, or the raising of a quota of funds.
- **Lobbying**: It entails the exertion of influence, smooth and measured pressure on other, exercise of persuasion cum-pressure. In essence, it means a group putting its points of view forward in an attempt to win the other groups support.

5.8 FUNCTIONS OF PUBLIC RELATIONS

- Public Relations is establishing the relationship among the two groups (organisation and public).
- Art or Science of developing reciprocal understanding and goodwill.
- It analyses the public perception & attitude, identifies the organisation policy with public interest and then executes the programmes for communication with the public.

CHECK YOUR PROGRESS

X. ………… is the manipulation of symbols to transmit accepted attitudes and skills.

a) Campaigns
c) Propaganda
b) Lobbying
d) All three

XI. Public relations' measure of success is expressed public opinion or other evidence of public support. (True/False)

XII. Public relations' immediate goal is ……………… of the organization with its publics.

XIII. A successfully executed online public relations strategy does not offers endless benefits to a business. (True/False)

XIV. ……………. means a group putting its points of view forward in an attempt to win the other groups support.

a) Campaigns
c) Lobbying
b) Propaganda
d) All three

5.9 CHARACTERISTICS OF THE ENVIRONMENT

- **Intensity of information:** If a consumer is overloaded with stimuli in an environment, they are much more likely to avoid the ad, or not comprehend it at all. We live in a world cluttered with advertisements; it can be difficult for companies to break through all of it and get to target market customers. Marketers have to be more creative since customers can now skip commercials; marketers use product placement in the actual movie or television show, the characters in the show use the brand name products and may even talk about how they like the brand name. This is all part of the advertising. Marketers are also making use of new social marketing movements such as Twitter and Facebook that can be programmed to reach customers that want to see your marketing messages.

- **Framing:** Messages can be framed to seem positive or negative and this will affect how customers assign value. “If you don’t use sunscreen, you could get skin cancer” or “Use sunscreen to moisturize and protect your delicate skin”.
- **Timing:** Many factors will influence how a message is interpreted and assigned value including: amount of time customer has to view a message, time of day, and type of medium used. A customer driving in the morning 70mph past a billboard for coffee may only have a few seconds to interpret the message, but since it is a time of day when that product is most consumed, they may be more likely to act on the message.

The value and meaning assigned is largely determined by internal factors, (thoughts, feelings, emotion, attitude, perception, motivation, personality, and lifestyle) which are different for each consumer. For example, a consumer who drinks lots of milk, sees an advertisement that says “Got Milk?” and since they already have positive feelings for the product they will purchase more milk, whereas a consumer who does not enjoy drinking milk and sees the same ad, may dismiss the ad or may try drinking more milk for a short period of time and then decrease consumption again.

5.10 CUSTOMER RELATIONSHIP MANAGEMENT

To be market-leader, companies must be customer-driven. Marketers are focusing on customers because to increase the market share. As Philip Kotler defines, “Marketing is managing profitable customer relationships. The two-fold goal of marketing is to attract new customers by promising superior value and to keep and grow current customers by delivering satisfaction”.

Customer relationship management (CRM) focuses totally on customers and the organisation’s entire gamut of functions related to value creation and value delivery. It is concerned with developing customer loyalty and brand loyalty to the highest possible level, to maintain long-term customer relationship.

Customer relationship management (CRM) is defined as a process of acquiring customers by understanding their needs, retaining customers by fulfilling the needs, and attracting new customers through customer specific strategic marketing approaches.

CRM is beneficial both to the marketers and the customers. The marketers benefit from reduction in customer acquisition cost, more loyal customers, expansion of customer base, savings in advertisement and promotion budget, increase in the number of profitable customers, the future possibility of introducing new products easily in the market and possibility to expand the business. CRM is beneficial to the customers in the form of improvement in service quality by the marketers, more attention, personalized service, less risk, more commitment and increased value for money.

- **Customer Value:** Customer value is the benefit that a customer will get from a product or service in comparison with its cost. This benefit might be measured in monetary terms, such as when a product saves the customer money that would have been spent on something else. A benefit also can be difficult to quantify, such as the enjoyment that a customer receives from a product or service. The term "customer value" should not be confused with the value of customers to businesses. It refers to the value that the customers receive, and does not explain the value of customers for the company.
- **Customer Satisfaction:** Customer satisfaction is a term frequently used in marketing. It is a measure of how products and services supplied by the marketers meet or surpass customer expectation. Customer satisfaction is defined as "the number of customers, or percentage of total customers, whose reported experience with a firm, its products, or its services (ratings) exceeds specified satisfaction goals".

  In researching satisfaction, firms generally ask customers whether their product or service has met their expectations. Thus, expectations are a key factor behind satisfaction. When customers have high expectations and the reality falls short, they will be disappointed and will likely rate their experience as less than satisfying. For this reason, a luxury resort, for example, might receive a lower satisfaction rating than a budget motel, even though its facilities and service would be deemed superior in 'absolute' terms."

  The importance of customer satisfaction diminishes when a firm has increased bargaining power. For example, cell phone plan providers, such as Airtel, Idea, Vodafone and BSNL participate in an industry that is an oligopoly, where only a few suppliers of a certain product or service exist. As such, customers have very low satisfaction level for many cellphone service providers and they would never prefer them if there were, say, 100 cell phone plan providers, because customer satisfaction would be far too low, and customers would easily have the option of leaving for a better contract offer.

- **Customer Delight:** Customer delight is surprising a customer by exceeding his/her expectations and thus creating a positive emotional reaction. This emotional reaction leads to the word of mouth. Customer Delight directly affects sales and profitability of a company as it helps to distinguish the company and its products and services from the competition. In the past customer satisfaction has been seen as a key performance indicator. Customer satisfaction measures the extent to which the expectations of a customer are met (compared to expectations being exceeded). However, it has been discovered that mere customer satisfaction does not create brand loyalty nor does it encourage positive word of mouth.

  Customer Delight can be created by the product itself, by accompanied standard services and by interaction with people at the front line. The interaction is the greatest source of opportunities to create delight as it can be personalized and tailored to the specific needs and wishes of the customer. During contacts with touch points in the company, more than just customer service can be delivered. The person at the front line can surprise by showing a sincere personal interest in the customer, offer small attentions that might please or find a solution specific to particular needs. Those front-line employees are able to develop a relationship between the customer and the brand.

- **Customer Loyalty:** Customer loyalty is the key objective of customer relationship management and describes the loyalty which is established between a customer and companies, persons, products or brands. The individual market segments should be targeted in terms of developing customer loyalty. Customer loyalty is both an attitudinal and behavioural tendency to favour one brand over all others, whether due to satisfaction with the product or service, its convenience or performance, or simply familiarity and comfort with the brand. Customer loyalty encourages consumers to shop more consistently, spend a greater share of wallet, and feel positive about a shopping experience, helping attract consumers to familiar brands in the face of a competitive environment. There are many definitions of customer loyalty. Yet each of them fails to realize that loyalty runs hand-in-hand with emotions. Customer loyalty is the result of consistently positive emotional experience, physical attribute-based satisfaction and perceived value of an experience, which includes the product or services. Consider whom you yourself are loyal to. Surely you will answer family and friends. Why? Because of the emotional bond you have with them. Your family and friends can do things you may not like, but you stay loyal because of that bond. The same applies with customer loyalty. To prompt customer loyalty you must build an emotional bond with your customers.

To build customer loyalty, customer experience management blends the physical, emotional and value elements of an experience into one cohesive experience. Retaining customers is less expensive than acquiring new ones, and customer experience management is the most cost-effective way to drive customer satisfaction, customer retention and customer loyalty. Not only do loyal customers ensure sales, but they are also more likely to purchase ancillary, high-margin supplemental products and services. Loyal customers reduce costs associated with consumer education and marketing, especially when they become Net Promoters for your organization.

5.11 REASONS FOR ORGANISATIONS LOSING CUSTOMERS

- ✓ If the price of a brand appears very high, and the customer perceives a mismatch between the price and the customer value, he would switch to a competitor’s brand.
- ✓ If a new product, which is advanced in technology, offering better features and performance enters the market, customers may opt for that.
- ✓ If the customers are dissatisfied with the service- pre sales, during sales, and after sales- they would switch over to a substitute.
- ✓ Customers can also move away towards the competitors’ products attracted by augmented benefits offered by them.
- ✓ Personal reasons can also make a customer to switch from one brand to another. Like shift of residence, change in preferences, family and friends influence, anger, emotions and sentimental reasons.

CHECK YOUR PROGRESS

XV. We live in a world cluttered with advertisements; it can be difficult for companies to break through all of it and get to target market customers. (True/False)

XVI. …………. can be framed to seem positive or negative and this will affect how customers assign value.

XVII. …………………. focuses totally on customers and the organisation's entire gamut of functions related to value creation and value delivery.

a) BRP c) CRM

b) ERP d) MIS

XVIII. CRM is beneficial to the customers in the form of improvement in service quality by the marketers, more attention, personalized service, less risk, more commitment and increased value for money. (True/False)

XIX. …………….. is surprising a customer by exceeding his/her expectations and thus creating a positive emotional reaction.

a) Customer delight c) Customer Satisfaction

b) Customer Value d) All three

XX. The individual market segments should be targeted in terms of developing ……..

a) Customer Value c) Customer Delight

b) Customer Satisfaction d) Customer loyalty

5.12 LET US SUM UP

Electronic marketing is normally associated with Internet marketing. Internet marketing captures data which feeds into the firm's database; the database is used to generate profiles and lists, which enable the firm to have effective direct marketing campaigns; and two of the media for direct marketing are the Internet using e-mails and CD-ROMs with hyperlinks to the Internet. Underlying electronic business has two phenomena: digitalization and connectivity. Consumer behaviour is described as the action and reaction of the consumer while searching, evaluating, buying and consuming the product to satisfy their needs. Consumer behaviour focuses on how individuals make decisions to spend their available resources (time, money and effort) on products and services for consumption and satisfaction purpose. As individuals we all differ in many ways, in likes, dislikes, attitudes, cultural background, income level, education, occupation, family background etc. Despite such differences we are all consumers. If a marketer can identify online consumer's buying behaviour, he or she will be in a better position to target products and services at them. Online Buyer behaviour is focused upon the needs of individuals, groups and organisations. In the mid-twentieth century, mass production techniques and mass marketing changed the competitive landscape by increasing product availability for consumers. However, the purchasing process that allowed the shopkeeper and customer to spend quality time getting to know each other was also fundamentally changed. Customers lost their uniqueness, as they became an "account number" and shopkeepers lost track of their customers' individual needs as the market became full of product and service options. Public Relations in its true sense are about human connections and the art of mastering human connections at a deep level. In the early days of PR, it was about relationships with not just the press but communities in various forms; the only difference was that these audiences were not online. Customer relationship management (CRM) focuses totally on customers and the organisation's entire gamut of functions related to value creation and value delivery. It is concerned with developing customer loyalty and brand loyalty to the highest possible level, to maintain long-term customer relationship.

5.13 KEY WORDS

DIGITALIZATION — It means converting text, data, sounds, and image into a stream of bits that can be dispatched at incredible speeds from one location to another.

CONSUMER BEHAVIOUR — It is described as the action and reaction of the consumer while searching, evaluating, buying and consuming the product to satisfy their needs.

PERSONALITY — It is defined as the thoughts, emotions, intentions and behaviour that people express as they move through their environment.

IMPULSIVENESS — Purchases are made without much thought beforehand.

SOPHISTICATION — Desiring products that are glamorous and prestigious.

MOTIVATION — It is an internal state that drives consumers to satisfy needs.

PROPAGANDA — It is the manipulation of symbols to transmit accepted attitudes and skills.

CUSTOMER RELATIONSHIP MANAGEMENT — It is defined as a process of acquiring customers by understanding their needs, retaining customers by fulfilling the needs, and attracting new customers through customer specific strategic marketing approaches.

CUSTOMER VALUE — It is the benefit that a customer will get from a product or service in comparison with its cost.

CUSTOMER — It is a measure of how products and services supplied by the

SATISFACTION marketers meet or surpass customer expectation.

CUSTOMER DELIGHT It is surprising a customer by exceeding his/her expectations and thus creating a positive emotional reaction.

5.14 ANSWERS TO CHECK THE PROGRESS

I. True
II. False
III. True
IV. Sophistication
V. Frugality
VI. Anxiety
VII. Motivation
VIII. True
IX. Public Relations
X. Propaganda
XI. True
XII. Mutual understanding
XIII. False
XIV. Lobbying
XV. True
XVI. Messages
XVII. CRM
XVIII. True
XIX. Customer delight
XX. Customer loyalty

5.15 TERMINAL QUESTIONS

- What is consumer behaviour and how the technology has influenced the consumer behaviour?
- How the consumer behaviour is under the great influence of internal factors. How these factors influence the consumer behaviour?
- What is role of Public relationship for consumer behaviour?
- How public relation is different from Marketing? What are the benefits of Online PR?
- Highlight the important dimensions of Public relations and why they are important for the marketing managers?
- What are the characteristics of the environment and how they influence the consumer behaviour?
- What is CRM and how they are beneficial for the marketers and customers?
- Discuss the essence of E-CRM and why the business organisations are losing customers these days?

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-6 PURCHASE DECISION PROCESS, POST PURCHASE BEHAVIOR AND MANAGEMENT

STRUCTURE

6.0 OBJECTIVES

After studying the Unit, you would be able to

- Identify the nature of consumer during buying process
- Understand the Behaviour of consumer while making purchase
- Talk about the different stages of buying process
- Elaborate the classification of consumer Behaviour
- Indicate the behavior of consumer after purchase
- Understand the post purchase dissonance

6.1 INTRODUCTION

Marketers frame their marketing strategies on the bases of assumptions how customers are going to behave in a given set of circumstances. It is assumed that customer will buy more if prices are charged low and sales automatically come down if the marketers increase the price of the product.

Consumer behaviour is a branch which deals with the whole process through which consumers have to go while purchasing and using the products to satisfy their needs. In the market, we can see the consumer are behaving differently to select and buy the product among various product to satisfy their needs.

Marketers assume that by understanding what are the factors that cause the consumers to buy particular brand of product, they will be able to identify, what product is needed in the market, which one is obsolete and how to present the products to consumers. For example, if a customer is selecting pain reliever, they may like the idea is one pain reliever is cheaper than another, but what they really want is fast pain relief, and will probably pay more if they think the more expensive brand can do that more effectively. Marketers need to understand the principal motivation behind each type of product to correctly target potential customers.

Nature of Consumer Behaviour:

- The consumer behaviour is influenced from number of factors which prevail in the surrounding environment.
- All the consumers do not behave in a same manner. Consumers behave differently due to different elements i.e., age of consumer, income level, education etc.
- Consumer behaviour is not static; it undergoes a change over a period of time depending upon the need s and wants of the consumers.
- The consumer behaviour varies across state and countries. The needs of urban's consumers different from the rural consumers.
- Consumers behave differently for the different products. They do not have same behaviour for the different products.
- Consumer behaviour is not influenced from the status of consumers infect it reflect it.

6.2 BUYER DECISION PROCESS

The consumer has to go through number of different stages in order to buy any product and services. Nobody is planning to buy anything until they don't recognise the need, once need recognised information is required to take the decision and at last decision would be taken. Let's have a look at each stage and offer a quick explanation of what it is all about:

- **Recognition of Problem/ Need:** Stage one is the recognition of the particular problem or need and here the buyer has a need to satisfy or a problem that needs solving. Problem recognition is in fact the beginning of the buying process. It is a matter of perception, we realize what we should have and what we have at present. A perfume bottle of Charlie may be purchased when one sees it in the window of a shop. Problem recognition is generally a slow process but can occur fast when purchases are made online impulsively.

  Digital Marketing efforts facilitate the problem recognition stage. Ads provoke us to buy many things. Similarly, an online product display also acts as an incentive to buy. Online promotional strategies make us realize what our unfulfilled needs and wants are. Psychological factors do influence Problem recognition stage.

- **Search for the Information:** Stage two is where we begin to online search for information about the product or service. Buyers here begin to look around to find out what's out there in terms of choice and they start to work out what might be the best product or service for solving the problem or satisfying any need. More often than not, there is little knowledge available about alternatives available. The search is mostly directed towards the products that are consistent with our needs. For example, a house wife buying a washing machine might start visiting the online market websites selling it and might start online discussing the need with her friends. She is interested in knowing which brands are on offer and their features.

  The amount and type of information collected is related to the product in question and the consumer's personality. To buy consumer durables like fridges and AC's we need a great deal of information from different sources. To buy fast moving consumer goods, we need limited information. The beauty of brand marketing is that it makes the consumer loyal to the brands. These consumers then continue to buy the same brands and sometime do not seek any information at all.

  Information seeking starts with cognitive internal search; recalling information stored in memory. This may lead to further stages of buying decision process. Alternatively, the consumer may start external search, seeking information from sources other than memory. The major external sources are peers, friends, colleagues, relatives whom we trust. In addition, we get information from different marketing communications, media, distribution channels and consumer's own experience.

- **Evaluation of Available Alternatives:** Stage three sees the evaluation of the available alternatives whereby the buyer decides upon a set of criteria by which to assess each alternative. In online market its easier for the consumers to make comparisons among different products and brands. When the consumer seeks information, he realizes the alternative choices available to him and gets the background against which these choices can be judged. The brands which a consumer consider while making a purchase decision forms an evoked set, which is a small proportion of the total brands available. Each brand in the evoked set is evaluated against some chosen criteria. A consumer may want to look into the following criteria before buying a product online. These may be the Brand name, the price, the functions performed, the appearance and looks, reputation of the company, warranty specifications, technical specifications and also the after sales service available. Each of the criteria is assigned some weightage, which may differ from consumer to consumer. After this it is all a matter of perception. Promotion, especially advertising, provides information to the consumers, enabling him to evaluate the different alternatives.

- **Make Choice:** We buy or select a product/service/supplier at stage four. Individuals or teams of buyers make the final choice of what to buy and from whom to buy it. After the alternative choices are evaluated, the brands are ranked, and the top- ranking brand may be purchased. The ultimate buying decision may undergo a change, if the preferred brand is not available. In such a situation, the second, ranked brand may be bought. The ultimate buying occurs only when the consumer finds a suitable outlet where the brand is available, and price negotiations are complete.

- **Post Purchase Evaluation:** Interestingly the process does not stop at the point of purchase because there is a stage five called the post-purchase evaluation. A satisfied consumer stores the product information in his memory and uses it next time at the time of problem recognition stage. A dissatisfied consumer may go for another brand next time he is out to buy. He will seek additional information and will consider another set of brands. Thus, consumer decision making is an extensive process. Promotions play a vital role helping customers decide during the process. But the companies should keep in mind that they do not rub in the offers for it may work adversely for them.

Let's look at an example based upon buying a new smart cellphone. The first stage is likely to be that you have a need for communication or access to the internet, or problem because you cannot interact with friends using social media without it. The value added by products such as Android, iPhone or Windows phone and others should satisfy your need or solve your problem.

So, the second stage is where you speak to your friends and surf the internet looking at alternatives, which represent stage two or your information search. As a buyer you might visit a local cellphone store and speak to the sales staff to help you complete stage three, i.e. your evaluation of alternatives. Stage four is the selection of product and you go and make your final decision and buy your Smartphone from a local store or using an e-commerce website. Stage five involves your post-purchase evaluation whereby you use the phone and have a positive, negative or mediocre experience of the product. If it doesn't satisfy your needs you take action and more importantly you will tell others of your problems. If you are pleased with the product, you will tell your friends and this will influence stage two (their information search) when they decide to buy a cell phone. Remember that organisations and businesses also go through this process and that teams of individuals contribute to the decision-making process. This is called a Decision-Making Unit (DMU).

CHECK YOUR PROGRESS

I. The consumer behaviour is influenced from number of factors which prevail in the surrounding environment. (True/False)

II. Stage two in ………………… is where we begin to online search for information about the product or service.

a) Product life cycle
b) Buying decision process
c) New product development
d) None of these

III. Consumers do not behave differently for the different products. (True/False)

IV. Information seeking does not starts with cognitive internal search; recalling information stored in memory. (True/False)

V. Consumer behaviour is not influenced from the status of consumers infect it reflect it. (True/False)

VI. For FMCG the purchase decision process tends to be ………….., and for habitual purchase behaviour or repeat purchases the decision process is short-circuited.

VII. Digital Marketing efforts facilitate the problem recognition stage. (True/False)

VIII. A satisfied consumer stores the product information in his memory and uses it next time at the time of ……………. stage.

6.3 CLASSIFICATION OF BUYING BEHAVIOR

In the human being wants are unlimited but the resources to satisfy them are limited. Hence consumers think rationally before taking the purchase decision. Consumer's decision varies with the type of product purchased to satisfy needs. There are differences in the buying behavior of consumers when he is going to buy toothpaste, cricket bat, a laptop and new car. Complex and purchases involved high expenditure demands more participation and deliberation of buyer. Consumers need more information about the products which are expensive for them. There are four types of buying behavior of the consumers on the basis of buyer involvement while buying the products.

	High involvement	***Low involvement***
Significant differences between brands	Complex buying behavior	Variety seeking behavior
Few differences between brands	Dissonance buying behavior	Habitual buying behavior

High involvement means the consumers is highly involved while purchasing the product. The consumers involved high when they are buying the expensive or luxuries products. For example, while buying the diamond a consumer is highly involved. On another side low involvement means, when consumers are not highly involved in buying the product. Low involvement is shown by the consumers for low priced products like buying the toothpaste. Significant differences between brands mean when there are significant differences between brands.

- **Complex Buying Behavior:** Consumers go through the complex buying behavior, when they are highly involved in the buying the products and they can easily identify the difference between the different brands. Consumers would be highly involved when the product is expensive and bought infrequently. Consumers do not know much about the products and need information to learn about product more. In this case the buyers go through a cognitive learning process. So, in this case consumers need to get proper information about the product’s attributes and marketers must assist the consumers by providing the detailed information regarding product features. For example, consumer would be highly involved in the purchase and has the information or awareness about significant difference between brands.

- **Variety seeking buying behavior:** Some time buying situations demand low involvements of the consumers but significant difference is there in brands. Consumers are brand switcher in this kind of buying behavior. Brand switching occurs because of seeking variety not due to dissatisfaction from the consumption of last product. For examples ladies most of the time use different brand of washing detergent just for variety. Therefore, it would be better for the marketers to retain the consumers to offer them incentives along with the core product i.e., discounts, free sample, gift coupons etc.

- **Buying behavior reducing Dissonance:** in some case while buying the product consumers are highly involved but it is difficult for them to distinguish the brands. Consumers are highly involved because the product is expensive and infrequently purchased. The buyers will visit around to get information what is available but will buy quickly because difference among the different brand is not easy. For example, consumers while buying the floor tiles buy them quickly as in brands of tiles, there are few differences.

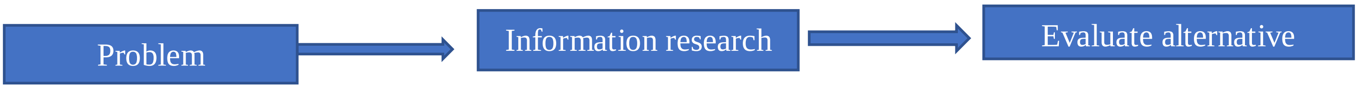
- **Habitual buying behavior:** Most of the products are purchases by the consumers which demand low involvement and significant difference between brands are absent. In this case there is low involvement of the consumers and there are few differences between the brands as consumers do not need extensive information about the products. The consumers buy product on routine bases for example toothpaste, soaps, etc. Consumers do not have any kind of attitude toward ant brand but they prefer to buy familiar brand. Hence marketers should use the promotional techniques to get the consumers familiar with the brand or product.

6.4 POST PURCHASE BEHVIOUR CONSUMER BUYING PROCESS

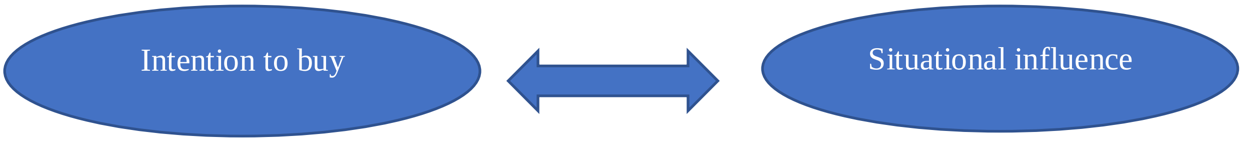
Post-purchase conduct is the response of the customer and the individual, it gives a thought regarding the products' preferences, inclinations and perspectives and satisfaction about the product. The stage is concerned with the behavior of the consumer after consumption of product; the behavior may be positive or negative. A satisfied consumer will go for repeat purchase or recommend the product to other. On other side dissatisfied consumer will feel restless or start search for another product to satisfy unsatisfied need. He will do the negative marketing of unsatisfied product in the society.

The customer while settling on a choice goes through three stages:

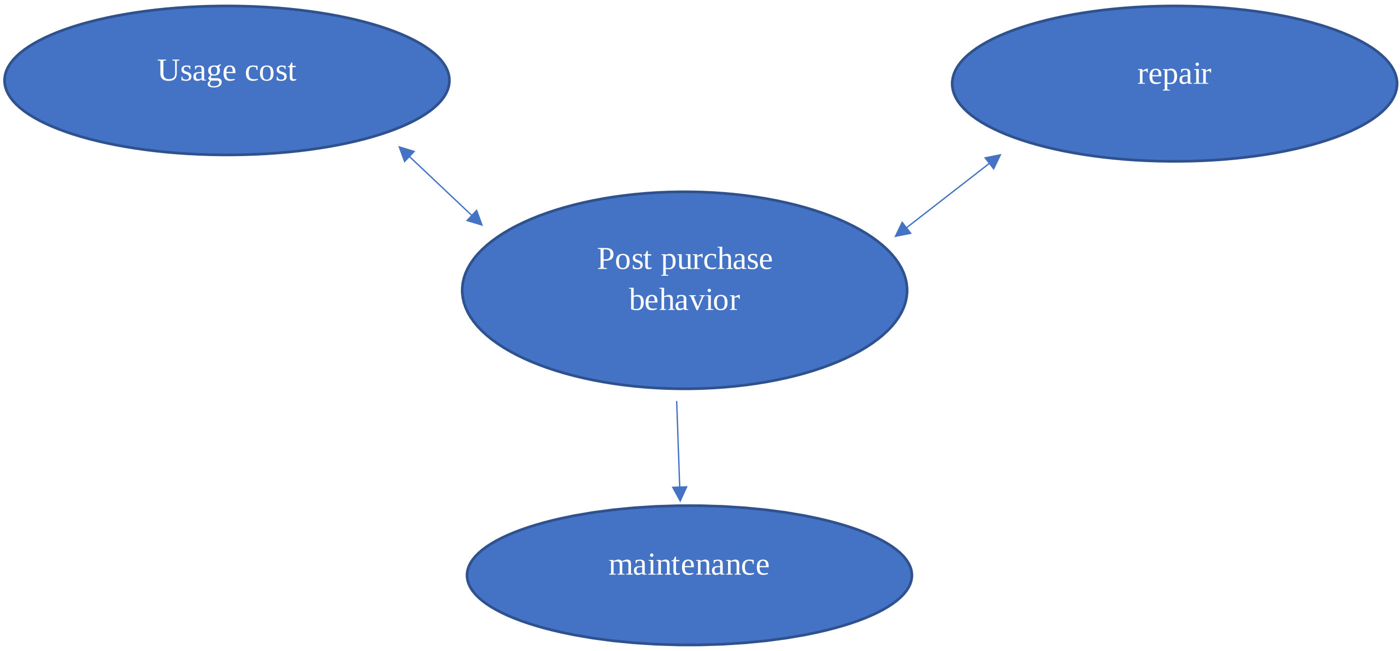
1. Pre purchase behavior activity which comprises us



2. Purchase process



3. Post purchase behavior



4. Feedback to marketer

Improves the product

Change marketing mix

Give more facility to customer

6.5 POST-PURCHASE DISSONANCE

A customer's reaction after he was involved in the process of purchasing some high involvement product, usually a very expensive one, like furniture or a house or a car maybe. This purchase creates doubt and anxiety in the mind of the consumer, especially when the degree of commitment is high. This dissonance often leads to consumption guilt, which is a negative emotion or feeling aroused by the usage of that expensive product.

- **Consumers can reduce dissonance by**
  - ➢ By focusing on more supportive that outweigh the dissonant belief
  - ➢ Reduce the importance of conflicting belief
- **Marketers can reduce dissonance by**
  - ➢ By increasing the desirability of the brand
  - ➢ By decreasing the desirability of the opposing brand
  - ➢ By reversing the purchase decision

CHECK YOUR PROGRESS

IX. …………… is shown by the consumers for low priced products like buying the toothpaste.

X. Consumers are brand switcher in …………….. kind of buying behavior.

a) Variety seeking buying behavior
b) Buying behavior reducing Dissonance
c) Habitual buying behavior
d) Complex Buying Behavior

XI. Consumers would be …………….. when the product is expensive and bought infrequently.

XII. Purchase creates doubt and anxiety in the mind of the consumer, especially when the degree of commitment is high. (True/False)

XIII. Consumers would be highly involved when the product is expensive and bought infrequently under ……………..

a) Variety seeking buying behavior
b) Buying behavior reducing Dissonance
c) Habitual buying behavior
d) Complex Buying Behavior

6.6 INTERNET MARKETING

The Internet is **not** synonymous with world wide web. The Internet is a massive network of networks, a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. The World Wide Web, or simply the Web, is a way of accessing information over the medium of the Internet. It is an information-sharing model that is built on top of the Internet.

6.7 INBOUND VERSUS OUTBOUND MARKETING

Outbound marketing was typically the traditional approach to market business and this was known as an interruption-based marketing. This is where we broadcast our message through advertising and other mediums and try to grab the attention of the user. When consumers are bombarded with on average 3,000 messages a day, we can understand why this form of marketing is becoming increasingly difficult. It certainly still works in some cases, but there are now other ways that can be more effective.

Inbound marketing is where marketers provide something of value that attracts a customer to come to shop. When marketers attract that customer, they try to build a relationship. After they build the relationship, they develop trust and this then allows them to sell their product or service to customers. This form of marketing, although difficult to grasp is becoming increasingly effective.

A permission-based approach such as this can also produce much better results because:

- When marketers build a relationship within any client online or offline their chances of a sales increase significantly. When they go into a sales meeting and fail to build rapport with the potential customer it is difficult to get the sale. Online is now becoming very similar.
- It’s generally more cost effective. It does involve more of the time, but there is less expense.
- The performance / return is relatively easy to track so marketer can adjust to get better results

6.8 THE INBOUND MARKETING METHODOLOGY

- **Attract:** In inbound marketing we want attract the right customer to your website and to your brand with a quality, answers that questions they have. We well use blogging, social media, keyword and SEO strategy and even outbound advertising to do this.
- **Convert:** After attracting the strangers on websites, a company will ready to convert them into prospect collect their contact info. Emails give important information for an inbound marketer. This includes eBooks, whitepapers, case studies and tips sheets.
- **Close:** With the help of tools like lead scoring closed-looks reporting, lead nurturing and vast lead intelligence, paired with lead follow up best practices and sales tactics, marketers will then work to transform those leads into now business.
- **Delight:** Delight refers to deliver the right information to the right person at the right person at the right time. Delight tools are such as Smart content, Email marketing, conversations inbox, attribution reporting and marketing automation.

6.9 LET US SUM UP

Marketers frame their marketing strategies on the bases of assumptions how customers are going to behave in a given set of circumstances. Consumer behaviour is a branch which deals with the whole process through which consumers have to go while purchasing and using the products to satisfy their needs. Marketers assume that by understanding what are the factors that cause the consumers to buy particular brand of product, they will be able to identify, what product is needed in the market, which one is obsolete and how to present the products to consumers. The buyer decision process represents a number of stages that the online purchaser will go through before actually making the final online purchase decision. The consumer buying decision making process and the business/organisational buyer decision process are similar to each other. In the human being wants are unlimited but the resources to satisfy them are limited. Hence consumers think rationally before taking the purchase decision. Consumer’s decision varies with the type of product purchased to satisfy needs. There are differences in the buying behavior of consumers when he is going to buy toothpaste, cricket bat, a laptop and new car. Post-purchase conduct is the response of the customer and the individual, it gives a thought regarding the products’ preferences, inclinations and perspectives and satisfaction about the product. The stage is concerned with the behavior of the consumer after consumption of product; the behavior may be positive or negative. A satisfied consumer will go for repeat purchase or recommend the product to other. On other side dissatisfied consumer will feel restless or start search for another product to satisfy unsatisfied need.

6.10 KEY WORDS

CONSUMER BUYING BEHAVIOUR	It is a branch which deals with the whole process through which consumers have to go while purchasing.
BUYER DECISION PROCESS	It represents a number of stages that the online purchaser will go through before actually making the final online purchase decision.
PROBLEM RECOGNITION	It is the beginning of the buying process.
HIGH INVOLVEMENT	Means the consumers is highly involved while purchasing the product.
COMPLEX BUYING BEHAVIOR	When consumers are highly involved in the buying the products and they can easily identify the difference between the different brands.
HABITUAL BUYING BEHAVIOR	Purchases made by the consumers which demand low involvement and significant difference between brands are absent.
DISSONANCE	It often leads to consumption guilt, which is a negative emotion or feeling aroused by the usage of expensive product.

6.11 ANSWERS TO CHECK THE PROGRESS

I. True

II. Buying decision process

III. False

IV. False

V. True

VI. Shorter/quicker

VII. True

VIII. Problem recognition

IX. Low involvement

X. Variety seeking buying behavior

XI. Highly involved

XII. True

XIII. Complex Buying Behavior

6.12 TERMINAL QUESTIONS

- What is Consumer behaviour? Discuss the nature of consumer behaviour.
- Describe in detail the buying decision process by taking any product category.
- How the different stages of buying decision process are important for the marketers in strategy formulation?
- What is buying behaviour? What are the classifications for the buying behaviour, explain them with examples?
- What is post purchase buying behaviour and why such behaviour is important for the marketing managers?
- Explain in detail the post purchase dissonance.
- What is Internet Marketing and what are the strategies under internet marketing for consumer satisfaction?

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-7 DIGITAL MARKETING STRATEGY: DIGITAL VS NON- DIGITAL MARKETING STRATEGY

STRUCTURE

7.0 OBJECTIVES

After studying the Unit, you would be able to

- Frame the marketing strategies
- Understand the important dimensions for marketing strategies
- Compare the digital and non-digital marketing
- Describe the tools for digital marketing
- Elaborate the digital marketing environment
- Find out the different factors affecting the marketing strategies
- Evaluate the how micro and macro environment influence the marketing strategies

7.1 INTRODCTION

Marketing is a method of advertisement that is used by the companies to attract customers and making them aware about the products and services those are provided by the companies. Main goal of marketing is to reach the desired segment of customers that matches with the company’s products/services or we can say to reach the customers that needs or wants our company’s products/services. Reaching the desired segment of customers ensures the profitability to the business firm.

Marketing is a tool that is used by the companies to sell their products/services to the customers. Marketing includes the sales, pricing, public relations, packaging and distribution. It is a kind of process that ensures the needs and wants of customers and delivering the products efficiently and ensures the profitability to the firm. It is all about addressing the needs of customers; through advertising, selling and delivering products to customers. The whole process includes following steps:

1. Coordination of identification
2. Selection and development of a product
3. Determination of its price
4. Selection of a distribution channel to reach the customer’s place
5. Development and implementation of a promotional strategy.

Through traditional marketing organizations straightforwardly target and discover clients, while through digital marketing their goal is to have individuals discover them. Accomplishment of a digital marketing exceptionally relies upon the clients approaching the Internet and being comfortable with the channels used to target them, be those informal communities or sites.

7.2 MARKETING STRATEGIES

It is a long-term, forward looking approach of any business firm having a goal of achieving the competitive advantage by understanding the wants and needs of their customers. A marketing strategy covers the company’s key brand messaging, value proposition; it contains the target customer’s data demographics and other elements.

Marketing strategy covers “four Ps” of marketing:

PRODUCT

PRICE

PLACE

PROMOTION

- **NON-DIGITAL MARKETING:** It is termed as a type of marketing that is not conducted online. It includes print, broadcast, direct mail, phone and other outdoor advertisements like billboards. Newspapers and radio help the company to reach targeted audiences.

It is not only one of the oldest methods of marketing but it is also one of the most researched. Many marketers adopt this method because it is tried and true. Traditional marketing is playing an important part in reaching the local audiences. Physical Ads those can be kept for longer period of time.

- It includes greater cost because it has various platforms.
- The scope of traditional marketing is limited, as the target audience is very specific.
- The campaigns tend to be rigid cause of no room for adjustments.
- It allows Uni-dimensional communication with having very little customer interaction.
- In non-digital marketing the employees can be available to the customers only during the working hours.
- A Non-digitalised advertisement takes time to spread among the audience.
- There is no any specific tool available to measure the performance of a campaign.
- Only general audience is targeted through non-digital marketing.
- Door to marketing is possible in case of traditional marketing the campaign can directly interact with the customers.
- This method requires huge investment as compared to the digital marketing.
- The reach to customers is very limited in traditional marketing.
- Non-digital marketing cannot target the audience groups effectively.
- Proper availability of budget is required to execute the campaign.
- Personalized advertisements are not possible.
- The conversion rates are low.
- Auditing is difficult.

- **DIGITAL MARKETING**: It refers to a marketing that is conducted online by the company. It includes email marketing, social media ads. It is becoming popular because the world is now becoming digitalised. It is more cost-efficient than the traditional marketing.

More audience can be targeted at once through digital marketing. Many customers prefer to purchase the products online that's why it becomes the most preferred platform for buying the products.

- It allows the multidimensional communication and also facilitates the active communication from consumers.
- The strategies of digital marketing have much room for facilitating the adjustments.
- Cost of promoting the campaign is very low, as predominant platform is internet.
- It has a large scope, all the information is made available to the customers as a whole.
- In digital marketing the company can target millions of people at once because all the people are connected with technology.
- The company/organization can provide 24/7 availability to the customers.
- The advertisement takes only few seconds to reach to your target audience.
- There are many tools available online to monitor the performance of marketing campaign.
- In digital marketing scheduling is possible making it possible for the employee to post even at non-working hours.
- Niche audience can be target with digital media.
- Two way communications is possible with the customers.
- It is more cost efficient the investment is not much.
- It can customize the advertisements according to the user's interests.
- This method is very cost effective and in some scenarios it is totally free.
- The conversion rates are high.
- Audit is possible at any time.

7.3 NON-DIGITAL MARKETING STRATEGIES

It relies on offline strategies such as direct mail, direct sales, word-of-mouth communication, and print advertisements in magazines, newspapers, billboards, radio and television. The main aim of the company behind the non-digital marketing is to create brand awareness. In the recent survey, only 4% of the respondents rated the leads generated from radio, print and TV ads as of high quality.

- **Signage**: It comprises of digital photos those are framed and lighted in the dark. Now the companies started putting signage on mobile taxis irrespective on the boring walls. This method helps to reach the wider segment of audience than ever before.
- **Billboards**: Using printed or the hand painted canvas images are used by many organizations. Billboards have come a long way from the highways to hallways. Non-digital marketing uses more images and less text. A picture express thousand words, so it is best way for creating brand awareness. Most of the business firms are more familiar with this system of marketing. Billboards give wide reach across the social classes and the geographical boundaries.
- **Direct mail**: Business firms and many other organizations use the direct mail system for conveying the people about their services. Mailing is a personalizing, creative method for targeting a particular segment of audience. Direct mail has a great power to influence the readers; it is also easier to understand. Faster feedback is received in case of the direct mails, more purchases are made and it is an greater ease for retrieval making of the good old fashioned direct mail the real deal in new age of marketing.
- **Flyers and brochures**: Flyers and brochures are come in handy in places such as: streets, malls, brick and mortal establishments. Flyers and brochures are used for special discounts and offers. Business cards introduce a business to the client without having any personal interaction. Non-digital marketing is more connective and interactive way of dealing with the customers in face-to-face interactions.
- **Face to face interaction**: This method has been used for around ages. Face to face interaction is more attractive way to deal with the customers and conveying the details regarding their business products. It helps the customers also for understanding all the details regarding the products they want to buy in their preferred language. Customers also provide the feedback to the company regarding the product experience and the kinds of modifications they want in the products.
- **Direct sales:** It involves the marketing and sales of products/services directly to the customers, but the location may be differing. These kinds of sales are done at customer’s homes or at their jobs. Direct sales are also done over the phone.
- **Tradeshows**: Tradeshows are continued to be a popular mode of marketing channel for many organizations and the business firms. In business to business space the

attendees are 34% more willing/likely to purchase the product than the persons who hear about the products through other channels of marketing.

- **Print marketing**: It creates awareness of the product and the service through advertisements in a magazines, pages, billboards and newspapers, etc. it is a targeted or broad way of marketing. But this mode of marketing is expensive as compared to the digital marketing.
- **Referral**: It is also known as word-of-mouth marketing. It leverages the company's existing customer's to advocate for company's business. Many business organizations provide referral payments to the customers who bring in other customers in the company.
- **Broadcast:** Many of the small business firms rely on the television and the radio advertisements to reach the target audience.
- **On-site activities**: Most of the organizations still invest in on-site marketing activities and events. It is a great way to make a lasting relationship/connection with the target market. On-site activities provide the opportunity to capture great content like photographs and videos.
- **Handwritten cards**: Sending the handwritten cards to the clients on some special occasions like birthdays, graduations and other special life events makes the clients feel special and this kind of special treatment adds a personal touch.
- **Speaking engagements**: Speaking engagements can be the great way for getting directly in front of the target audience and it helps to create brand awareness. It can be an effective way of marketing tool because speaking engagements helps to build face-to-face connection brands with the consumers.

CHECK YOUR PROGRESS

I. Reaching the desired segment of customers ensures the ……………. to the business firm.

II. A marketing strategy covers the company's key brand messaging, value proposition; it contains the target customer's data demographics and other elements. (True/False)

III. …………. is termed as a type of marketing that is not conducted online.

IV. ……………… is more cost-efficient than the traditional marketing.

V. ……………….. interaction is more attractive way to deal with the customers and conveying the details regarding their business products.

a) Direct Mail b) Display advertising

c) Face to face marketing d) Flyer and poster

VI. Flyers and brochures are not used for special discounts and offers. (True/False)

VII. The technique, business firms rely on the television and the radio advertisements to reach the target audience is known as ………………

a) Broadcast c) Display advertising

b) Direct Mail d) Face to face marketing

VIII. Speaking engagements can be the great way for getting directly in front of the target audience and it helps to create brand awareness. (True/False)

7.4 DIGITAL MARKETING STRATEGIES

There are numerous digital marketing strategies those are used by the business firms for advertising their products and services. These are explained further:

- **Email marketing:** It is one of the earliest ways of digital marketing. Email marketing is composed of sending the commercial message regarding the company's product to a group of targeted customers using email and delivering the personalized emails and the targeted messages at the right time. With the help of email marketing software the company can maintain separate email lists on the bases of customer's likes and dislikes.
- **Social media marketing**: It is a form of marketing that utilizes the networking sites as a tool of marketing. The main goal of social media marketing is to produce that kind of content that the customers share with their social network groups. Many customers try to finds their favourite brands on social media so proper social media strategy is now become necessary for every organization to reach to their targeted audience. With the increased popularity of Google, face book, twitter and YouTube social media marketing became more popular. Posting the right kind of content is necessary so that the people will share your company's content, it helps to reach to the new customers. These kinds of sites also are used as an effective form of communication to engage with your customers.
- **Search engine optimization**: It is an essential strategy in digital marketing. It is the process in which the website's visibility is affected or making a web page visible in a search engine's unpaid results, it is often referred to as "natural results". This process is done by increasing the volume of keywords that a site ranks for.

Search engine optimization targets the various kinds of search such as video search, image search etc. it also covers the factors like how the search engine works, about what the people search for and the kind of keywords typed into search engines. The kind of search engines are preferred by the company's targeted customers also finds out by the search engine optimization.

- **Mobile advertising:** It is also one of the biggest growth areas in the digital marketing. The advertising is done via mobile phones. The mobile phones which are used for advertising are: smart phones, wireless phones or tablet computers.
- **Search engine marketing**: It is a type of internet marketing uses the search engines for advertising the company's website or business to customers using internet. The more traffic to company's website is generated via advertising or paid links. It includes all the paid listings, search engine optimization and other related services that will increase traffic to the company's web site.
- **Online advertising:** It is also known as internet advertising or web advertising, is a form of marketing and advertising that uses the internet services to deliver the promotional marketing messages to the consumers. Despite of many benefits of online marketing there are some reasons the consumers finds this mode disruptive and blocking the ads. Online advertising includes: Social media marketing, search engine marketing and Mobile marketing.
- **Affiliate marketing:** Affiliate marketing occurs when the business firm organize third parties for bring in customers for them. The payment is given to the third parties on the basis of sales generated through their advertising. Affiliate generates he traffic to the company through their affiliate networks; once the purchase is made by the visitors they receive the payment. The desired actions can be:
    - ➢ Email submission
    - ➢ Phone call
    - ➢ Filling out an online registration form.
- **Display advertising:** It is done through advertising on websites. The main purpose behind display advertising is to deliver brand messages to site visitors and deliver general advertisements. Display advertising includes different formats and contains many items such as:
    - ➢ Flash
    - ➢ Images

    - ➢ Text
    - ➢ Video
    - ➢ Audio.

- **Adware:** It is software that once installed it automatically displays advertisements on the user's machine. All the ads may appear in the software itself, and integrated into the web pages visited by the user, or in pop-ups.

7.5 DIFFERENCE BETWEEN TRADITIONAL MARKETING AND DIGITAL MARKETING

The key distinction between traditional and digital marketing is how the marketing message is communicated to an audience. Here are some quick comparisons of digital and traditional marketing.

- **Cost:** Traditional Marketing has recurring costs that can add up quickly and result in a poor return on investment. An advertisement in the local newspaper or on the radio or television would only be worthwhile if it hits the target audience on the day it is transmitted or written. What are the odds of that happening?

  Any more publicity will require a re-run of the print or broadcast campaign, which will necessitate additional funding. Compare this to content marketing, which build once on website and can use for months or even years without costing anything extra. Digital marketing has the potential to be less costly.
- **Marketing Approach:** Since the marketer must build a personal relationship with the audience, traditional marketing requires a more personal approach. There is no need for a face-to-face partnership in digital marketing since all meetings take place online. However, in order to boost its image in the minds of the audience or customers, the marketer must maintain a considerable amount of online presence.
- **Target Audience:** Every marketing campaign has a specific target audience in mind. It is easy to reach out to the local audience or customers in the traditional market. Marketers need to figure out who target market is and then position marketing messages (via a channel) where they can easily see, hear, or communicate. It caters to a wide range of consumers and audiences. For example, Using newspaper advertisements, banners, and radio or television advertisements.

Digital marketing has a broader scope. To enter a global audience, digital marketing is made possible across the internet. It employs organic and paid search engine and social media material, as well as email marketing, video marketing, and influencer marketing. Single piece of content in digital marketing has a particular target market.

- **Customer interaction and participation:** Since traditional marketing mediums are not versatile enough to enable consumer engagement, traditional marketing provides very little interaction with the viewer. The viewer is left to see, hear, or interpret the marketing message without being able to quickly provide feedback.

  Engagement and interaction are made easier with digital marketing. Customers can easily provide feedback in the form of product reviews, buyer experiences, testimonies, and other similar products. This enables the company to quickly gather consumer reviews and implement it into their analytics.
- **Ease of measuring results:** Traditional marketing makes it difficult to track outcomes. Marketer never know how far ads can go when send fliers, give away brochures, or advertise in newspapers or magazines. Marketers can easily monitor results with digital marketing. Instead of going blind-eyed, they get a clear-sighted marketing strategy. To grasp where they are going, they have access to a lot of details about their clients and prospects. For example, they shall know how many people visit website and how many people watched video.

Below is a table of distinguish between Non-Digital Marketing and Digital Marketing

NON-DIGITAL MARKETING	**DIGITAL MARKETING**
In this marketer cannot target audience group effectively.	In digital marketing can target audience very effectively.
Personalized advertisement is impossible.	Can customize the advertisement as per the interest of audience.
It can take long duration to implement campaign.	In this campaign can implement very fast and easily.
Huge budget is required to implement a campaign.	It is cost effective, in some scenario it is free of cost.
Can generate positive ROI based on the product.	Very high ROI compared to advertising cost.

Static ads mostly non-engaging or non-interactive.	Easily create engaging and interactive advertisements.
Measuring campaign results are difficult.	Can easily measure the campaign metrics and data.

CHECK YOUR PROGRESS

IX. The main goal of ……………….. is to produce that kind of content that the customers share with their social network groups.

a) Display advertising
b) Email marketing
c) Social Media Marketing
d) Search Engine marketing

X. …………… is the process in which the website's visibility is affected or making a web page visible in a search engine's unpaid results.

XI. Affiliate marketing occurs when the business firm organize third parties for bring in customers for them. (True/False)

XII. The main purpose behind display advertising is to deliver brand messages to site visitors and deliver general advertisements. (True/False)

XIII. ……………….. is software that once installed it automatically displays advertisements on the user's machine.

7.6 DIGITAL MARKETING ENVIRONMENT

It refers that all the surrounding of the business which affects the Digital Marketing internally and externally. Study of these factors is called Digital Marketing environment analysis, as the study of these factors or surroundings is very beneficial for developing marketing strategies. Environment which surrounds the Digital Marketing is flexible, unpredictable in nature and complex. Due to flexible in nature and cannot stands still, it gives number of new opportunities to the business, therefore it is very important to frame new strategies according to the changes in the environment.

7.7 FACTORS AFFECTING DIGITAL MARKETING

INTERNAL FACTORS: Digital markets exist in the environment where there are lot of changes are going to take place and all such changes directly affect to the Digital Marketing environment. The internal factors include:

- **Target Market:** The first factor in the online market Environment is target market. It is foremost factor in business. Before deciding for a digital marketing, marketer should firstly analyse or determine how the customers through use of digital marketing get the information about services and products. Marketers always post or update content which is suitable and interesting for the target market.
- **Budget:** In every organisation cost is a big factor in every environment. In case there is shortage of funds then business should avoid the major channels for Digital market. Businesses where there have huge budgets, they would have more options like pay per click (PPC), setting new websites, and short message (SMS) services. Digital marketing is cheaper than traditional method of marketing but it involves much cost.
- **Talent:** Talent is the most appropriate factor in the Digital Marketing environment. Without this Digital Marketing cannot establish as talent come from experiences and the knowledge. If there is no skilled talent in the environment than everything is out of the place. It includes to know the; Alternatives or options, Implementing ideas and Measure that ideas.
- **Time factor:** Online marketing takes time as lot of time required for planning, implementing and monitoring etc. Monitoring on the strategies of digital marketing is done all the time or constant monitoring is required. In small businesses this factors hardly influence the strategies.
- **Channels:** In Digital Marketing marketers need to use the various channels to approach the target markets**.** Manager should evaluate the all these channel for their cost effectiveness.

7.8 MICRO (INTERNAL) ENVIRONMENTAL FACTORS

It also called operating environment as micro environment factors also directly influence the electronic market environment. The Factors are:

- **Customer: E**very business wants to have more sale of products, so they try to increase more and more customers. Customer is a major part of marketing model and It is very important for the company to know about their customers. There are lot of customer in the market therefore company should have knowledge about the customer that what type of media customer used for buying process.
- **Competitor:** Next factor in micro environment is competitors. Marketing Manager should aware about the strategies; plan of competitors in the market and with techniques of benchmarking overtakes the competitor activities and strategies.
- **Suppliers/distributors: T**hey also affect the online marketing environment. This factor includes all digital channels start from seller place to buyer place. With rapidly growth of the internet the digital connection between the business and consumer has grown as there is a less need of intermediator like wholesaler and retailer etc.
- **Organisation structure:** Digital market is all depends on the organisation, which includes behaviour of the employee, skills, technique, adoption of new methods in the organisation. In some organisations peoples (employees, managers) do not want to change in the organisation and uses old methods for marketing. To overcome this types of situation manager have to setup or modify all organisational structure in such way that it is beneficial for Digital Marketing.

7.9 MACRO (EXRERNAL) ENVIRONMEMTAL FACTORS

Macro environment mean general environment of business. Macro forces are Uncontrollable in comparison to the micro forces of environment. The growth and survival of business depend upon its adaptability to Macro environmental factors which include Political, Economic, Social, Technological, Legal and Environmental forces.

- **Political:** The political environment has many factors influencing the trading environment; they are shaped by the interplay of government agencies, public opinions and consumer pressure groups. Some examples of these factors include taxation, investment and management of business and public affairs. So marketers need to frame marketing strategies as per the rule of government.
- **Economic:** The United Kingdom has an open economy and ranks among the sixth largest in the world and it has best market place in the world. Issues such as

Inflation, GDP, employment levels, and interest rates need to be considered to ensure sustainability of a competitive and strong position in the online market.

- **Social:** The key factors that make up social forces and the implications for online marketing include; social communities based on demographic profile, social exclusion and cultural factors. The highest important shift in demographic trends is that over 50% of the population of the world lives in an urban setting. As well as emerging markets in Russia, India, Brazil and China represent the market growth.
- **Technological:** Technology is vital for competitive advantage and it is the major driver of globalization. Marketers need to understand digital and internet technology as making mistakes can lead to significant consequences.
- **Legal:** Laws are in place to provide a framework of control and regulations to enable individuals and businesses to go about their business in a legal manner. There are many legal and ethical considerations in the online trading environments and marketers have to understand and work with them. PLT will follow the Data protection and privacy law so their customers are protected online.

7.10 MARKET OPPORTUNITY ANALYSIS

Market opportunities analysis focus on the discovery of new opportunities and to evaluate the current position of company’s current technological, competitive and financial willingness to make use of that. It includes identifying the demands of customer which could not be met in past and identifying the target Markets and resources capacity to fulfil the requirements of the market’s needs.

There are several ways to find out the market opportunities:

- **Consumer segmentation:** for understanding the demand of market first thing is to do market segmentation and identification of consumer segment. There are basically two types of segmentations:

  Soft variables: lifestyles, purchasing motivations, attitude of customers towards products.

Hard variables: age of consumer, geographical area of consumer, education level, income level and gender of the consumer.

- **Purchase situation analysis:** This type of analysis can be used for the expansion of the business like uncover expansion opportunities. There are several question arises in this analysis.

  When the people buy the product?

  Is it be there when people's need it?

  Where people makeup purchase?

  How they would pay for them?

- **Direct competition analysis:** After analysing purchasing situation and demands in the market next step is to find or analyse the supply. Its first aspect is to study about the players exist in the market where marketer going to compete. Several questions are arises in this analysis

  What are the growing brands and product of our Industry and why?

  What is the competitive advantage we have over them?

- **Indirect Competition analysis:** Market opportunities are also found with the help of analysing substitute industries. This type of analysis is helpful to provide additional opportunities in the growth.
- **Analysis of other industries:** In this type of analysis company have not only objective is to continue operating business within the same industry but also to expand the business in other industry or expand the business model.
- **Complementary products and services analysis:** In this type of analysis marketers have to monitor performance of the other companies which are complementary to the first one.

CHECK YOUR PROGRESS

XIV. Online marketing takes time as lot of time required for planning, implementing and monitoring etc. (True/False)

XV. Marketing Manager should not aware about the strategies; plan of competitors in the market and with techniques of benchmarking overtakes the competitor activities and strategies. (True/False)

XVI. Digital market is all depends on the organisation, which includes behaviour of the employee, skills, technique, adoption of new methods in the organisation. (True/False)

XVII. The growth and survival of business depend upon its adaptability to …………….. factors.

7.11 LET US SUM UP

Digital Marketing draws heavily on Internet communications to co-ordinate many marketing activities, such as market research, New product development, market segmentation, positioning, product distribution, customer services, promotion, customer feedback, etc. These internet or electronic based activities should, in turn, be integrated with the overall marketing strategy to support the corporate objectives of the E-commerce. Implementing Digital Marketing strategy, requires an adequate understanding of changes in the way customers think and act on the World Wide Web. In other words, in a dynamic business environment, Digital Marketing is capable of cutting through the clutter to provide a precious opportunity for businesses to improve accessibility to the world and thrive. Marketers need to deliver product and services as per the requirement of customers. Digital Marketing Environment refers that all the surrounding of the business which affects the Digital Marketing internally and externally. Study of these factors is called Digital Marketing environment analysis, as the study of these factors or surroundings is very beneficial for developing marketing strategies.

7.12 KEY WORDS

MARKETING	It is a tool that is used by the companies to sell their products/services to the customers.
NON-DIGITAL MARKETING	It is termed as a type of marketing that is not conducted online
DIGITAL MARKETING	It refers to a marketing that is conducted online by the company.
SIGNAGE	It comprises of digital photos those are framed and lighted in the dark.
REFERRAL	It is known as word-of-mouth marketing.
EMAIL MARKETING	It is composed of sending the commercial message regarding the company’s product to a group of targeted customers
SOCIAL MEDIA	It is a form of marketing that utilizes the networking sites as a tool of

MARKETING	marketing
DISPLAY ADVERTISING	It is the advertising done on websites.
MACRO ENVIRONMENT	Macro forces are Uncontrollable in comparison to the micro forces of environment.
PURCHASE SITUATION ANALYSIS	The analysis can be used for the expansion of the business like uncover expansion opportunities.

7.13 ANSWERS TO CHECK THE PROGRESS

I. Profitability
II. True
III. Non-digital marketing
IV. Digital Marketing
V. Face to face marketing
VI. False
VII. Broadcast
VIII. True
IX. Social Media Marketing
X. SEO
XI. True
XII. True
XIII. Adware
XIV. True
XV. False
XVI. True
XVII. Macro environmental

7.14 TERMINAL QUESTIONS

- What is the requirement of Digital Marketing Environment analysis? Discuss the Customer Centric Online Marketing.
- What is Off line marketing strategies? Discuss in detail the various activities taking place in this.
- What is on-line marketing strategy? Discuss in detail the various activities taking place in digital marketing.
- Describe in details the non-digital marketing strategies.
- What are the different digital marketing strategies and how they are useful for the marketers?
- What is traditional marketing and how it is different from the digital marketing?
- What is digital marketing environment? Discuss in details the different factors of macro environment and how they are affecting the marketing?

- What is market opportunity analysis? how the different factors of environment are important for analysis?
- Discuss in detail the Factors that marketers need to evaluate. What are the various variables affecting the Digital Marketing Environment?
- Describe the Market opportunity analysis. How marketers can find out the marketing opportunities prevailing the market?

.

UNIT-8 DIGITAL MARKETING DECISIONS- PRODUCT, PRICE, DISTRIBUTION AND PROMOTION

STRUCTURE

8.0 OBJECTIVES

After studying the Unit, you would be able to

- Take the digital marketing decisions
- Understand the Internet and marketing mix relationship
- Find out the important components of digital marketing
- Describe the digital marketing mix
- Frame the digital marketing strategies
- Understand how effective marketing leads to profit
- Elaborate how internet has changed the market
- Fin out the role of distribution and promotion for marketing effectiveness

8.1 INTRODUCTION

Digital marketing targets a specific segment of the customer base and is interactive. Digital marketing is booming and includes search results ads, email ads, and promotional tweets, anything that involves marketing with customer feedback or a two-way interaction between company and customer.

Digital marketing is a combination of two words namely digital and marketing. In simple terms, Digital means anything can see or hear from an electronic device such as a TV, cell phone, radio. Marketing means understanding the needs of the consumer and providing the products or services accordingly. Digital Marketing is designed to reach the target audience to analyse their needs, promote product and services and market awareness through various digital platforms available online. It is done mainly on the Internet.

The Internet provides an endless array of both useful and useless type of information. Marketers can discover how many cans of Coke are left in a Coke machine on the one side of the world; watch live coffee being brewed in a coffee pot; or make contact with old friends and new friends, tour a museum, explore libraries and encyclopaedias all online. There is no doubt about it; the Internet has its good and bad aspects in ethical, moral and social perspective. Equally in marketing, the Internet has its positive and negative features. Internet can help marketers in more ways than one.

Both the organization’s own web site and other organization’s web sites and associated technologies can help marketers in many ways, from conducting research, to database building, relationship management customer service, new product development, internal communications, cost reduction and last but not the least promotion, selling and distribution.

8.2 MARKETING MIX WITH INTERNET

Marketing Research	➢ Market information ➢ Competitor information ➢ Customer information ➢ Miscellaneous information ➢ Collect cost saving ideas
Database Building	➢ World-wide club ➢ Dynamic relationship marketing
Customer Service	➢ Self-servicing customers ➢ Self-service customer abuse ➢ Self-service cost saving
New Product Development	➢ Collecting new ideas ➢ Tailor-made, products

Internal Communications	➢ Intranets ➢ Extraneous
Cost Reduction	➢ Print and distribution ➢ Phone calls ➢ Customer service ➢ Collecting cost saving tips ➢ Revenue generation
Distribution	➢ Products ➢ Services ➢ Purchases
Selling	➢ Few fairies' tale sales stories ➢ New markets ➢ Small value, big turnover ➢ Sales management tool
Promotion	➢ Have a presence ➢ Interactive advertising ➢ Creative sponsorship ➢ Sales promotions ➢ Public relations ➢ Database marketing

8.3 COMPONENTS OF DIGITAL MARKETING

- **Website streamlining (SEO)**: Search Engine Optimization is a cycle to change and adjust the site's information to improve the site's positioning in Search Engine.
- **Pay Per Click (PPC:** It's a training to pull in online rush hour gridlock towards their site by paying some measure of cash to ad distributer.
- **Email Marketing**: It's a straightforward and direct method of advanced advertising to arrive at end clients by sending them an email containing some significant data.
- **Partner Marketing**: In this sort of advanced advertising, the advertiser pays commission to member advertisers to advance their brands on various stages.
- **Content Marketing**: Contents like text, recordings, designs, pictures, and so on are utilized to associate with the focused on crowds to speak with them.

8.4 DIGITAL MARKETING DECSIONS (DIGITAL MARKETING MIX)

The marketing mix is a key foundation on which most modern marketing strategies and business activities work. As in the case of traditional marketing, Digital Marketing also uses combination of components of the marketing mix. E-marketers attempt to deliver value to ultimate customers, whether individuals, business, household, to satisfy their needs and wants in the best possible manner. Digital Marketing attempts to do the costs and benefit analysis. Benefits are based on consumer perceptions of the product by evaluating the features, attributes, brands and after-sale services. The costs associated with a product or services include the price plus other factors such as time invested in the purchase process and psychological factors.

In Digital Marketing, the internet provides facility to increase benefits while reducing the costs. The benefits include mass customization, digital delivery of products, and one-stop shopping. Lower costs are the result of direct selling, segmented pricing, and expanded delivery services through the web. The increased benefits and decreased costs can be explored by looking at the product, price, place and promotion components of the Digital Marketing mix.

- **Product:** With the introduction of Digital Marketing, the possibilities have increased for the development of several new and innovative products. Companies are working consistently to explore the customer needs and match those needs with the technological developments to create new products that can deliver value to users. The product life cycles (PLC) of the products have shortened because there has been advancement in the product development at a very high pace. With Digital Marketing the competition has also increased in the market and things are moving at a very fast face. So competition, speed of innovations and research and developments in product has become the phenomena for survival and growth thus building the potential for greater profits.

  The web is giving opportunity to both small and big business to grow. Even the small businesses can compete with the large giants globally using the web. Distance and time are not a constraint now. Transport, courier and logistic service firms like FedEx, DHL, and TNT facilitate the global exports, imports and speedy delivery of the products to the customers.

- **Price:** The standard pricing methods and strategies are not useful in Digital Marketing. Traditional pricing theory is based on the economic assumptions of laws of returns to scale which tells declining fixed costs as the volume expands. But this theory of cost advantage may not hold true in case of Digital Marketing. The constant product innovations makes the life cycles short thus require innovative pricing policies.

  The marketers have created efficiencies by lowering the costs associated with the final exchange of products marketed on the web. Self-service websites offer potential for costs and thus lower pricing on the web. Also the automation of order processing and payment adds to the potential savings.
- **Place:** The Digital Marketing uses internet and thus have expanded the ways marketers sell their products. Digital Marketing has resulted into disintermediation of many traditional channels of distribution and re-intermediation of many new and innovative channels of distributions. Many new channels of distribution have evolved from the use of internet to market goods. E-tailers represent a new distribution channel for traditional firms as well as for companies that exist only on the internet such as Amazon.com, eBay, Flipkart. In delivery process also e-marketers have developed new ways to deliver products and services including digital products, more efficient customer service, and new ways to sell directly to the customers. Innovative new intermediaries have emerged because of special needs associated with Digital Marketing
- **Promotion:** In traditional marketing, the consumer is passive in the communication process as they cannot interact with the company or marketer. They can only see and read advertisements, listen to radio ads, and view the TV commercials. Digital Marketing offers the interactive and active media where the consumers must choose, i.e., click, to see something and can view it for as long or as short as they like. They can have direct interactions with the company or marketers online. Also even in the personal selling process where the consumer is more involved, different techniques of promotion can be used by the marketers.

  Digital Marketing also changes the other promotional activities of a company. Nowadays, all advertisements contain the web address for the company offering the products and services. This is done with the intention that customers can further interact online to get more information.

CHECK YOUR PROGRESS

I. Digital Marketing is designed to reach the target audience to analyse their needs, promote product and services and market awareness through various digital platforms available online. (True/False)

II. .... is a cycle to change and adjust the site's substance/information to improve the site's positioning in Search Engine.

a) PPC
b) SEO
c) Partner marketing
d) All three

III. ……………… is straightforward and direct method of advanced advertising to arrive at end clients by sending them an email containing some significant data.

IV. The ……………….. is a key foundation on which most modern marketing strategies and business activities work.

8.5 EFFECTIVE DIGITAL MARKETING MIX DECSION IS EQUAL TO PROFITS

What's an effective "marketing mix" and why should marketer care? An online "marketing mix" is what brings all of the different components of successful selling on the internet into a congruent and cohesive system to achieve its business goals. It effectively targets the right group of people, with the right products and services, at the right time and place, which all lends itself to the end goal of substantial and increased profits. It's also been called the 7 Ps of marketing.

The 7 Ps of the online marketing mix include **People, Product, Price, Promotion, Place, Process, and Physical Evidence.**

- **People:** This is where it all begins, people. First it's very important that marketer must find out through research if there are enough people in demand of a certain types of products and services. Typically marketer can get a good idea of the demand for certain products and services by research. In certain markets there may be several different types of products which can cater to a particular segment of people. If there is a high demand of certain types of products, then this can be great news for business. This will help marketers when it comes to creating their sales funnel and for future product development for repeat sales to that segment of people down the road. The competitors may offer something similar, but if the marketers treat customers and satisfy their needs than it will result into them dealing with the same marketer again in the future!

- **Product:** Do the marketers have right products for the market they are trying to target? As mentioned earlier, marketers absolutely need to have the right types of products that are in demand for market. But, how do they know?

  If there is already products and services available than marketers should think, what can they do to offer a better product to this group of people than the competitors. What can give them the edge to stand out? Marketers can look for the consumer needs from online forums where people are asking where they can find a certain script that does "this or that", or who provides the best solution to a service they need. Skype groups, Facebook groups can be great for this type of research as well.
- **Price:** Price is the next thing that's important within the "Marketing Mix". This is actually an area where marketers have to be somewhat careful, plus, be mindful of what their target market might actually be willing to pay. Price is a touchy area. Pricing has a lot to do with how a product is perceived by the customers as well. If price of the product is too low, then it might be considered as somewhat inferior to the competitions. If marketers try and increase the price too high, it could result into creating a difference for people who may not think the benefits of the product or service are worth the asked price.

  The marketers should consider the competitor's products and services, their positioning, and pricing also how fit it is in the market. If company already have current customers, they can take their feedback about their perception.
- **Promotion:** The fourth "P" in the "Marketing Mix" is Promotion. This part of process in the marketing plan can include several components such as how the marketers may be communicating about the product to the customers, including:

  - ✓ PPC
  - ✓ Article marketing
  - ✓ Social media marketing such as YouTube, Facebook, Twitter etc.
  - ✓ JV partners
  - ✓ Media buys

The selection of promotion media and how the marketers will promote will depend on the budget, the message to communicate, and the group of target customers. Things to consider:

- ✓ What is the promotion budget?
- ✓ Access to partners willing to promote
- ✓ The types of incentives that can be offered

- **Place:** At first glance, marketers might think place is obvious. The place here means selling the products or services using web site. Some places to consider listing product: Clickbank, https://seller.flipkart.com/, *ebay.in* , PayDotCom, *sellers.snapdeal.com/*
- **Process:** The next thing to consider within the marketing mix is the process of how products are delivered. This is generally the technical part of the equation. For this the companies are going to need an effective "sales system".

  It's how marketers are going to get paid, and deliver the products. For most digital products this is all done online. However, if the products or services happen to be physical, marketers might require the use of a distribution channels to support the physical distribution of the products. There are several effective sales systems available in the market today which is capable of handling the main offer and any possible up-sell or down-sell offers.
- **Physical Evidence:** When it comes to online marketing "physical evidence" pertains to how organisations, products, or marketers are presented in the market place.

  Here there is need to give importance to branding, which is extremely important. This can include the professionalism in how products, logo, and the entire message that "brand" is trying to get across to the public is perceived. It's important that company and its products and services come across in a congruent manner, where each part can complement one another, to create the overall "brand". When customer see the Nike swoosh, or a McDonalds sign, they know immediately what their logos stand for.

8.6 INTERNET HAS CHANGED THE MARKET

The internet has changed the way to sell products and services. That's a fact, consumers now use the internet to research and purchase products/services online, so firms need online strategies to attract and retain customers. The Digital Marketing mix considers the elements of traditional marketing, presenting the marketing mix online. Let us discuss this further:

- **E-Product Strategies:** We walk into a shop and see a product we like, we can assess it, touch it. Online, this immediate tangibility disappears. But, is that a disadvantage? Within the Indian e-commerce sales are increasing at extremely high rates. Why? What does buy products online offer over one to one sale? Firstly, there are clear online facts about the product, customers are purchasing. The buyer knows immediately about product features, the facts, not sales persons assumptions.

  The buying process is also customised for returning visitors, making repeat purchases easier. Organisations can also offer immediately ancillary products along with the main purchase. Example, offer to buy extra printer cartridges along printer purchases online. The product can also be customised to consumers' needs. www.nike.com offer customised trainers to users online. Users can design and see their trainers online before they order.

- **E-Price Strategies:** As mentioned in Marketing Mix section, pricing is always difficult to do and must take into account many considerations. Traditionally pricing was about finding about costs, discovering how much consumers are willing to pay, taking account competition pricing then setting your price. The internet has made pricing very competitive. Many costs i.e., store costs, staff cost have disappeared for complete online stores, placing price pressures on traditional retailers.

  The internet gives consumers the power to shop around for the best deal at a click of a button. Website such www.mysmartprice.com compares mobile prices from different websites informing consumers of where the best deal is. Such easy access to information helps to maintain prices within the online world. The growth of online auctions also helps consumers to dictate price. The online auction company www.ebay.com has grown in popularity with thousands of buyers and seller bidding daily.

  E-pricing can also easily reward loyal customers. Technology allows repeat visitors to be tracked, easily allowing loyalty incentives to be targeted towards them. Payment is also easy, Paypal, or credit cards allows for easy online payments. However, the downside to this is internet fraud, which is growing rapidly around the world.

- **E-Place Strategies:** One of the biggest changes to the marketing mix is online purchasing. Consumers can purchase directly from the manufacturers eliminating the role of various intermediaries such as retailers and wholesalers totally from the transactions. The challenge for online retailers is to ensure that the product is delivered to the consumer within a reasonable time. Location is important within place strategy. Online location can refer to where links are placed on other websites. Placing a link on www.google.com home page would generate high consumer traffic. Knowing customer and knowing where they visit should help to understand where to place online links and advertisements.
- **E-Promotion strategies:** Promoting products and service online is concerned with a number of issues. Having a recognisable domain name is first stage towards e-promotion. Organisation such as www.apple.com, www.samsung.com, has successfully positioned the brand on the online world as an online bank.

  Most of the organisations today have websites and they use them for online promotions and Digital Marketing. Placing banner advertisements on other web pages is a common form of e-promotion. Banner ads must be placed where potential customers browse. Web public relations are another approach to promoting online. News worthy stories based on product or service launches can be placed on the company's web page, or WPR articles sent to review sites for consumers to read. Online promotion helps in attracting the attention of consumers and influencing their purchase behaviour. Direct email is a popular and common form of e-promotions, although slowly becoming the most disliked form of online promotion by many consumers. Organisations can send e-leaflets to hundreds and thousands of respondents, hoping a small percentage will reply. The problem is that for every 100 emails sent the response rate is not more than 1 to 2 per cent. Direct emailing is also known as SPAM which stands for Sending Persistent Annoying email.

CHECK YOUR PROGRESS

V. The product life cycles of the products have shortened because there has been advancement in the product development at a very high pace. (True/False)

VI. Traditional pricing theory is not based on the economic assumptions of laws of returns to scale which tells declining fixed costs as the volume expands. (True/False)

VII. In traditional marketing, the consumer is ……………. in the communication process as they cannot interact with the company or marketer.

VIII. Marketer can not get a good idea of the demand for certain products and services by research. (True/False)

IX. ……………. is important within place marketing strategy.

a) Value
b) Price
c) Location
d) Advertising

8.7 DISTRIBUTION

Place or distribution channel means how and where the product will be provided to the customer. In Digital marketing, there are numerous alternatives for the organization to sell on a wide scope of platforms. There are number of channels to use but not every channel is used to distribute every product. The product could be distributed via single channel or set of channels, depending upon the type of product they are selling, budget, audience to be reached etc.

some of the channels are:

- Social media stages (Facebook Shop)
- Dedicated online store (WooCommerce, Shopify,).
- Online commercial centres (Amazon, eBay)

Selecting the right channel for the product is crucial. The channel should be easily accessible, and the decision on the right kind of channel highly depends upon the targeted audience.

Whenever choosing place open doors in organization's web based showcasing blend, the requirement for sorting out how much market inclusion it is ready to reach inside that place. An online store can possibly have more extensive inclusion if there's a spending plan accessible for contacting the crowd/ people.

8.8 PROMOTION

Under digital marketing promotion means utilization of web based showcasing specialized apparatuses to arrive at objective people. Direct communication, sales promotion, public relations, and advertising helps to reach to the people digitally. The organization and its online items require various methodologies when arriving at its clients.

Organizations can’t put resources into each channel available in the market, the need to comprehend the crowd initially is critical for delivering and implementing a compelling promoting system digitally. Despite the fact that being limited by spending plan, time, areas, innovation, each organization can discover and use viable strategies for web-based advertising to achieve pre decided goals. The introductory showcasing procedure should incorporate limitations to keep away from plans that we can't execute as expected. For instance, putting resources into site. With the help of web analytics, the progress of the selected tool and strategy can be tracked in digital marketing. Digital marketing is quick and provides huge data for future policy making. Online promotion blend incorporates:

- **Advertising** - utilizing relational interchanges media.
- **Sales advancement** - helps present moment to animate fast deal.
- **Public Relations** - making a good picture for the intended interest group proceed.
- **Direct advertising** - utilizing client information bases for conditional and social advertising via reactions and estimations.

8.9 PROMOTION TOOLS

- Twitter can bring organizations image mindfulness up in a very brief timeframe; the chance of utilizing hashtags will keep the brand to stay apparent active on the web. Moreover, this stage offers a ton of helpful publicizing choices .twitter handles helps in one on one conversations and helps in creating a campaign starting with a # (hashtag) and can reach to larger crowd and become trending on twitter .
- Google+ is fit for advancing the product both among private clients and different organizations. So in case the organization is keen on Business to business activities, Google+ may be a decent wagered, even it can improve the SEO over the long-time.
- Pinterest is a visual stage. Pinterest permits organization to flaunt the item contributions, comparative pictures, and curated pin-sheets. Pinterest is a desert spring for specialists, ladies and grooms. Travel schedules, proposed pressing records, arranging thoughts, make activities, and wedding motivation are easily done on this platform .

- Instagram has huge amount of dynamic clients and has similar publicizing alternatives as Facebook. It's a visual stage, hHere people can follow the products official account and the people will get the information about the product, offers etc. in the form of Instagram stories and posts. Even the number of followers tells about the liking of the product by the people around the globe.
- LinkedIn This stage is known as the essential long range informal communication stage for experts. Business can utilize LinkedIn to refresh clients in the list, associate with industry accomplices, and enrol new colleagues. Being an expert platform, it is preferred decision for business-to-business web-based media advertising.
- YouTube is a video real time stage which can be of extraordinary assistance to any organization as it can empower them to discover a gathering of individuals whom the organization wants to focus on and change over them into clients by instructing, engaging and giving answer for clients issues.
- Facebook permits the organization to target clients by their area, Age, Gender, pay, etc., which assists the business with directing people to the online business shops. Facebook help the business in accomplishing staggered advertising Like and offer are astounding highlight. If somebody likes the organizations page the person will get information about new items and advancements .
- Email advertising is one of the conventional computerized showcasing channels that is both viable and economical. The leads produced by the marketers, can be utilized to send messages and contact the likely clients. The business can send clients pamphlets, digests, lists, direct ads, update messages, messages illuminating customers about extraordinary arrangements and limits, and the email promoting effort can be either centred around raising the image mindfulness or transformation rates
- In Display Advertising business advertisements put on the outsider sites or applications. This advertisements is normally connected to a greeting page of the advanced website. Display promoting can be a solid match in the event that business realize that some particular sites or applications are especially valuable for the business intended interest group, and with their assistance, we can raise the image mindfulness and pull in new people. The costs of these advertisements are generally determined on the CPM premise.
- Affiliate showcasing is affiliating with different organizations and influencers in business specialty, organization can spread its image to a more extensive crowd and

increment its deals. it helps in setting advertisements or connections on partner sites and visitor presenting on sharing leads and trading complementary whoops.

CHECK YOUR PROGRESS

X. In Digital marketing, there are numerous alternatives for the organization to sell on a wide scope of platforms. (True/False)

XI. Under …………….. promotion means utilization of web based showcasing specialized apparatuses to arrive at objective people.

XII. …………… can bring organizations image mindfulness up in a very brief timeframe.

a) Twitter
b) Google Plus
c) Facebook
d) Pinterest

XIII. ………………. has huge amount of dynamic clients and has similar publicizing alternatives as Facebook.

XIV. ………….. permits the organization to target clients by their area, Age, Gender, pay, etc.

a) Twitter
b) Google Plus
c) Facebook
d) Pinterest

XV. Email advertising is one of the conventional computerized showcasing channels that is both viable and economical. (True/False)

8.10 LET US SUM UP

Digital marketing targets a specific segment of the customer base and is interactive. Digital marketing is booming and includes search results ads, email ads, and promotional tweets - anything that involves marketing with customer feedback or a two-way interaction between company and customer. The marketing mix is a key foundation on which most modern marketing strategies and business activities work. As in the case of traditional marketing, Digital Marketing also uses combination of components of the marketing mix. E-marketers attempt to deliver value to ultimate customers, whether individuals, business, household, to satisfy their needs and wants in the best possible manner. In Digital Marketing, the internet provides facility to increase benefits while reducing the costs. The benefits include mass customization, digital delivery of products, and one-stop shopping. What’s an effective “marketing mix” and why should marketer care? An online “marketing mix” is what brings all of the different components of successful selling on the internet into a congruent and cohesive system to achieve its business goals. It effectively targets the right group of people, with the right products and services, at the right time and place, which all lends itself to the end goal of substantial and increased profits. In Digital marketing, there are numerous alternatives for the organization to sell on a wide scope of platforms. There are number of channels to use but not every channel is used to distribute every product. The product could be distributed via single channel or set of channels, depending upon the type of product they are selling, budget, audience to be reached etc. Under digital marketing promotion means utilization of web based showcasing specialized apparatuses to arrive at objective people. direct communication, sales promotion, public relations, and advertising helps to reach to the people digitally.

8.11 KEY WORDS

MARKETING Marketing means understanding the needs of the consumer and providing the products or services accordingly.

SEARCH ENGINE OPTIMIZATION It is a cycle to change and adjust the site's substance/information to improve the site's positioning in Search Engine.

EMAIL MARKETING It's a straightforward and direct method of advanced advertising to arrive at end clients.

PARTNER MARKETING In this sort of advanced advertising, the advertiser pays commission to member advertisers to advance their brands on various stages.

MARKETING MIX It brings all of the different components of successful selling on the internet into a congruent and cohesive system.

DISTRIBUTION It means how and where the product will be provided to the customer.

PROMOTION It means utilization of web based showcasing specialized apparatuses to arrive at objective people.

8.12 ANSWERS TO CHECK THE PROGRESS

I. True

II. SEO

III. Email Marketing

IV. Marketing Mix

V. True

VI. False

VII. Passive

VIII. False

IX. Location

X. True

XI. Digital marketing

XII. Twitter

XIII.	Instagram	**XIV.**	Facebook	**XV.**	True

8.13 TERMINAL QUESTIONS

- Define the term marketing mix. Explain structure of marketing mix and its elements.
- What are the issues and challenges in the way of Digital Marketing? What do you think are the possible opportunities for Digital Marketing?
- What is marketing mix and what would be the role of internet in marketing mix?
- What are the critical components of digital marketing and how they influence the digital marketing mix?
- How the marketing manager ensure the effectiveness of marketing strategies with the help of marketing mix?
- Why marketing mix effectiveness is important and how it will contribute towards profitability of organization?
- What kind of changes have taken place in the marketing mix as a result of internet?
- What do you mean by marketing mix? Discuss various elements of marketing mix with suitable examples.
- Write short notes on the followings: -
    a. Digital Marketing Mix Strategies
    b. E-Product Mix
    c. E-Price Mix
    d. E-Place Mix
    e. E-Promotion Mix

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT 9- DIGITAL MARKETING STRATEGY FORMULATION AND EXECUTION

STRUCTURE

9.0 OBJECTIVES

After studying the Unit, you would be able to

- Have basic understanding for the digital marketing strategies
- Understand why digital marketing strategies are important and its components
- Find out how we can lake digital marketing strategies more effective
- Compare hoe inbound marketing is different from outbound marketing
- Effective use of inbound marketing methodology
- Know the process for creating digital marketing strategy
- Elaborate the segmentation, targeting and positioning
- Develop the positioning marketing strategy
- Know the importance of positioning and its importance

9.1 INTRODUCTION

The Internet is **not** synonymous with *World Wide Web*, it is a massive network of networks or a networking infrastructure. It connects millions of computers together globally, forming a network in which any computer can communicate with any other computer as long as they are both connected to the Internet. The World Wide Web, or simply the Web, is a way of accessing information over the medium of the Internet. It is an information-sharing model that is built on top of the Internet.

Digital Marketing is the fastest growing and most exciting branch of marketing today. As the world becomes ever more connected, keeping up with developments and trends is vital for marketers trying to reach new audiences, who are more discerning, fragmented and cynical than ever. Technology and software are changing at such a high rate that it seems almost impossible to keep up with trends. Products and services are evolving and adapting to the online sphere. The web is constantly shifting, growing and changing everything is fleeting.

Digital marketing refers to the promotion of goods or services through the use of digital technologies such as the internet, social media, cell phones, and other digital platforms. It comprises not only advertising that is shown on websites, but also other kinds of online activities like email and social networking. Every aspect of Digital Marketing is digital, meaning that it is electronic information that is transmitted on a computer or similar device, though naturally it can tie in with traditional offline advertising and sales too.

9.2 DIGITAL MARKETING STRATEGY IMPORTANT

To achieve any aim, having a strategy is almost as important as having a plan. It optimises process execution, allowing for high efficacy while keeping an eye on different areas to evaluate data analytics and take appropriate action.

This allows for continuous improvement in various digital marketing techniques such as search marketing, conversion rate optimization (website user experience), email marketing, and social media marketing to help achieve business goal of attracting new customers and strengthening existing relationships.

Company also needs a competitive and long-term content marketing plan to reach and convert more consumers by leveraging the power of online content and combining digital and conventional marketing platforms.

As companies embrace digital transformation in their core operations and new avenues for reaching consumers, having a solid digital marketing plan is critical to their success.

9.3 COMPONENTS OF DIGITAL MARKETING STRATEGY

A digital marketing strategy should include the following elements:

- **Creating a brand:** To identify brand, use brand guidelines. Consider the company's USPs (Unique Selling Propositions).
- **Demographics:** Understanding demographic details such as age, gender, and place, as well as the reasons that motivate customers to select products and services, is key to developing consumer personas. Use Google Analytics to help build buyer personas and segment them based on organic traffic keywords.
- **Identifying target markets:** Perform market research and find possible target markets using market research tools including Google Keyword Planner, Google Trends, and Facebook Audience Insights.
- **Doing competitive analysis:** Evaluate the rivals and perform a SWOT analysis using web and social analytics tools including similarweb, Semrush, and SocialBakers.
- **Appropriate resource management and allocation to achieve objectives:** Manpower (in-house or outsourced, external marketing agencies) and budget are two resources.
- **Reviewing and improving campaign results:** Use digital marketing analytics tools like Google Analytics to monitor, assess, and report on the performance of digital marketing campaigns.

9.4 WAYS TO MAKE DIGITAL MARKETING STRATEGIES MORE EFFECTIVE

- **Involve Customer:** Customer needs to be motivated online to post their views regarding various kinds of products and there queries and questions should also be cleared promptly.
- **Consider all the customers:** All the customers should be kept in mind before marketing a product. The taste as well as preferences of all the customers should be considered and segmental.
- **Social welfare focused advertising:** Nowadays there is a trend of working for social welfare, so some social work should also be included so that customers feel attracted.

- **Advertising Frequency:** Specific occasions need to be consider while deciding the frequency of online advertising.
- **Advertising through effective ways**: The advertising of the products and services should be done in such a way so that customers feel attracted to see that particular product.
- **Target Customers:** The customers which are beneficial as well as profitable for the organization should be targeted so that they stood beneficial for the company and therefore benefit the company in long run afterwards.
- **Optimum use of resources:** Online Marketing is economical in terms of time, money as well as efforts involved. So all the resources which are used should be used in wise as well as efficient way.
- **Adequate Investment:** Investment in Online Marketing should be Adequate; there should be neither more nor less investment for Digital Marketing, it must be as per the requirement of the company and the competition in the market.
- **Market's Competition:** One should work by seeing others, that is the other competitors workings can't be ignored whenever there is an advantage than should be grabbed immediately.
- **Content of the advertisement:** The content that seems to be interesting should be taken into account so that customers feel attracted to see that particular advertisement.
- **Target market planning:** Proper planning should be done before targeting customer. All information about the customers should be collected before targeting any customer's segment.
- **Content Updating:** The content should be uploaded regularly so that the customers do not feel uninterested in the products.
- **Competitor's Strategy:** The strategy of competitors should be known so that counter strategy can be made by the companies. This is important because customer these days switch companies or products very easily.
- **Innovativeness:** Time to time updating of the technology should be there in the organization so that nothing outdated should be there in the company.
- **Contact with Customers:** The customers should be contacted time to time and proper reviews of the customers and their preferences about the various products should also be known.

9.5 INBOUND VERSUS OUTBOUND MARKETING

Outbound marketing was typically the traditional approach to market business and this was known as an interruption-based marketing.

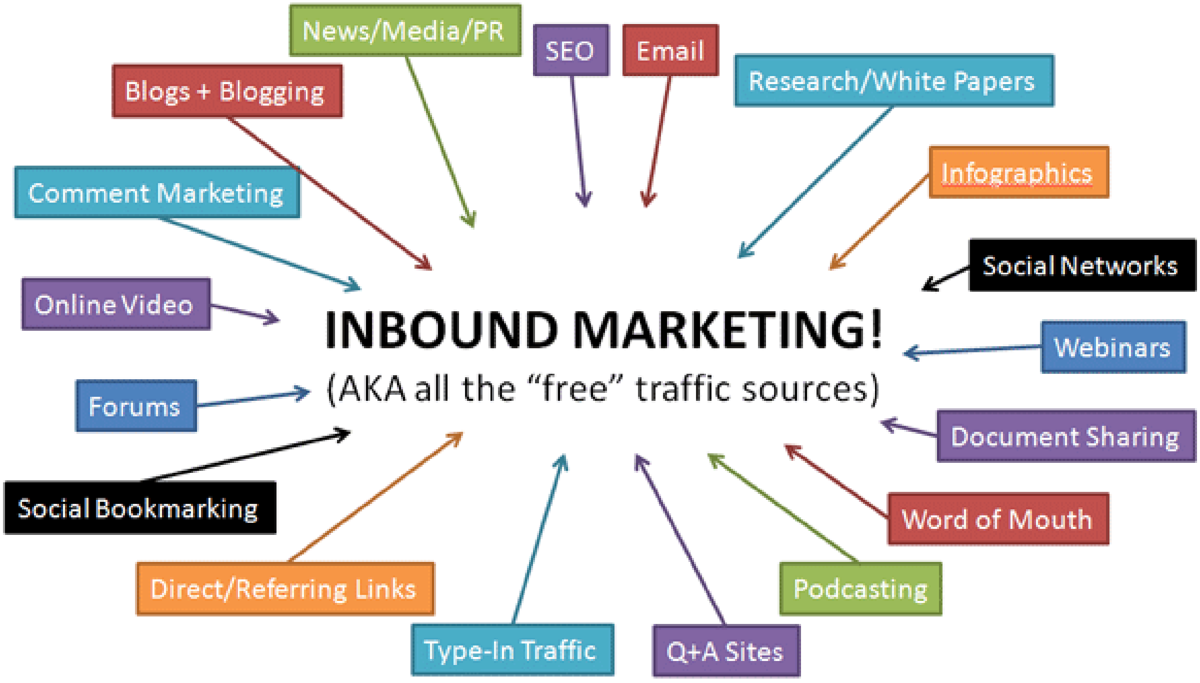


Image Source: http://www.seomoz.org/blog/inbound-marketing-is-taking-off)

Inbound marketing is where marketers provide something of value that attracts a customer to come to shop. When marketers attract that customer, they try to build a relationship.

9.6 INBOUND MARKETING METHODOLOGY

1. **Attract:** In inbound marketing marketers want attract the right customer to website and to brand with a quality, answers that questions they have. We have to use blogging, social media, keyword and SEO strategy.
2. **Convert:** After attracting the strangers on websites, a company will ready to convert them into prospect, collect their contact information. Emails give important information for an inbound marketer. This includes eBooks, whitepapers, case studies and tips sheets.
3. **Close:** With the help of tools like lead scoring closed-looks reporting, lead nurturing and vast lead intelligence, paired with lead follow up best practices and sales tactics, marketers will then work to transform those leads into now business.

4. **Delight:** Delight refers to deliver the right information to the right person at the right person at the right time. Delight tools are such as Smart content, Email marketing, conversations inbox, attribution reporting and marketing automation can be used by the marketers.

9.7 BREAKDOWN OF THE KEY FORMS OF DIGITAL MARKETING

Types	Explanation
Search Engine Optimization (SEO)	This is a process used to optimize a website so that it appears more in Google searches.
E-mail marketing	E-mail marketing is marketing through delivery of e-mails to current and potential clients.
Affiliate marketing	Affiliate marketing is where marketers get third party companies to promote company with a commission paid for a lead or sale.
Online PR	PR can be a very effective tool for promoting business and there are many ways of doing this online.
Online Advertising	There are many forms of online advertising. Although advertising through the likes of newspapers is not that effective as targeted advertising online can be very effective. For example, if we are searching to buy a Canon camera and the ad is directly relevant to this then we don't view this as a bad thing.
Social media	Social media marketing is marketing through conversation and engagement with potential customers.
Website	The website is a key Digital Marketing tool used to promote business online.

CHECK YOUR PROGRESS

I. Digital Marketing is the fastest growing and most exciting branch of marketing today. (True/False)

II. ………….. refers to the promotion of goods or services through the use of digital technologies such as the internet, social media, cell phones, and other digital platforms.

III. Customer needs not to be motivated online to post their views regarding various kinds of products to involve the customer in strategy formulation. (True/False)

IV. …………….. need to be considered while deciding the frequency of online advertising.

V. The customers which are beneficial as well as profitable for the organization should be ………………..

a) Targeted

b) Segmented

c) Both

D) None of both

VI. The strategy of competitors should be known so that counter strategy can be made by the companies. (True/False)

VII. Outbound marketing known as an …………………..

a) Interruption-based marketing

b) Psychological based marketing

c) Content based Marketing

d) All three

VIII. ………………….. is where we get third party companies to promote our company with a commission paid for a lead or sale.

a) Email Marketing

b) Direct Marketing

c) Affiliate Marketing

d) Online PR

9.8 PROCESS TO CREATE AN DIGITAL MARKETING STRATEGY

Online and brick-and-mortar businesses require Digital Marketing strategies. A comprehensive Digital Marketing strategy can launch or increase sales substantially for a business. Digital Marketing requires knowledge of social media, search engine optimization (SEO), blogs, email lists, affiliate marketing and more.

- **Create a Customer Persona:** Previously, digital marketers accrued qualitative and quantitative data approximately the target audience that blanketed age, gender, profession, hobbies, and location. The definition of the capability consumer has been taken to another level currently, with buyer personas taking over. Now, the most effective advertising strategies are incomplete without them. Buyer personas are a term used to explain the proper customer that can be defined through surveying and contacting the pre-defined target organization. To define personal consumer persona, marketers could make use of on-line equipment like MakeMyPersona and Persona Creator.

- **Identify Goals and Tools:** Every expert marketer knows how vital this level is, without dreams, the strategy certainly will not work, so defining them is primary challenge. Make certain that the goals are suitable and lead them to measurable. Example of a vulnerable advertising intention: Increase the conversions at the internet site next year and create two promotional gives.

- **Focus on Blogging:** Creating notable content material is essential for any digital marketing strategy. Blogging is one way to create content for a website and enlarge content material advertising method. This key component of the method plays a few vital functions, including bringing extra visitors to website, reinforcing social media presence, positioning towards different organizations, and ranking all the ones lengthy-tail seek queries and key phrases.

- **Evaluate Existing Digital Marketing Channels:** Analyze the virtual property and channels that are already the usage of, inclusive of website, weblog content, social media account, word-of-mouth, local advertising, Google AdWords, paid advertising, and so on. Ask, how powerful had been they and how can they be used next year?

- **Automate Marketing:** Marketing automation is huge, and it’s best getting higher with new generation and software. Today’s main marketing automation platforms can do wonders for virtual advertising method, making it easy for automate activities such as content advertising, e-mail advertising and marketing, lead generation, and greater. It is easy to find advertising automation platforms which are integrated together with CRM, which improves the whole sales procedure.

- **Nail Mobile Optimization:** It’s a cellular-first era, and there’s no excuse for now not enhancing web page for a brilliant mobile revel in. Mobile optimization includes web page velocity, website online layout, and diverse different SEO practices to make sure that folks that come for web page the usage of a mobile device have a very good revel in. Review cellular responsive layout and e-mail templates to make sure that the use of the latest methods and offer handiest applicable.

- **Easy for Customers to Reach:** Online business is all approximately connecting with ability customers, so traffic to internet site ought to be excessive to ensure nice outcomes. Digital marketing strategy therefore should make certain that visitors are not exerting unnecessary attempt to hook up. Engaging together with customers is priceless in business. To remove the pointless effort, marketers ought to make sure that touchdown pages do now not ask for inappropriate facts. Make call-to-actions honestly seen at the landing page and other pages at the website.

- **Use the Right Technology:** Online marketing attempt is wasted if an ineffective or irrelevant technology is used. For a proper digital advertising approach, marketers shall need equipment along with advertising and marketing automation software in addition to enterprise equipment. There are equipment that allow marketer to during each level of the virtual advertising method, inclusive of equipment that assist identify goals to gear that assist song in which traffic are coming from.

- **Confirm Differentiators:** If corporation is not unique from competition, marketer should now not count on excessive profits. Make sure you convert your difference from the others via talking in your clients and asking them why they decided on your enterprise. Every business enterprise has a unique promoting proposition, an attribute that units you apart from the opposition. Do the research and work to discover what this is? Then, restore your existing differentiators if wanted. Incorporate this message into your content material marketing campaigns.

- **Track Process:** Constant tracking ought to be provided to make sure that the method is running successfully. This is executed by assigning teams that would supply this task. Remember to examine from the errors and continually find development opportunities.

9.9 MARKET SEGMENTATION, TARGETING AND POSITIONING

It is very important to learn what market segmentation, why it important is and the different dimensions used by marketers to segment the population. How marketers evaluate and select potential market segments is explained as is the development of a targeting strategy. We understand how a firm develops and implements a positioning strategy and creates a customer relationship management strategy to increase long-term success and profits.

9.10 E-MARKET SEGMENTATION

Market segmentation is an important process in conventional marketing. Segmentation is an important step in ensuring successful implementation of marketing strategies. Market segmentation means dividing a market into different groups or categories according to certain criteria such as household income, level of income, level of education, age, gender, hobbies and more. Such data can be collected through questionnaire survey or from statistical department. Once we have those groups ready, we can then use differentiated marketing strategies to target those groups based on their needs and wants. For example, ask yourself what you can sell to the senior citizens? Your answers could be nutritional products, tour packages, medicines, indoor exercising machines and more.

Now, how do we do market segmentation in the Internet marketplace, or rather marketspace? The concept is basically the same, the only difference is companies have a much larger marketplace, and the main concern is language and cultural differences. For example, if marketers wish to promote inbound tour packages, they can target adult travelers from different countries, using website that offers different major languages in the world, such as English, French, Spanish, German, Chinese and Japanese. If companies cannot do it themselves, they can always outsource to companies that provide translation services. Besides, organizations might want to customize the website according to different cultures, particularly those relating to food and things they like to see.

One of the most popular Internet outsourcing companies is **Odesk.com.** In this website, one can outsource jobs to suitable contractors who are mostly individual freelancers. One can bargain with them the price as well as other terms and conditions for handling their jobs. Jobs that can be outsourced are webpage design, contents writing and editing, graphic design, translation, copywriting, customer support and more.

9.11 BASES FOR SEGMENTATION

The next step in developing a market segmentation strategy is to select the most appropriate bases on which to segment the market. The major bases to segment consumer markets are the following:

1. Geographic segmentation
2. Demographic segmentation

3. Psychographic segmentation

9.12 TARGETING

In targeting, the marketers evaluate the attractiveness of each potential segment and decide which of these groups they will invest resources against to try to turn them into customers. The customer group or groups selected are the firm's target market. Target marketing can be carried out at different levels:

- Undifferentiated Marketing
- Differentiated Marketing
- Concentrated Marketing or Niche Marketing
- Customized Marketing or Micro Marketing

CHECK YOUR PROGRESS

IX. Digital Marketing requires knowledge of social media, search engine optimization (SEO), blogs, email lists, affiliate marketing and more. (True/False)

X. ………… is one way to create nice content for a website and enlarge content material advertising method.

a) Email Marketing
b) Blogging
c) Advertising
d) Branding

XI. Online marketing attempt is wasted if an ineffective or irrelevant …………… is used.

XII. Constant tracking ought to be provided to make sure that the method is running successfully. (True/False)

XIII. Market …………… means dividing a market into different groups or categories according to certain criteria.

a) Positioning
b) Targeting
c) Segmentation
d) None of these

XIV. A ……………. market consists of a set of buyers who share common needs or characteristics that the company decides to serve.

a) Positioned
b) Target
c) Segmented
d) None of these

XV. An ……………………….. strategy is one that appeals to a wide-spectrum of people.

XVI. A ……………….. is often useful for smaller firms that do not have the resources or the desire to be all things to all people.

9.13 MARKET POSITIONING

The concept of positioning was propounded by two advertising executives, Al Ries and Jack Trout. They considered positioning as a creative exercise done by marketing people with both existing as well as new products. According to them positioning starts with a product, a piece of merchandise, a service, a company, an institution or even a person. But positioning is not what is done to a product. Positioning is what marketers do to the mind of the prospect. That is, marketers position the product in the mind of the prospect.

9.14 DEVELOPING A POSITIONING STRATEGY

Positioning means developing a marketing strategy aimed at influencing how a particular market segment perceives a product or service in comparison to the competition. Developing a positioning strategy entails gaining a clear understanding of the criteria that the target consumers use to evaluate competing products and then convincing them that your product will meet those needs. Positioning can be done in many ways. Marketers must devise a marketing mix that will effectively target the segment's members by positioning their products to appeal to that segment. A first step is to analyse the competitors' positions in the marketplace. Who are the direct competitors and what products or services are they providing? Indirect competition can also be important. The following is a list of some established product positioning strategies. (Already discussed in Unit 3)

- Against a competitor
- Product Categories
- Away from a Competitor
- Benefits
- Product Attributes
- Usage Occasions
- Users

9.15 PRODUCT POSITIONING AND DIFFERENTIATION

Marketers must also develop a positioning strategy that includes offering a product or service with a competitive advantage, providing a reason why consumers will perceive the product as better than the competition. Once a positioning strategy is set, marketers must finalize the marketing mix by putting all the pieces into place. The elements of the marketing mix must match the selected segment. This means that the goods or services must deliver benefits that the segment values, such as convenience or status. Furthermore, marketers must price this offering at a level these consumers will pay, make the offering available at places consumers are likely to go, and correctly communicate the offering's benefits in locations where consumers are likely to take notice. Finally, marketers must evaluate the target market's responses so they can modify strategies as needed. Over time, the firm may find that it needs to change which segments it targets or even redo a product's position to respond to marketplace changes.

In volatile markets, it can be necessary even urgent to reposition an entire company, rather than just a product line or brand. When Goldman Sachs and Morgan Stanley suddenly shifted from investment to commercial banks, for example, the expectations of investors, employees, clients and regulators all needed to shift, and each company needed to influence how these perceptions changed. Doing so involves repositioning the entire firm. This is especially true of small and medium-sized firms, many of which often lack strong brands for individual product lines. In a prolonged recession, business approaches that were effective during healthy economies often become ineffective and it becomes necessary to change a firm's positioning. Repositioning a company involves more than a marketing challenge. It involves making hard decisions about how a market is shifting and how a firm's competitors will react. Often these decisions must be made without the benefit of sufficient information, simply because the definition of "volatility" is that change becomes difficult or impossible to predict.

9.16 PRODUCT POSITIONING PROCESS

The product positioning process involves the following stages:

i. Defining the market in which the product or brand will compete (who the relevant and prospect buyers are)
ii. Identifying the attributes (also called dimensions) that define the product 'space'

iii. Collecting information from a sample of customers about their perceptions of each product on the relevant attributes

iv. Determine each product's share of mind

v. Determine each product's current location in the product space

vi. Determine the target market's preferred combination of attributes (referred to as an ideal vector)

vii. Examine the fit between the product and the market.

9.17 IMPORTANCE OF POSITIONING

Product positioning is a crucial ingredient in the buying process and should never be left to chance. It is company's opportunity to influence the market's perception of their products and services. Failure to proactively address product positioning is unlikely to end well. With or without marketer input, customers will position the product, probably based on information from the competitors, which will not flatter the organizations. Clear, concise, meaningful product positioning also helps companies cut through the relentless advertising and marketing noise of the marketplace. In customer's mind, product positioning gives the messages some context so they can be better heard and accepted.

CHECK YOUR PROGRESS

XVII. …………….. is developing a product and brand image in the minds of consumers.

a) Positioning
b) Targeting
c) Segmentation
d) None of these

XVIII. Effective positioning involves a good understanding of competing products and the benefits that are sought by the target market. (True/False)

XIX. ………….. means developing a marketing strategy aimed at influencing how a particular market segment perceives a product or service in comparison to the competition.

XX. Once a positioning strategy is set, marketers must finalize the marketing mix by putting all the pieces into place. (True/False)

XXI. Product positioning is a crucial ingredient in the buying process and should never be left to chance. (True/False)

9.18 LET US SUM UP

The goal of marketing is to create value and satisfy needs. However, everyone's needs are not the same. Understanding needs is a complex task. Market segmentation is an important process in conventional marketing. Segmentation is an important step in ensuring successful implementation of marketing strategies. Now, how do we do market segmentation in the Internet marketplace, or rather marketspace? The concept is basically the same, the only difference is companies have a much larger marketplace. Market research analysis using segmentation is a basic component of any marketing effort. The marketer will have to try different segmentation bases or segmentation variables, alone or in combination, to find the best way to view the market structures. In contrast to consumers, industrial customers tend to be fewer in number and purchase larger quantities. In targeting, the marketers evaluate the attractiveness of each potential segment and decide which of these groups they will invest resources against to try to turn them into customers. The customer group or groups selected are the firm's target market. A target market consists of a set of buyers who share common needs or characteristics that the company decides to serve. The business can positively influence the perceptions of its chosen customer base through strategic promotional activities and by carefully defining the business' marketing mix. Effective positioning involves a good understanding of competing products and the benefits that are sought by the target market. In volatile markets, it can be necessary even urgent to reposition an entire company, rather than just a product line or brand. Product positioning is a crucial ingredient in the buying process and should never be left to chance. It is company's opportunity to influence the market's perception of their products and services.

9.19 KEY WORDS

DIGITAL MARKETING	It refers to the promotion of goods or services through the use of digital technologies such as the internet,
TARGET CUSTOMERS	The customers which are beneficial as well as profitable for the organization.
OUTBOUND MARKETING	Typically, the traditional approach to market business.
INBOUND MARKETING	Inbound marketing is where marketers provide something of value that attracts a customer to come to shop.

DELIGHT — It refers to deliver the right information to the right person at the right person at the right time.

AFFILIATE MARKETING — In this, we get third party companies to promote our company with a commission paid for a lead or sale.

MARKET SEGMENTATION — It means dividing a market into different groups or categories according to certain criteria.

TARGETING — In this, marketers evaluate the attractiveness of each potential segment and decide which of these groups they will invest resources against to try to turn them into customers. The

UNDIFFERENTIATED MARKETING — An undifferentiated targeting strategy is one that appeals to a wide-spectrum of people.

POSITIONING — It is developing a product and brand image in the minds of consumers.

9.20 ANSWERS TO CHECK THE PROGRESS

I. True
II. Digital marketing
III. False
IV. Specific occasions
V. Targeted
VI. True
VII. Interruption-based marketing
VIII. Affiliate Marketing
IX. True
X. Blogging
XI. Technology
XII. True
XIII. Segmentation
XIV. Target
XV. Undifferentiated targeting
XVI. Concentrated strategy
XVII. Positioning
XVIII. True
XIX. Positioning
XX. True
XXI. True

9.21 TERMINAL QUESTIONS

- What is digital marketing strategy and why it is important?
- What is the importance of digital marketing strategies? What are the critical components of digital marketing?
- Highlight the important dimensions that must be considered by the marketers to make the digital marketing strategy more effective.
- What is Inbound marketing and how it is different from outbound marketing?

- Discuss the process for developing the effective digital marketing strategy.
- What is market segmentation, and why is it an important strategy in today’s marketplace?
- Explain the major variables used to segment the consumer markets. Give example of each.
- How do we segment the business markets?
- What is target marketing? What are the different levels of target marketing?
- What is market positioning? How the companies do position of their products? Discuss with examples.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-10 DIGITAL MARKETING MECHANISMS: WEBSITES- COMPANY AND RETAIL SERVICE PROVIDERS

STRUCTURE

10.0 OBJECTIVES

After studying the Unit, you would be able to

- Understand the marketing and offline marketing channels
- Basic understanding of digital marketing mechanism
- Comparison among goods and services
- Know the digital marketing strategies for retailing
- Elaborate the critical factors influencing the digital marketing mechanism
- Understand the electronic channels and its benefits

10.1 INTRODUCTION

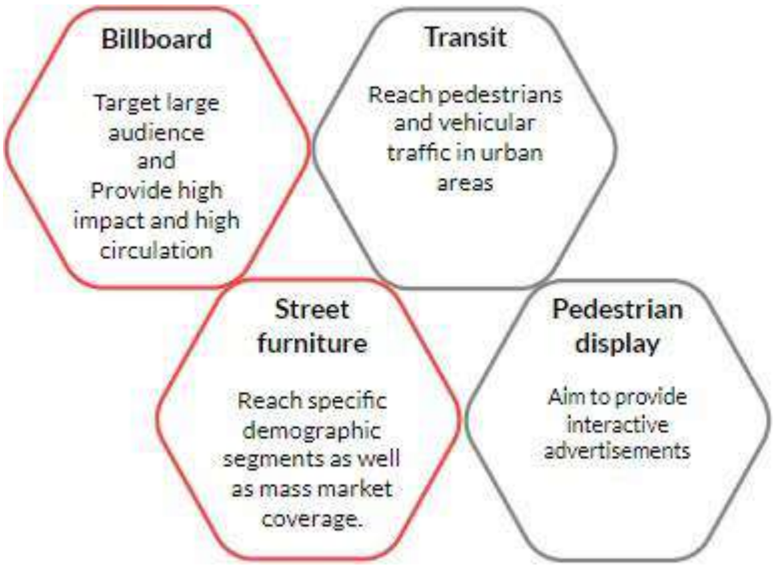
A business enterprise is a system that is created and managed by the people. The single main purpose of business is to create a satisfied customer. The business should always aim at creating and delivering values, to satisfy the customers. In order to understand clearly the business system, knowledge about marketing processes, systems and sub-systems are required. This will also help in knowing how the business and customers can be in advantageous position. From a general point of view, marketing is a practice of identifying and satisfying customer needs. With the explosion of digital technology, including the full use and spread of computers and smartphones, businesses are beginning to try new ways of marketing. Marketing is a process through which value is communicated to the consumer.

10.2 OFFLINE MARKETING CHANNELS

- **TV and Radio Advertising:** It is one of the heavily used channels for marketing due to its wider reach to the local and rural people. But doing this there are certain things which should be kept in mind so that our marketing campaign should reach to max target audiences;
  - ➢ Deciding the right media vehicle or TV channel for Ad.
  - ➢ Deciding the placement of ad in chosen vehicle i.e., time of Ad.
  - ➢ Deciding the frequency of Ad.

- **Print Media Advertising:** Print media advertising includes advertisements through magazines, newspapers and pamphlets. To get max of desired responses along with the factors which we should keep in mind like Tv or radio advertising we have to keep certain additional factors too in print media.
  - ➢ Along with keeping in mind the selection of right media vehicle the section of the newspaper or magazine where we have to place the ad should also be checked. And ad should be placed in the relevant section of product only
  - ➢ Days of week should also be considered while this for ad. There should be more ads of discounts or sale on sun days than any other days.

- **Outdoor Advertising:** Outdoor advertising is used to advertise about broad messages, branding and support various campaigns. Some of ways of outdoor Advertising are


Billboard
Target large audience and Provide high impact and high circulation
Transit
Reach pedestrians and vehicular traffic in urban areas
Street furniture
Reach specific demographic segments as well as mass market coverage.
Pedestrian display
Aim to provide interactive advertisements

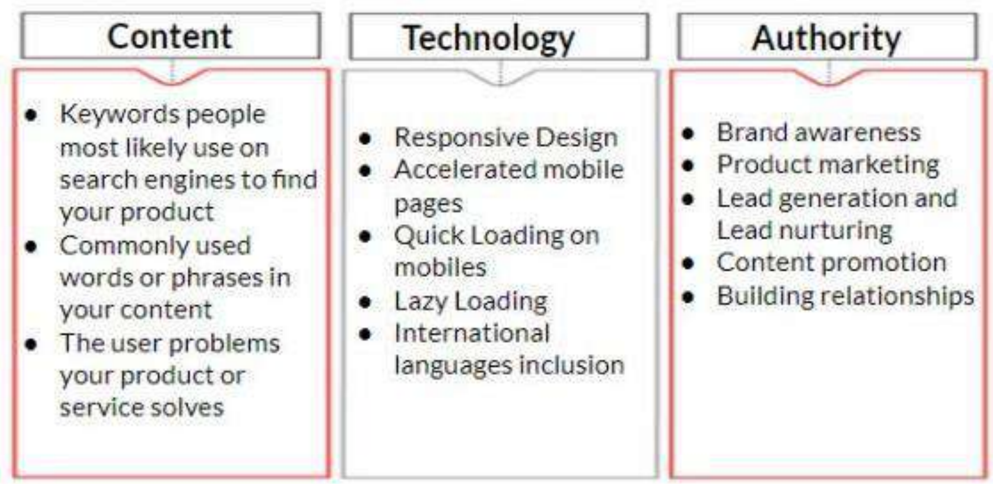
- **Event Advertising:** Event marketing helps to create more meaningful and long-lasting relationship with the customers. It involves presenting and advertising product via human commercials in a social gathering.

  Some of the areas where event advertising can be done:

  - Trade shows or social gatherings
  - Conferences which are majorly company specific marketing events.
  - On field events like of road shows, Converts etc.

10.3 DIGITAL MARKETING MECHANISMS

- **Search engine Optimization:** Search marketing strategies are used to increase any firm's presence online so that whenever a potential customer searches for their brand or a related service, their owned media assets are displayed at the top of search results. This can be done by improving the three factors-


Content
• Keywords people most likely use on search engines to find your product
• Commonly used words or phrases in your content
• The user problems your product or service solves
Technology
• Responsive Design
• Accelerated mobile pages
• Quick Loading on mobiles
• Lazy Loading
• International languages inclusion
Authority
• Brand awareness
• Product marketing
• Lead generation and Lead nurturing
• Content promotion
• Building relationships

- **E-Mail Marketing:** Emails are often used as a method by the companies to contact with their groups. E-mails are having a great role in brand awareness, Product marketing, Content promotion and lead generation. A normal person checks his email multiple times in a day. To build up retail presence, we can send him significant data about the thing is moving in the business. On the off chance that we send important data about their advantage individual would begin looking us as an asset and we can utilize this position to pitch items.

Effectiveness of an email campaign is calculated on the following parameters:

- ➢ Deliverability: Rate at which email reach to intended subscriber`s inbox
- ➢ Open Rate: Percentage of people that open email once it reaches to their inbox.
- ➢ CTR: Percentage of people that click on CTA’s.
- ➢ Unsubscribes: No. of people who opt out of email list.

- **Search Engine Marketing:** The major difference between SEO and SEM is this that SEM is a paid marketing strategy whereas SEO is an organic one. SEM is able to bring more visitor in a shorter period of time. SEM is also referred as Pay Per Click marketing. Main purpose of SEM is to:
    - ➢ Build Brands
    - ➢ Generate leads
    - ➢ Sell product

  Three major components of SEM are:

  - ➢ **Keywords:** These are the relevant words that shows on the web page in the first search result. Many of brands conduct keyword research and create campaigns that target best keywords.
  - ➢ **Ad text**: There should be customized text in the ad. Headline is the first thing that a person sees when he made a search. So, it should be attractive as it made customer click on the Ad.
  - ➢ **Landing Page**: It is the page where whole of traffic come from the campaign. So, this should be specific to the product and should use some of same keywords that are used in the Ad.
  - ➢ **Display Ads:** These are used to give general advertisements and brand messages to site visitors. Generally, are of three types:
    - ✓ Banner Ads: Appears on the side frame of the website.
    - ✓ Text only ads: Mostly seen in you tube videos they don’t have any graphics or link included in them.
    - ✓ Rich media ads: Combination of heavy graphics and images which provide interactive experience to the viewer.

- **Social Media Marketing:** It is one of the major forms of marketing which is going more and more popular these days due to presence of large number of youngsters on the platforms like of Facebook, Instagram and so on.

  Throughout the consumer decision-making process, social media marketing serves the following purpose:

  - Awareness stage: Increase brand awareness, increase reach and impressions, Build an online community.
  - Consideration stage: Engage users, get traffic to website, get users to fill the forms.
  - Purchase stage: Convert users into customers, increase revenue, reduce cost of acquiring new customer.
  - Delight stage: Manage and build online reputation, Provide customer service.

The above-mentioned techniques are the major ones for marketing digitally to any of website of a company. But along with them there are some of other techniques are there:

- **Content marketing**: Content Marketing is making significant and appropriate content dependably on stages. A brand can use content marketing as a gadget to achieve better brand care by displaying itself as an association that is in contact with and important to the world. For example, this is the way Nike is persistently running ad campaigns for athletes that are on the climb. The channels that can have an influence in substance advertising technique include:

  - Blog posting
  - E-book and white papers
  - Infographic

- **Affiliate Marketing** - This is a kind of Marketing based on publicizing where affiliate company will get commission for advancing another person's product or services on their site.

These are the marketing mechanisms with which we can digitally market website of any company and so on.

But along with all these when we came to retail service providers there can be some of other ways or mechanism with which they can market their product. These ways no doubt are digitals but are not online.

- **By Mobile or cell phone:** Suppose a retail store owner message his customers about the new price slash or new arrivals at his store. This is a digital way to market product but this involves no use of internet. So, it can be referred as digital but not online method.

CHECK YOUR PROGRESS

I. The business should not always aim at creating and delivering values, to satisfy the customers. (True/False)

II. ……….. media advertising includes advertisements through magazines, newspapers and pamphlets.

III. ……….. involves presenting and advertising product via human commercials in a social gathering.

a) Event Advertising
c) Email Advertising
b) Print Advertising
d) Social media Advertising

IV. ……………………. strategies are used to increase any firm's presence online.

a) Search engine optimization
c) Social Media Marketing
b) Email Marketing
d) All three

V. ………….. are having a great role in brand awareness, Product marketing, Content promotion and lead generation.

VI. ……………… is the rate at which email reach to intended subscriber`s inbox

VII. …………………. is the page where whole of traffic come from the campaign.

10.4 CHARACTERISTICS OF GOODS AND SERVICES

There are three basic attributes on the basis of which the differentiation of evaluation of goods and services take place. They are Search, Experience and Credence (SEC), explained below:

- **Search qualities:** This is a characteristic that can estimate before the purchase or consumption of a product. This is the quality on the basis of which some goods/services can be searched. Goods have a higher degree of these search qualities as compared to services. (a) For goods it can be all physical attributes like, price, color, design, looks, style, shape, size, etc. So, they are easy to search. (b) For services this can be price, convenience, presentation, promptness of service, courtesy, mannerism, ambiance, etc. Here these are not easy to search, so other factors like tangibility are used for help in searching.

- **Experience qualities:** The quality of goods and services are those which can only be assessed after purchase, utilization, use or consumption, i.e., after the experience. Services have higher experience attributes than goods.
- **Credence qualities:** There are certain attribute of any goods or services that can’t be assessed even after use/utilization/consumption. Because the customers lack certain ability to analyse or estimate them, like the effect of some nutritious food or vitamin. Some health service provider may claim that the food contains organically produced fertilizers, but there is no way to ascertain it. But, however, when services are offered by experts, professionals, specialists, it creates credence for services

10.5 DIGITAL MARKETING STRATEGIES FOR RETAIL SECTOR

Digital market has made his space in every field, due to its vast advantages. Every sector of industry is adopting digital methods of marketing. Following are the few online marketing strategies for retail service provider:

- **Using Facebook Ads to get more revenue:** In the retail industry it is one of the major strategies to generate more revenues. There are three stages of running a successful Facebook ad which will add up in resume:
  - ➢ To have a start, start with engagement campaigns.
  - ➢ Create a sense of urgency to the customers, by putting some of discounts on visits on particular time or a particular day.
  - ➢ This will dive more traffic to store.
- **Google shopping campaigns:** Google owns 92% of search engine market. Google Shopping ads are the main thing clients see, and they are additionally more unique than customary Google advertisements. Google Shopping highlights item pictures,

valuing, dispatching data, and star evaluations. The genuine stunt is to utilize Google Shopping efforts not withstanding other Google advertising systems.

- **Re-target customers:** It was proven that only 2% of the customer which visit website will purchase. There are such countless reasons why a client didn't choose to purchase on their first visit. They may have had more inquiries they required addressed first. Possibly they are holding for a very decent arrangement. Perhaps they outrageously need your item, yet it is not at the first spot on their list at this moment, and they were taking a gander at your site like window customers do in a shopping center.

  Retargeting ads should work to solve those potential points. For example, if it's the price, then an extra discount will sway them.

  Two main types of retargeting are:

  - Using email lists
  - Using a Facebook pixel on selected pages.

- **Use Influencer marketing for unique and value driven content:** Influencer showcasing is intended to be explicit, relatable, and individual. As clients need more pertinent substance, it's not difficult to perceive any reason why 65% of impact promoting spending plans are required to increment in 2022.

  Discovering credible influencers to advance the image, permits to acquire trust and enlist new clients. Instagram has been the home for most influencers. Instagram's expansiveness of client created content permits to discover influencers who are normal fits for vertical.

- **Buy online and pick up in store:** It is one of the most common method used by every big retailer as of the name suggests that a user or customer can buy online from the website of the retailer and can collect his or her item from the store. This allows store owner to advertise his products at two different platforms and by this he can pitch more product to the customer as customer has to visit store too for collecting the items. Store owner can track the activity of consumer on the web page and can give more discount on the product which customer has viewed but not purchased.

10.6 FACTORS FOR DECIDING THE RIGHT MARKETING MECHANISM

- **Goals:** Your choice of digital marketing channels depends on the goals you want to achieve. The goals can be: Brand awareness, creating new leads, improving online search ranking, increasing website traffic, improving conversation rate.
- **Budget:** Marketing costs contrast from one channel to another. Some offer apparatuses that you could use free of charge, while others must be utilized for an expense. All things considered, some others like paid hunt fall some place in the middle with execution showcasing benefits, where you possibly pay when somebody clicks your promotion.
- **Target Audience:** One approach to target explicit personas is to know on which channels they are generally dynamic or locked in. With this data, it is likewise simpler to think of channel that will target your crowd.
- **Competitors:** On the off chance that your rival is offering to clients through a specific advertising channel, you can either go a similar course and exploit that channel’s current market or look the alternate route where there is less rivalry, in spite of the fact that you shall have to consider as well if there's sufficient chance for you to sell in less mainstream channels.
- **Functionality:** Contingent upon the size or nature of your business, you could conceivably require each one of those pleasant to-have highlights that a specific advanced advertising channel offers. Consider beginning with essential highlights and afterward increasing things once you see they're worth putting resources into.

CHECK YOUR PROGRESS

VIII. ……………….. are certain attribute of any goods or services that can’t be assessed even after use/utilization/consumption.

IX. Google owns 92% of search engine market. (True/False)

X. In the retail industry ……………… is one of the major strategies to generate more revenues.

XI. It has proven that only 2% of the customer which visit website will purchase. (True/False)

XII. Discovering credible influencers to advance the image, permits to acquire trust and enlist new clients. (True/False)

XIII. ………………… has been the home for most influencers.

a) Instagram

b) Facebook

c) LinkedIn

d) Snapchat

10.7 ELECTRONIC CHANNELS

Electronic channels are becoming more and more popular day by day as there is no direct man to man interaction. Internet, e-commerce, telephone, television, etc. are the examples of electronic channels. For retailers, the internet has become the main commercial medium. Many efficient distribution services are provided by the internet. Industries such as financial services, software, data processing and economy dominated by the electronic activities distance is not important. The service firms provide movies on demand, banking and financial services, multimedia libraries, database, video conferencing facilities, etc. through electronic channels. The businesses and consumers are able to exchange the not only information, but also different kinds of products and services through internet. The World Wide Web has provided many opportunities in speed, price and comparability to meet the expectations of the consumers. Due to the global market, service companies have to improve the quality constantly to be in competition. Technology has become important as well as it links the service providers, distributors and the consumers. As compared to the traditional methods, the electronic distribution will gain a big market share in the future. Multichannel retailers and the pure players share the online shopping market share. Pure players are the companies which sell the products online and do not have stores. Multichannel retailers are those which supplement the conventional stores with the online services.

10.8 BENEFITS OF ELECTRONIC CHANNELS

- **Quality Control:** Through the electronic channels the service organisations can reach the consumers without any change in the standard package of the services. The electronic channels do not interpret the services. Whatever is designed is transmitted to the consumers. The centralized design of the service package is facilitated by the electronic channel which leads to better quality control.
- **Cost:** As compared to the human being the cost of distribution through the electronic channel to reach each consumer is low.
- **Customer Convenience:** The most important benefit of the electronic channel is convenience to the consumers. Through the electronic media the consumers can access whatever and wherever consumers want the service.

- **Distribution:** Through the electronic channels cost effective market coverage is possible.
- **Customer choice:** A wide variety of the services can be offered to the consumers, according to the choice of the consumers. Movies, cables programmes, etc. are some of the electronic channels offered by the service organisations.

10.9 THE DESIGN OF A SERVICE DISTRIBUTION SYSTEM

While designing the electronic channel it should be kept in mind that the execution of the system should be effective and efficient. If the execution is poor, even the strong and sound system will lead to negative or poor results. Distribution can be used as a strong weapon if it is designed properly. Following are the points which should be kept in mind while designing the distribution system.

- **Align the system properly:** Service principles, intermediaries and customers are involved in the indirect distribution system. The service principles design the business system planning which has many dimensions. In the process which involves service production, delivery and consumption, the dimensions such as personal and business and the customers personal and social dimensions play a very important role. To make the distribution system work effectively there should be proper alignment of the dimensions, intermediaries and the customers. The goals of the distribution system and the organisation goals should match.
- **Remember each part of the sales cycle:** “Before sales”, “during sales” and “after sales” form the sales cycle of the organisation. “Before the sale” there is initial contact between the service sales employee and the customer. Service organisations should see that the location of the services is convenient to the consumers. This stage of the sales is mainly affected by the design of the service process and promotion methods used by the service organisations. “During the sale”, the service firms try to facilitate the customer participation. During this stage of the sales cycle, the service firms should focus on the quality performance and they should try to perform right at the first time. In the last stage, i.e., “after the sale” the service organisations try to retain the customers through the customer relationships, customer complaint handling, service recovery strategies. If all the three stages of the service sale cycle are

performed effectively and efficiently, these will have a positive impact on the quality perception of the consumers.

- **Balance retail and wholesale intermediaries:** The functions of the retailers and the wholesalers are different. Retailers come in direct contact with the consumers whereas the wholesalers are intermediaries between the producer and the retailers. So, both the retailers and the wholesalers require different skills. There should be a balance between the activities of the wholesalers and the retailers. The activities of both of them should be clear. For the efficient and effective working of the service firms, there should be a balance and coordination between the wholesalers and the retailers.

CHECK YOUR PROGRESS

XIV. Electronic channels are becoming more and more popular day by day as there is no direct man to man interaction. (True/False)

XV. The …………….. has provided many opportunities in speed, price and comparability to meet the expectations of the consumers.

XVI. The centralized design of the service package is facilitated by the electronic channel which leads to better quality control. (True/False)

XVII. …………………. can be used as a strong weapon if it is designed properly.

10.10 LET US SUM UP

In the era of digital economy Internet marketing strategies are the one that promote effectiveness and success of companies on the market. It is a fact that companies which nowadays are not present online are literally invisible for the consumers. And here comes the digital marketing. Well, marketing has been a part of our system since ages. In the earlier times too, people use to market their products by announcements and with the invention of printing press marketing becomes cheaper and easy as now one has to just print poster. With the passage of time marketing techniques has also been evolved. With invention of Tv and Radio era of advertisements came in and when social media gain its advantage there has been a numerous no. of new ways of advertising., These days ads put a vast impact on every person`s purchasing decision. These are ads only who create awareness about the product and tell its features. After creating awareness, they try to give a sense of urgency to the customer pitch their product. With the use of digital media, it has become very easy to figure out that which method of advertising works for one and which is not working. As we can track each and every visitor coming to our site from where he is coming what he is looking for what he is purchasing and what he is looking but not purchasing. Digital marketing has given a significant insight to the companies that what planes should be made to have maximum no of sales and how we can increase our sales and can left our competitor far behind. As traditional methods of advertising were way costlier and more inefficient than the digital ones. So, with the passage of time they are losing their importance.

10.11 KEY WORDS

PRINT MEDIA ADVERTISING	It includes advertisements through magazines, newspapers and pamphlets.
EVENT ADVERTISING	It involves presenting and advertising product via human commercials in a social gathering.
SEARCH ENGINE OPTIMIZATION	Search marketing strategies are used to increase any firm's presence online.
SEARCH ENGINE MARKETING	SEM is able to bring more visitor in a shorter period of time.
LANDING PAGE	It is the page where whole of traffic come from the campaign.
AFFILIATE MARKETING	This is a kind of Marketing based on publicizing where affiliate company will get commission for advancing another person's product or services on their site
CREDENCE QUALITIES	Attribute of any goods or services that can't be assessed even after use/utilization/consumption.

10.12 ANSWERS TO CHECK THE PROGRESS

I. False
II. Print
III. Event Advertising
IV. Search engine optimization
V. E-mails
VI. Deliverability
VII. Landing Page
VIII. Credence qualities
IX. True
X. Facebook ad
XI. True
XII. True
XIII. Instagram
XIV. True

XV. World Wide Web XVI. True XVII. Distribution

10.13 TERMINAL QUESTIONS

- What is digital marketing mechanism? What are the off-line marketing channels?
- Describe the digital marketing mechanism and discuss its channels.
- What is the role of product in digital marketing mechanism? Describe the characteristic of good and services.
- What are the digital marketing strategies for the retailing?
- Define the digital marketing mechanism. What are the factors important for effective digital marketing mechanism?
- How would you define the electronic channels and how they are beneficial for the business organizations?

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-11 SEARCH ENGINES- GOOGLE, BING, YAHOO, VIDEO HOSTING AND ENTERTAINMENT- YOUTUBE, AMAZON PRIME, NETFLIX, HOTSTAR

STRUCTURE

11.0 OBJECTIVES

After studying the Unit, you would be able to

- Learn how Search Engine work
- Understand the Google as Marketing tool
- Have knowledge about Google+, Bing, and Yahoo as marketing tool

- Explore video marketing and its tools
- Highlight the different platforms and strategies for video marketing
- Learn about YouTube and why customers like it most
- Understand the Video Hosting and its platforms
- Importance of Entertainment for customers and how digital marketing involve in it
- Have basic understanding for Netflix, Amazon Prime and Hotstar

11.1 SEARCH ENGINE

Searching is one of the most used moves at the internet. Search engines like Google as a tool, are very famous and mostly used websites. Website online proprietors use main search engines like Google and yahoo for filing their sites on it, and for searching. Regular users use number one search engines like Google essentially for looking, and every now and then for filing their Webpages.

Search engine help the customers to find out the required information to the take the purchase decision. In the search engine either we talk about Google, Yahoo or any other search engine the show the results of website of companies. The sequencing of the website all would depends upon the algorithm, SEO and SEM. The customers can visit the companies' website as per the click they would have on the given links on the search engine results pages. The companies have lot of paid and unpaid strategies to bring their companies' website at the top on the search engine research page.

In order to become expertise in SEO and SEM, at beginner stage you ought to learn how to use engines like Google for looking the information. Individual should know that there are two methods of looking: by handle ding customer questionnaire or by using the resources of class. If individual have keyword to describe about their work art than it will be easily to understand or learn about it.

11.2 GOOGLE- MEDIUM OF MARKETING

Marketing on google is done through the digital marketing tools used for search engine marketing, such as adwards, google analytics. Google helps to market the products through the google adwards and google analytics by payperclick to the companies who are advertisers. Google marketing strategy is developed on the product qyality and usability but there are also other espects effects the marketing strategy.

11.3 HOW TO ADVERTISE ON GOOGLE

- **Create a account on google**: First of all have to create an account on google and log in.
- **Setting of budget**: After logging into google ads, have to set daily budget for ads. It may be increase or decrease.
- **Set a location:** later have to set location where we want to taget the market/customers.
- **Choose a network :** After that the next step is to choose a network.
- **Select keywords:** Keywords are very important in google marketing. Before any advertisement, we have to choose the proper and attractive keywords that's related to business and product.
- **Create ads:** The last step would be to create and cast it.

11.4 GOOGLE TOOLS FOR MARKETING

In every second, there are around 40,000 quires o the google. So google search is the backbone of google bussinesses and processors. It is a powerfull and effective search engine , where a lot of people search information on the daily bases. They also provide various types of tools to the developers and marketers to develop their business.

There are the basic uses of google marketing:

1. Improves SEO Ranking.
2. Build Brand awareness
3. Attract and engaged many people with brand.
4. Maintain and create brand reputation.

11.5 GOOGLE +

Google+ is a powerfull marketing tool as compare to other social networks. We can share informations, images and videos. It is a tool where marketers promote their goods and services and connect with other individuals and make good relationships with them. If a marketer creates an attractive profile and pages of their business it attracts the new peoples and companies.

Google+ is Google's own social network, enabling brands to build relationships with prospects, customers, and other businesses. Similar to other networks, you can share information and images, promote yourself as an individual and if, used correctly.

- Google+ is a powerful search engine marketing tool for companies, influencers, and your own personal branding.
- Google can be a powerful social media marketing tool.
- The importance of Google+ should be assessed on an individual basis, as it is based on your unique appeal. If is thought to be of benefit to your brand, it should be used as part of your overarching social media strategy.

So, Google+ is Google's flagship social network, allowing users to connect easily and often, through interests and friendship. It's not an independent service, instead, it is integrated and connected through all Google services as part of the ecosystem.

Interactions with individuals or companies are based on profiles or pages. Understanding google+ in many respects, Google+ is more of an `information network` than a social network.

11.6 BING

Under the leadership of the Ballmer company, Microsoft try to design new products or a new service and in this try they make a software Bing in 2009 they started this search engine Bing. Bing come with new features of software and also helps in marketing.

11.7 BING MARKETING

Bing has been used for advertising and promotion of goods and services by the business organisations. Markets have been taking the help from the Bing to promote their products and customers have been getting the information regarding the products and services.

Benefits for advertisers

- Bing face less competition
- Bing is economical than other

- Provide more and more options
- Help to tackle target consumers
- Provide control facilities
- Provide more transparency in the process
- Provide information about the demographic
- Handle many of consumers

CHECK YOUR PROGRESS

I. ……………. help the customers to find out the required information to the take the purchase decision.

a. Emails
b. Blogs
c. Search Engine
d. All three

II. Website online proprietors use main search engines like Google and yahoo for filing their sites on it, and for searching. (True/False)

III. YouTube helps to market the products through the google adwards and google analytics by payperclick to the companies who are advertisers. (True/False)

IV. …………… is a powerfull and effective search engine , where a lot of people search information on the daily bases.

V. Google+ is Google’s own social network, enabling brands to build relationships with prospects, customers, and other businesses.

a. Blogging
b. YouTube
c. Google +
d. Search Engine

VI. Bing has been used for advertising and promotion of goods and services by the business organisations. (True/False)

11.8 YAHOO

Yahoo is a brand with related to research, predicts, crate and associates in the motion brand shows the new achievements and the way of doing work and yahoo logo is made from wordmark. This is organized as the dictionary and the web index which help to search by most of people in march 1994 the web name as jerry and David ‘s Guide to the world wide web as the name Yahoo.

Later this is known as Yahoo dictionary and also the domain name is set for yahoo is the Yahoo.com

- In yahoo, advertisers can make the small chat group for the advertising.
- In yahoo can find best types of keyboard .
- Yahoo is automatically rolling the ads and advertisers can select the best platform for there advertisement.
- Yahoo search marketing is used by most of advertisers.
- Send the users to related pages and help to optimize website.

11.9 PRODUCT AND SERVICES OF YAHOO

- Yahoo! home page - This is the main page is main home page
- My Yahoo!
- Yahoo! Auctions
- Yahoo! Developer Network
- Yahoo! Entertainment
- Yahoo! Lifestyle
- Yahoo! Mail   This is mail developed by the yahoo and there is for type of mail developed by yahoo
- Yahoo! Maktoob
- Yahoo Mobile
- Yahoo Movies
- Yahoo! News
- Yahoo! Research
- Yahoo! Smart TV
- Yahoo Sports
- Rivals.com
- Yahoo! Gemini,
- Yahoo! Search

- Yahoo! Shopping
- Yahoo! Small Business
- Yahoo search
- Yahoo video

11.10 VIDEO MARKETING

Video marketing is a part of marketing, by the means of which a company communicate the features of its product and services to its potential consumers. The basic objective of marketing by video is to convert the potential consumers to real time consumers.

A video is a representation of visual images with background music. Videos can consist of different pictures which changes with the frame per second. It is said that a picture says thousand words. Video is considered as the most effective Way to grab attention of the people. We humans, remember the thing visually more than remembering them by listening or talking. Companies promote their product or services by adding the information about their product in tits and bits of the video. A video could engage the consumer in a story and promote its product through it. Many companies try to hit the audience on the emotional part and try to connect to them.

11.11 EVOLUTION OF VIDEO MARKETING

Before the evolution of web, there were very limited options for the companies to introduce their product in the market. Companies used to beg the mainstream media that were television channels to play their advertisements on it and tell the story to the customers on the behalf of company. But now, with the evolution of internet, there are plenty of options to showcase the product promotional videos.

11.12 TOOLS OF VIDEO MARKETING

There are a lot of video marketing software which helps a manager to create a perfect marketing video for his product and services.

- **Flipagram:** Flipagram is a video editing software which combines the images to create short video stories. It is very easy to use and perfect for those videos which are going to be posted on Instagram and Facebook. This tool is available on the different operating software like play store and apple app store since 2014 for free. It is one of the popular editing apps.

- **Stop Motion:** Stop Motion is really instinctive and will help to make amazing videos with this technique. This application is available for iOS and has features of green screen, image import and sound effect. This application can be used on phone as well as on laptops. This is a paid application, which helps to edit the video frame by frame while watching. It contains the new feature of thumb menu to control all the editing options available for each frame.
- **YouTube Editor:** This interesting tool is integrated by YouTube, which is available for free. It contains the interesting editing capabilities. Transcripts and annotations can be added to the videos to make them look interesting and search engine optimization friendly. It makes the editing easy as well as promotes the YouTube as a platform for uploading videos.
- **Animoto:** It is one of easiest apps to use for video editing. One can create a sunning video in just three simple steps:
    - ➢ Add pictures and videos
    - ➢ Customize video by adding music, style, text and other features of choice
    - ➢ Upload it to website

  In this video is ready in just few minutes and the application works for both editing and sharing videos.
- **WeVideo:** WeVideo is the web-based video editor which helps to create digital stories though still photographs. It also allows to voice over during the video plays. One can crop pictures, add text to the slides and edit audio files with the help of this app. Moreover, this application is available on mobile, so that you can get your videos edited on the go.
- **PowToons:** PowToon is free application for creating animated videos and presentations, which can be showcased on website and can be shared on social media handles. This app provides wide range of templates, characters, backgrounds and transition effects. It allows the user to create its own template from scratch.
- **Picovico:** Picovico is an amazing online tool that automatically creates the slideshow of the photos and videos, in the matter of few seconds. The main exciting features are voice-over, video clips, and pre-defined video templates. The videos created with the help of this app can be exported to any device and allows to upload directly to social

media handles. It offers videos for professional and well personal use and API option for heavy size videos.

- **Wideo:** Wideo is easy and fun to use as it offers the selection of templates, which can be customize to deliver message or just create own template from scratch. Moreover, you can create your own logos and add them into videos to make it more interesting. This app is also available in a lite edition as well.
- **Stupeflix:** Stupeflix is a great video editor around for mixing photographs, videos and music. It features wide selection of beautiful themes, which can be used for making the video making process quicker. Photos, videos, as well as text, a soundtrack, and even maps could be inserted with the help of this app. There is feature where two people can voice-over the video at the same time. After completing the video, you can post your videos on YouTube or on website and can download it. Stupeflix is a website which also available in the form of an app.
- **Sellamations:** There are very few applications and software which offers the animation feature and sellamations is one of those. It would be the best to choose this out of others. Doodle animations which are very attractive and fun to watch, can engage the viewers, can be done in this software. One can create fully customize specially for your brand and add that to your video. This software beneficial for professional voice-overs and script writing.

11.13 PLATFORMS FOR VIDEO MARKETING

Now, in the era of digitalization there are billion option to post videos, all that can become the way for promotion and marketing of a product by the means of video.

- **YouTube**: It is one of the most used app now a days. This platform provides the opportunity to share videos for free. Brands create there channels and post the videos for advertising there product and services. YouTube creates a lot of public engagement for brands. For example, t-series which is a music company has account on YouTube, they continuously keep posting the new songs on YouTube and made its mark with having most subscribers all across the world. YouTube has created a lot value in similar way for many brands. There is an another way marketing is done on YouTube which is, the youtubers who post videos regularly and become famous, brands approach them to use their product and review them in there videos. This encourage the subscribers of that particular youtuber to buy your product and lend your services.

- **Instagram:** There may not anyone to not have an Instagram accounts. This is one of the most used app all over the world. The most common way to promote brand or product is to ask brand ambassador to make a video while using the product and post that on their official accounts. It helps the audience to get products and services and boost up sales. The other way is to make an account of brand and post videos there. It will increase the brand value and more people will get to know about product.
- **TickTok**: This is one of most popular apps among the people. Specially, teenagers really enjoy making videos on it. This application offers to make videos while using the background music. Brand have find their way to get the best use of this for marketing. They approach the creators who have a large number of followers and ask them to make the videos on the official sound tracks of the brand in an interesting way. They are asked to make a particular hashtag which is to be written in the caption of the video. When people make videos on the same track inspired from famous creators, the hashtag comes in the top used trending lists. This helps to gain lot of new customers to the brands and add on to the value and goodwill of the brand. For example, recently Coca-Cola made a new track for its video and it was in trending on TickTok also.
- **Television:** It has been the most traditional platforms for advertising the products. But, it have been working good even after the evolution of wide range of social media apps. Primarily, videos were made only to be played on television screens for promoting the products and services.

11.14 VIDEO MARKETING STRATEGY

There are some points which one need to take account of while making video for marketing purpose.

- **Allocation of resources:** For making the video, planning is going to be the first step. We need to plan and to set a budget for it. Things at least required are a decent equipment, a good editing software, a marketing agent for a good team and most important element, time to create it.
- **Storytelling:** Storytelling is the most important element for making the video. Brainstorm the ideas to decide what story you want to tell, the way it is going to be

told and how you will show product in that story. Select the most suitable story that shows the best of your product.

- **Audience engagement:** Telling a great story is not the only that you need to do, you must engage audiences while you do so. There must be some elements in video which could hook the customer throughout the video and to make them watch it till the end.
- **Duration of the video:** There is not a fix duration for the marketing videos although there are recommendations for it. But generally, it is believed that shorter the video is, better it is. Be very carefully will editing, cut everything unnecessary out and keep the major content in such a way that whole attention of the customer goes towards the product. Attention spans are short, so make the best that can be made out of it.
- **Publishing the video:** Use the wide range of platforms to upload videos and to promote product and services. The different platforms for video marketing are YouTube, whatsapp, Instagram, Facebook etc. The most effective way that could be possible to make video reach the most number of people, when the brand ask its brand ambassador to upload the video on their official accounts.
- **Analysis:** Track the statistics of video and to determine that when the video perform at its best. Identify the reasons for its best performance, so that you can add those elements in the future promotional projects.

CHECK YOUR PROGRESS

VII. ………………. is a part of marketing, by the means of which a company communicate the features of its product and services to its potential consumers.

VIII. …………….. is considered as the most effective Way to grab attention of the people.

a. Email
b. Blog
c. Video
d. All three

IX. …………….. is a video editing software which combines the images to create short video stories.

a. WeVideo
b. Animoto
c. Flipagram
d. All three

X. ……………. is available for iOS and has features of green screen, image import and sound effect.

XI. …………….. is the web-based video editor which helps to create digital stories though still photographs.

a. WeVideo
c. Flipagram
b. Animoto
d. All three

XII. Under……………. the most common way to promote brand or product is to ask brand ambassador to make a video while using the product and post that on their official accounts.

11.15 YOUTUBE

As we all know YouTube is a video sharing service where we can access to various videos. People can watch, share, like and download videos from YouTube. We can access videos through smart phones, PCs, laptops etc.

For teenagers it is an entertaining and learning source as well. As a teen, we can discover things which we like. For young generation, it is a source for watching movies, comedy shows, life hacks as well as a good source of education. Users can also create their personal YouTube channel and subscribe the channel which they want. As it is a video sharing website, that makes it easy to watch online videos.

YouTube was started by three employees of PayPal online payment service. They are Jawed karim, Chad Hurley and Steve Chen. YouTube was first launched in the year 2005. It gained so much popularity very fast and has now become one of the most visited websites in the history of internet. According to one of the data, it is seen that visitors watch around 6 billion hours of data every month.

We can find all kinds of videos on YouTube. There is always something new to watch on YouTube. Google has purchased YouTube in 1.65 million dollars. Why you should use it, depends on how you check out the site and see what you can get out of it.

YouTube is a video platform that is driven by two types of users:

- **Video Creator:** Video Creators are those people who have their YouTube channels and upload videos for viewers.
- **Video Viewers:** Video Viewers are those people who watch videos, interact videos and subscribe to channels.

You can be both a creator and viewer. YouTube is for everyone, whether you are an individual who is looking for entertainment stuff/creative stuff or CEO of an organization with a large budget for a video ad campaign.

Although it's user base ranges from young to older generation. There are almost 50 different languages on YouTube and a wide variety of content. Since it is owned by Google, all you need is a Google account to access you tube.

11.16 A BRIEF HISTORY OF YOUTUBE

In 2005, a content and photo sharing sites were more, the founder of YouTube noticed a problem that there was an explosion in the number of expensive and inexpensive ways to capture videos but there wasn't a good way to share videos.

On 14th February 2005, Hurley registered the trade mark, logo and domain of you tube. Three months later, in May 2005, a small group of early employees launched a site www.youtube.com. In November 2005, Sequoia capital gave funding to you tube. In December, YouTube officially become a corporation.

The first office of YouTube was located above a pizzeria and Japanese restaurant in San Mateo, California. Over a tine, YouTube has become popular, its employees increased which led to formation of chain. In October 2006, Google came knocking at the door. At the time YouTube had 65 employees. On the day, the acquisition was announced YouTube moved into a former GAP Office in San Bruno, California. The much-discussed deal was settled one month later.

There wouldn't be a YouTube without PayPal. YouTube was created by three former PayPal employees, staffed by other employees of the company and bankrolled by the people who used to work at PayPal which was launched in 1998.

All knowledge which Karim got from PayPal was summed up in two words "STAY FLEXIBLE". This mantra has guide YouTube to gain success and popularity in the world.

Today, YouTube is the largest online video destination in the world and third most visited website overall. This site almost got 2 billion views a day. Nearly about the prime-time audience of all three majors US network combined. This platform comprises the largest video sharing community in the world and includes users, advertisers and over 10,000 partners.

11.17 WHY AUDIENCE SHOULD USE YOUTUBE

There is wide variety of videos on YouTube. You can either watch videos or create videos. It is a source of entertainment, learning for the people who want to start their own business. It is the cheapest way for advertising. When it comes to marketing, we can easily give information about product and services to the users. YouTube is a great tool for getting your product in market in very low cost.

For singers and the person who creates music, this is the best way to reach the public. Earlier it was really very difficult to show talent to the public but YouTube made it easy in very little cost.

When a music creator creates a video with the help of public reviews and comments, he can easily know what public think about the video and can make changes as per public demand. Audience can also subscribe channels for regular update. So that whenever a person uploads any video, audiences get a notification of that. So, when you put a new video song on your channel, the people who are interested in your music can follow.

We can not only watch videos in fact it is also very easy to download video from YouTube. It is relatively easy and fast but depends on internet connection.

For watching videos all you need is to search for a particular video which we want to watch. We can easily access to the video. When it comes to the trending videos it is on the top of the you tube, so we can easily get updated about the trending videos on YouTube.

Video files are often very large and it takes time to send the whole video but with the help of YouTube we can easily share a video simply by sending other person a URL link, that is the address of the relevant internet page.

Nowadays, the term ‘Viral Video’ is common. Viral videos are the video clips that people have liked and shared too much by e-mails with millions of people around the world.

Companies have realized that they can harness this ability and can easily reach potential customers. So, companies started creating their own YouTube accounts for posting advertisement and other marketing video.

11.18 HOW TO KEEP AUDIENCE AWAKE

For any video creator the most important task is to keep the audience awake. Probably many of the users who are watching videos are tired. So, it is very much important for a you-tuber to help them stay awake, to actually pay attention and consider what they are saying.

First of all, tell your audience what they will get from your talk, so that the users time can be saved. We can use various strategies for this purpose:

- SPEAK LESS THAN THE TIME ALLOTED: When a you-tuber present any video, it is necessary that he gives great information in very short period of time. To attract the audience you can say that "I know I have total time of 20 minutes but I will only talk for 10 minutes. This will surely attract the audience and make them happy not to have to listen as long as they expect.
- USE SILENCE EFFECTIVELY: When we make a statement we have to wait in silence to see how people receive it. It is necessary for the user to use silence effectively. Don't throw more words and statement without giving your audience the chance to understand each sentence.
- EMPHASIZE KEY WORDS: It is necessary that the speaker should not speak in the same tone throughout the entire presentation because due to this no one understands what is really important. Make it obvious to your audience that what they really need to pay attention to.
- USE NUMBERS AND EMPHASIZE THEM: A person can pay much more attention when we use number strategies to solve this situation like "Number 1…., Number 2…." Every time when we say a number, it reengages audience's attention and helps their brain to listen.
- ADD SOME EMOTION OR HUMOR TO YOUR TALK: To engage audience and make topic interesting, we can use emotional words like "I am excited today to be here to tell some good news" etc.
- DO SOMETHING UNEXPECTED: In every video, do something unexpected so that it make the interest of the users. Like you can ask from the audience to give their views on the topic. This will surely change the energy. People start talking and come to some understanding before he went on.

11.19 VIDEO HOSTING

Video hosting platform is also known as video sharing platform. It provides a platform for businesses and individuals to share video. The video hosting platform handles video streaming and playback, providing users with a seamless video experience. It also helps to increase the traffic to the website. In addition to these advantages, a good hosting platform offers: Content security, Thumbnail management, adding subtitles, Basic analysis So far, everything looks good. Video hosting platforms allow to post videos online, increase engagement and provide excellent visibility to content. Therefore, we can customize video player (using a hosting platform), customize video, and understand audience.

11.20 VIDEO HOSTING PLATFORMS

- YouTube
- Vimeo
- Amazon Prime
- Netflix
- Hotstar
- Facebook
- Hippo Video
- Moj

11.21 ADVANTAGES OF VIDEO HOSTING

The biggest advantage of online video hosting is that videos can be shared in multiple locations and managed from one control panel at the same time. This means you can get the best experience with the least amount of effort and share it widely. This will help you create more videos and achieve better results. Here are some of the benefits of video hosting platforms: Great viewing experience. You can sharpen the video hosting image (adapted to your device) and control the video player. Time-saving tool. Hosting services can do a lot to automatically download videos. If necessary, you have the option to compress the video and automatically encode it for viewing on other devices. Otherwise, you have to remember that Windows Explorer requires H.264 codec and Google Chrome requires Webm codec. Video marketing is more effective. If you can manage all your videos

- **Save time with integration:** The hosting platform can be integrated with other apps, so you can target videos to different channels. You can also edit them after posting, or download them quickly if needed.
- **High availability:** The hosting platform plays videos in data centres around the world, so you can access them simultaneously in multiple countries. Hosting offers automatic backups. When a site goes down, the video doesn't go anywhere else.

11.22 ENTERTAINMENT

In today's fast-paced Internet world, entertainment is the most stressful part of everyday life. Entertainment falls into several categories such as television, films, music, animation, games, etc. The biggest challenge in the media and entertainment industry is attracting viewers and increasing fan base. Increased use of smartphones and lower costs of accessing digital data have led to increased competition. Today, billions of people have access to social channels like YouTube, Facebook, Twitter, Instagram, Pinterest and more. It makes it easy to promote movies, music albums and short films.

11.23 USEAGE OF DIGITAL MARKETING TO PROMOTE THE ENTERTAINMENT INDUSTRY

YouTube and social media are the largest platforms helping to advance the entertainment industry. Let's take a closer look.

- YouTube Marketing: YouTube is the second largest search engine reaching millions of people making huge profits from this platform. You are more likely to get into the 18-49 age group using mobile devices than television. It is the largest platform in the entertainment industry to instantly receive updates from people and connect with them via video.
    - ➢ Put your video in the best place when people are looking for video content.
    - ➢ Place multimedia and entertainment ads alongside other popular videos for your category. Improving YouTube video views and sharing.
    - ➢ Increase the likes and subscribers of your YouTube channel.
    - ➢ Build your YouTube channel brand.
- Social media marketing: through more forms of entertainment channels.

11.24 BENEFITS OF DIGITAL MARKETING TO THE ENTERTAINMENT INDUSTRY

Digital marketing is growing rapidly and many industries have benefited from it. Now, let’s take a look at the biggest benefits of digital marketing in the entertainment industry.

- Reach a large audience.
- High-yeld investment program (HYIP) for upcoming entertainment and media updates.
- Find the best followers for your entertainment fans.
- Make a good impression of your project.
- Encourage the audience to participate in viewing the media.

11.25 AMAZON PRIME, NETFLIX AND HOTSTAR

Netflix and Amazon Prime’s core business strategy is that users will have to browse both apps and continue to pay more for a membership. The competition is intensifying and it is up to someone to decide which application to choose and why. When people can make payment transactions online and they become available, they will hesitate when they read both reviews, which means that the two applications will vibrate and create strong competition with each other.

Amazon Prime Video and Netflix are two of the most popular streaming services in the United States. The more people are interested in streaming online, the more competitive the market will be. This confusion has led many consumers to wonder which of the two players in the code-shortening industry is better. Netflix v/s Amazon Prime Video: Which Streaming Service Is Better?

Of course, both services have different content, so having both is not a bad idea. However, if you only use one, it can be difficult to choose between the two. If you decide to use Netflix or Amazon Prime Video, the following part provides a detailed description of both services so you can decide which one is best for you.

Netflix v/s Prime Video Functionality

The Netflix Experience: Online Payment Service AlphrHOME, PAYPALNetflix vs. Amazon Prime Video: Which Streaming Service Is Better? Kassandra, December 24, 2020 Amazon

Prime Video Email and Netflix are two of the most popular streaming services in the United States. The more people are interested in streaming online, the more competitive the market will be. This confusion has led many consumers to wonder which of the two players in the code-shortening industry is better. Netflix vs Amazon Prime Video: Which Streaming Service Is Better? Of course, the content of the two services is different, so it's not a bad idea for both. However, if you only use one, it can be difficult to choose between the two. If you decide to use Netflix or Amazon Prime Video, this article provides a detailed description of both services so you can decide which one is best for you. Freestar Netflix vs. Prime Video Netflix Experience Netflix is mostly digital nowadays. Use

The Prime Video Experience: At the bottom of the list is the Show All option, which does not include Netflix. There is a lot of free content on this service, but some movies and shows charge additional fees for rentals, third-party subscriptions, or purchases. However, Prime members can see a long list of available Prime content.

Netflix v/s Prime Video Price Comparison

Netflix vs Prime Video Price Comparison Netflix and Amazon Prime Video offer at least a 7-day trial, but often you can use it for 30 days. If you want to give it a try before you buy, this is your chance to see what each product has to offer and how it works. If you’re a student, Amazon also offers a 6-month free trial. So, if you are still learning, there is no reason not to use Amazon Prime Video.

As mentioned earlier, Netflix charges a monthly fee, while Prime Video offers several options. With Prime Video, you create an account and only buy the content you want to watch, or pay monthly or yearly for your Amazon Prime account. If you choose Amazon Prime, you get free 2-day shipping, Prime Music Service, and more.

Hotstar

Hotstar is an Indian video streaming service owned and operated by Star India, a subsidiary of Walt Disney Company India. This service mainly includes content from the Star India and Fox networks, including films, TV shows, sports content and original programming, as well as third-party content such as HBO, Showtime and Sony Pictures.

Hotstar operates international versions of the service in Canada, Singapore, the United Kingdom and the United States. The service is aimed at overseas Indians and is mainly focused on Star India home entertainment and sports content (Disney is a self-service in this market). Disney's Star has expanded the integrated version of the Disney service to other markets such as Indonesia and plans to launch the service in Malaysia, the Philippines and Thailand in 2021.

Hotstar, India's leading OTT platform, and Zapr Media Labs, a media technology company, today announced a strategic partnership to promote India's next generation mobile audience analytics. In a development process that will appeal to advertisers and agencies, the two companies will work together to create a deeper understanding of mobile audiences that brands can use to create personalized communications and offerings.

Internet access in India has skyrocketed in recent years, especially on mobile screens, but mobile marketing is constrained by the lack of platforms for active user engagement and audience segmentation. While many brands have invested a lot of money in mobile in recent years, especially with banner and display ads, marketers have been frustrated by the lack of online brand building tools that can be used to conduct in-depth audience analysis. With an obvious leap towards digital marketing, this partnership could lead to more rigorous audience analysis and greater accountability for results in the mobile marketing world.

CHECK YOUR PROGRESS

XIII. ………… was started by three employees of PayPal online payment service.

a. YouTube   c. Instagram

b. Email   d. FaceBook

XIV. Video Viewers are those people who watch videos, interact videos and subscribe to channels. (True/False)

XV. ……………… platform is also known as video sharing platform.

XVI. Amazon Prime Video and Netflix are two of the most popular streaming services in the United States. (True/False)

XVII. Hotstar is an Indian video streaming service owned and operated by Star India, a subsidiary of Walt Disney Company India. (True/False)

11.26 LET US SUM UP

Searching is one of the most used moves at the internet. Search engines like Google as a tool, are very famous and mostly used websites. Website online proprietors use main search engines like Google and yahoo for filing their sites on it, and for searching. Regular users use number one search engines like Google essentially for looking, and every now and then for filing their Webpages. Google helps to market the products through the google adwards and google analytics by payperclick to the companies who are advertisers. Google marketing strategy is developed on the product qyality and usability but there are also other espects effects the marketing strategy. Google+ is a powerfull marketing tool as compare to other social networks. We can share informations, images and videos. Under the leadership of the Ballmer company, Microsoft try to design new products or a new service and in this try they make a software Bing in 2009 they started this search engine Bing. Bing come with new features of software and also helps in marketing. Yahoo is a brand with related to research, predicts, crate and associates in the motion brand shows the new achievements and the way of doing work and yahoo logo is made from wordmark. This is organized as the dictionary and the web index which help to search by most of people in march 1994 the web name as jerry and David 's Guide to the world wide web as the name Yahoo.

Video marketing is a part of marketing, by the means of which a company communicate the features of its product and services to its potential consumers. The basic objective of marketing by video is to convert the potential consumers to real time consumers. There are a lot of video marketing software which helps a manager to create a perfect marketing video for his product and services. As we all know YouTube is a video sharing service where we can access to various videos. People can watch, share, like and download videos from YouTube. We can access videos through smart phones, PCs, laptops etc. For any video creator the most important task is to keep the audience awake. Probably many of the users who are watching videos are tired. So, it is very much important for a you-tuber to help them stay awake, to actually pay attention and consider what they are saying. Video hosting platform is also known as video sharing platform. It provides a platform for businesses and individuals to share video. The video hosting platform handles video streaming and playback, providing users with a seamless video experience. It also helps to increase the traffic to the website. In addition to these advantages, a good hosting platform offers: Content security, Thumbnail management, adding subtitles, Basic analysis So far, everything looks good. In today's fast-paced Internet world, entertainment is the most stressful part of everyday life. Entertainment falls into several categories such as television, films, music, animation, games, etc. The biggest challenge in the media and entertainment industry is attracting viewers and increasing fan base. Increased use of smartphones and lower costs of accessing digital data have led to increased competition.

11.27 KEY WORDS

SEO Digital marketing tool to have high ranking for the company’s website.

GOOGLE+ It is a tool where marketers promote their goods and services and connect with other individuals and make good relationships with them.

VIDEO MARKETING It is a part of marketing tool to communicate the features of product and services to its potential consumers.

FLIPAGRAM It is a video editing software which combines the images to create short video stories.

WEVIDEO It is the web-based video editor which helps to create digital stories though still photographs.

VIDEO CREATORS These are those people who have their YouTube channels and upload videos for viewers.

VIDEO VIEWERS These are those people who watch videos, interact videos and subscribe to channels.

VIDEO HOSTING It is a platform handles video streaming and playback, providing users with a seamless video experience.

11.28 ANSWERS TO CHECK THE PROGRESS

I. Search Engine
II. True
III. False
IV. Google
V. Google+
VI. True
VII. Video marketing
VIII. Video
IX. All three
X. Stop Motion
XI. WeVideo
XII. Instagram
XIII. YouTube
XIV. True
XV. Video hosting
XVI. True
XVII. True

11.29 TERMINAL QUESTIONS

- What is search engine and how it contributes for the promotion of products and services?

- How Google is useful marketing tools for marketers? How marketing is done under Google? What are the tools of Google?
- What is Google+ and Bing? How they are useful tool for the business organizations?
- What is Video Marketing? Highlight the platforms used for promotion of products and services.
- Discuss the various tool video marketing used by the markets for marketing and what are their features?
- How YouTube has been emerged as one of the best marketing tools in the online market?
- What is video hosting and what are its benefits? Discuss the different platform for video hosting.
- Why entertainment in important for the customer and how Netflix and Hotstar contribute for the same?

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT-12.A MOBILE PHONES, E-MAILS, BLOGS

STRUCTURE

12.A.0 OBJECTIVES

After studying the Unit, you would be able to

- Use the Mobile strategy for product promotion
- Know the development of mobile and its benefits for business
- Learn about mobile marketing strategies and their importance
- Understand the Email marketing and its benefits
- Evaluate the success of email marketing strategy
- Elaborate the Blog Marketing and its significance
- Know the importance of Business Blog and its requirements

12.A.1 INTRODUCTION

Marketing is a concept to anticipate the needs and wants of targeted customers and satisfy them in order to create the new customers and retain the old customers. There are different ways of marketing that an organization opts for in order to perform more efficiently. Digital Marketing is one of the most efficient and effective way of marketing these days.

Mobile marketing is a part of digital marketing. It is a multi-channel strategy aimed at reaching large number of targeted audiences via social media, websites, emails, multimedia messages and different apps etc.

According to a report on “mobile marketing 2018” of Sherry Moats on brandonagency.com, there are over 3.5 billion individual internet users on mobile phones. On an average individual user spend 69% of their media time on smart-phones and 80% of the global internet usage is on mobile devices. As now 48% of the buyers use mobile devices or smart phones to start search on search engines therefore one must optimize their mobile marketing strategy to perform better than their competitors in the market.

Mobile connectivity not only enables people to connect to the Internet via a cellular telephone, PDA or other gadget, but also consolidates the **different communication channels** in a simple, yet effective manner. Cheaper than traditional means for both the consumer as well as for the marketer and easy enough for almost any age group to understand and engage with Mobile Marketing.

12.A.2 MOBILE STRATEGY

While considering mobile strategy, there are a number of things that need to take into account. These things need to be **planned in advance and implemented** as campaign takes shape.

- **Define goals:** What would business like to achieve, sales leads, newsletter signups. Remember to think SMART (Specific. Measurable. Achievable. Realistic. Timely.)
- **Research market:** Gain as much market understanding as possible to ensure that business approach is correct.
- **Plot presence:** Decide what mediums to use (MMS, mobile website, social media), and how to cross pollinate between them.
- **Define message:** What tone and angle is business going to take to inspire target audience to get involved.
- **Create sharing opportunities:** Create share points, and invite users to share content or information between handsets or platforms easily.

- **Feed campaign:** Fan the flames by releasing updates and encouraging renewed interest.
- **Track, analyse and optimise:** Mobile Marketing is great in that it allows for precision analysis and tracking, making campaign optimisation an ongoing process.

12.A.3 UNIQUE BENEFITS OF MOBILE

With direct response and relatively lower cost of communication, mobile is changing the face of the marketing and advertising horizon. Below has been given the seven features, unique to mobile phones, highlighting the unique benefits and challenges of the mobile phone.

- The mobile phone is personal
- The mobile phone is always carried
- The mobile phone is always on
- The mobile phone has an inbuilt payment system
- The mobile phone is available at the point of creative inspiration
  - The mobile phone presents accurate audience measurement
  - The mobile phone captures the social context of media consumption

12.A.4 MOBILE PHONES: REVOLUTIONARY DEVICES

Mobile phones are central to the lives of most people in developed countries and are growing in importance in developing countries. Since their mainstream adoption in the 1990s, they have remained primarily communication devices. We use mobile phones to talk to other people and we carry mobile phones with us so that other people can talk to us.

However, the situation is changing. Mobile phone manufacturers have developed mobile devices that can serve many functions beyond voice communication such as taking photos and listening to music. Mobile network operators are offering services that give greater value to subscribers, such as portable email for business users. Mobile phones are now equipped with cameras with the potential to turn them into portable bar code scanners. Handset manufacturers are developing RFID chips that can turn mobile phones into mobile wallets able to carry and exchange electronic money securely and engage in other transactions with RFID readers in the physical world.

The combination of more powerful mobile devices, more innovative mobile operators and change in the mobile network infrastructure (such as 3G networks able to carry large amounts of data at high speed as broadband connections do for computers) is setting the stage for an enormous change in an already fast-moving sector. Mobile devices are taking the place, where numerous technologies meet and create applications that are useful for both consumers and businesses across the globe. The mobile phone of the future is a device that enables users to communicate, connect, transact and innovate. In most markets, phones with the characteristics below are already becoming available:

- **A communicative device:** The mobile phone will continue to be a device that is used to communicate with others. Although this may be extended beyond voice to instant messaging and email, it is important not to forget communication is a central strength of mobile devices. As it becomes easier and cheaper to transfer larger amounts of data, sharing photos and videos with others will further extend this role.
- **A connective device:** Mobile phones enable people to connect to other sources of data anytime, anywhere. This is what is happening with mobile email. As data on the web becomes more structured, mobile devices will become more and more powerful as entry points to tasks that have moved from offline to online but are currently still only available through fixed computers.
- **A transactional device:** Mobile phones are ideal devices to be used for payments and transactions. There are a wide range of applications that aim to transform the mobile phone into an electronic wallet that can be used as a payment device.
- **An intelligent device:** Mobile phones are a place where multiple applications can meet and fuse. Mobile devices that integrate a phone, a camera, a location finder (GPS) and a connection to the internet make it possible for a user to request context-dependent information such as finding out where a store selling a product they want to buy is located. As usage increases, mobile phones can become agents of change, tools that facilitate connecting things in the physical world to information about them in the digital world.

12.A.5 SIGNIFICANCE OF MOBILE MARKETING

Significance components of mobile marketing are given under below:

- **Availability of fast administrations:** People need great, simple and speedy administrations. They are searching for a specialist cooperative which can satisfy their fundamentals. An organization can contact a client by means of mobile marketing showcasing strategies. A client can likewise send input without any problem.
- **Mobile has become an unreplaceable piece of life:** Mobile is a vital piece of day-by-day life. It can certainly convey to anyplace and that is the reason turns into a most loved gadget for e-shopping in recreation time.
- **Mobile is not utilized uniquely for e-shopping:** An actual store additionally utilizes useful administrations for offering types of help and promoting. Many stores will save your number to tell you about special items and limits offers.
- **SMS promoting:** “SMS” & “MMS” showcasing get a huge change advertising. Presently clients are getting offers and limits or get advised available to be purchased by means of “SMS” or “MMS”.
- **Primary Contents for Ranking:** Google will make search file dependent on versatile search positioning instead of work area search positioning; thus, your site search positioning will be founded on portable looking as opposed to work area looking.
- **Importance of Mobile Marketing:** Advertising on portable destinations or directly to useful appears to be more compelling in showcasing.

12.A.6 MOBILE MARKETING STRATEGIES

- **Mobile applications:** Mobile applications, document you to remember notices for versatile applications plan. Portable applications can incorporate gadgets, virtual machines or local applications, I think Facebook is an extraordinary illustration of promotions in an application.
- **In-game:** in this strategy we include advertisements inside a portable game.
- **SMS:** Utilizing SMS, advertisers can send direct messages to clients with offers and other data.

- **MMS (Sight & Sound Message administration):** MMS advertising sends direct messages that contain media things like pictures, video, or sound.
- **QR Codes (fast reaction standardized identification):** QR codes can be checked with the camera on a mobile phone. Once filtered, it will take shoppers to the connected site where they can see more data about a business.
- **Location-based:** Area based advertising utilizes the client's useful area to showcase a business in that space through GPS. For instance, you may be visiting Chicago. You may see advertisements for eateries close to you.
- **Mobile Banner Advertisements:** These are standards that are similar advertisements you will see on a work area, yet will be more uncertain to fit the cell phone's screen.
- **Near Field Communication (NFC):** “NFC” is a sort of invention that can send information to a purchaser's cell using a short-range remote association. “NFC” can share site pages or online media. It is most suitable for zones that are packed.
- **Bluetooth or Proximity Marketing:** Purchasers get area clear notices, which is empowered from their Bluetooth.
- **Voice Marketing:** Voice promoting utilizes a mechanized and (PC) created call to purchasers. The customer has the alternative to tune into the message or they may hang up. I often consider voice showcasing with legislative issues or noble cause.
- **Pay Per Call:** This kind of notice documents the client to tap the telephone number that is directly recorded in the advertisement.
- **Mobile Search Advertisements:** Useful inquiry advertisements are like Google's however are enhanced for portable.

CHECK THE PROGRESS

I. ………….. is a multi-channel strategy aimed at reaching large number of targeted audiences via social media, websites, emails, multimedia messages and different apps etc.

a. Email Marketing
c. Mobile Marketing
b. Blog Marketing
d. All three

II. Mobile phones are central to the lives of most people in developed countries and are growing in importance in developing countries. (True/False)

III. Handset manufacturers are developing ………… chips that can turn mobile phones into mobile wallets able to carry and exchange electronic money securely.

IV. Utilizing SMS, advertisers can send direct messages to clients with offers and other data. (True/False)

V. Area based advertising does not utilizes the client's useful area to showcase a business in that space through GPS. (True/False)

12.A.7 E-MAIL MARKETING

Email marketing is useful to enhance business through emails where businessmen can approach infinite numbers of customers with just a single click. Business uses different ways to send Email, as Email marketing is one of the best and most effective way of communication. E-mail marketing allow to keep a proper check on the results of email campaigns on the real time basis. Senders and receivers can communicate directly with each other.

Concept of the Email marketing is very simple, although there is different way a marketer can use e-mail. The different ways to use email marketing are:

- Use of software to sync all the emails together etc.
- Sending one email to thousands of customers on the real time and check the success rate, delivery percentage to know customer’s behaviour.

12.A.8 BENEFITS OF E-MAIL MARKETING

- **LOW-COST:** The biggest benefit of email marketing is its cut a larger portion of cost as compared to other mainstream marketing channels. In case of email marketing there is no print or postage costs and also no need to pay fee for big billboards, magazine or television channel. Businessmen who uses the Email marketing might consider investing in specialist software to automate, track and evaluate their emails. There may be a small cost expense for sending thousands of emails at a one time, but these costs are very much lower than what other marketing media would expect us to pay using their marketing channels.
- **EASY TO CREATE:** In case of Email marketing it does not need a group of people or technical expert in order to use it smoothly. Its use is very simple to use as well as

it is possible to create an email with fancy templates, videos, images and logos. But, most of people uses simple plain text emails, it shows that it's the content of an email that is the most important thing not the format.

- **EASY TO TRACK:** The other benefit of email marketing is that it's easy to check where we have committed the error. Email marketing software also permit us to track open, click-through and conversion rates, making it simple to spot how an email can be improved. These changes can be made immediately, whereas print or broadcast advertising requires quite a bit of effort to improve the error.
- **GLOBAL:** Email marketing give an option to send a message to hundreds/thousands of customers around the globe. It helps you spread the word amongst a global audience and also indicate us who's actually reading our content. This kind of flexibility is not there in traditional marketing.
- **COST EFFECTIVE:** The costs of email marketing are very much effective as it is much lower than any other type of marketing medium. In case of Email marketing there are no advertising fees or printing or media costs.
- **EASY SORTING:** In electronic mail marketing the mail listing is made from costumer who want the e-mail communications from the company. Customers who are interested about the goods or services are maximum possible to have interaction with the enterprise.
- **FLEXIBLE LAYOUT:** The marketer can send a simple text, pictures or connect files, whichever fits to his message. Flexible design gives an option to show the companies branding.
- **SCALABLE:** The email marketing can also be used to reach large group of audiences or smaller targeted lists whichever is preferred by marketing manager.
- **PERSONALISATION AND SEGMENTATION: W**ith the help of email marketing manager can also personalise the messages. He/she can also segment his marketing list, so that his customers receive messages that they are interested in, this will help boost customers engagement with the business.
- **SHAREABLE:** Forwarding and sharing of email is very easy and the costumers can easily forward the mail content to other people which helps in building companies' reputation by means of word-of-mouth or viral marketing. This will help influence new customers to boost their engagement with the business.

- **CONVERSION AND INCRESED SALES**: If the company have any new policy customers can simply click on links and follow the company's new policies immediately. Email marketing is also very useful at every phase of the purchasing process like Marketing manager can guide the costumer to choose his product, nurture the relationship with the customer and also inspire him/her for his future purchases.
- **MEASURABLE**: We can also evaluate the progress of our product by using web analytics software which helps in comparing our products. We can test different copy, subject lines and designs to see which is most efficient. This allows us to optimise future plan for our product.
- **BENCHMARK:** We can compare outcome from other industry which are in competition. There are thousands of free email marketing benchmarking reports available online, we just have to find these benchmarks by searching online. Benchmarking data can help you to evaluate and prioritise improvement opportunities in the future.
- **TEST BEFORE SENDING:** There is several software which help to, personalisation, email copy, images or messages and ensure the email content is effective and you can send it to the customers.
- **LESS INTRUSIVE:** As compared to cell phone marketing, customers can see our mail at any time that fits him. Customers also can change their choices if they like our email or unsubscribe if they do not need our e-mail communications any more.
- **ENVIROMENTAL FRIENDLY:** The email marketing is very much environment friendly as it is sent through means of internet and we don't have to print anything like in postal services.
- **TIME SAVING**: Through automation marketer could make emails to be despatched to clients primarily based on movement they've carried out at the website e.g. They can send a welcome e-mail while a person signs up at the website, or give them an e-mail presenting a discount in product if person remove items from purchasing cart. Once a template is generated it could reuse for many email campaigns.
- **REAL TIME MARKETING:** With the help of e-mail marketing a marketer can easily get in touch with customers in real-time. By using different automated software's like website activity, recent purchase or shopping cart a marketer can easily reach to the right customer, at the right time, in the right place and with the right offer.

12.A.9 CHALLEGES WITH E-MAIL MARKETING

- **NEED FOR WELL DEFINED TARGET COSTUMER:** First of all, to make the e-mail advertising and marketing a success whether small or at big scale, every company want highly focused client. Company cannot shoot in the dark because it will make the efforts of marketing officer fail. So, e-mail should be sent to the best targeted costumers. Otherwise, business will be losing a major client and the marketer will be wasting his and companies valuable time and efforts on wrong customer. Any such targeting a wrong costumer will waste not only companies time and resources but also be end up in, dropping credibility and also leads to negative effect companies brand image.
- **NEED FOR PERSONALIZATION:** Now, an increase of personalization is needed to seize and preserve the consumers attention. Without the personalization, the emails will no longer will be preferred by target audience. However, with the vital personalization, the ROI to marketer e-mail marketing improves. It is a crucial task that e-mail marketers need to understand. They need to use the information and facts that they have already got about a costumer to send personalised emails.
- **NEED FOR DEVELOPED LAYOUT SKILLS:** Marketer wants that the costumer must read their emails. However, until marketer message didn't seem appealing, the target consumer will become bored and will no longer reply to the conversation as marketer need them to. It is why e-mail marketers want a mail design skill. Businesses are looking for marketing persons who specialized in such skills.
- **NEED TO CONSTRUCT BETTER CREDIBILITY:** Now the competition level in the market has increased, the achievement of the marketers e-mail marketing relies upon at the credibility of company's brand value. If the company has a famous brand name, then there is no problem. Otherwise, marketers should first attempt to construct credibility. First of all, emails should firstly send to whose customers which have opted for the electronic mail advertising. Email layout also have an effect on the credibility of company's brand. Apart from that, the email should not feel like spam message.
- **LOWER POSSIBILITY FOR GETTING OBSERVED:** Now that each marketer desires that his message must be observed while each costumer has thousands of every day's emails receiving in his account, the possibilities of e-mail being opened

and read are low. Marketer has to do something which make him stand out to make sure his e-mail has been read. Getting observed is one of the most difficult components of e-mail marketing. The e-mail headlines appearance should be appealing for being opened. The message of the e-mail has to be brief and to the point and the use of words and pictures should be the proper proportion.

12.A.10 EVALUATING THE SUCCESS OF E-MAIL MARKETING

There are thousand reasons why the business needs to measure his overall success consistently. Marketer should know that what they are doing with their target customers is actually working or not. After that they should to be able to demonstrate their effectiveness to target customers. Finally, marketer need to use the company's success to construct loyalty from the consumers, even while they expand their audience base.

There are lots ways to check the success of marketer email campaigns:

- Open rate: No. of times the people open the e-mail
- Click-through rate: No. of people who click on the link in the e-mail
- Conversion rate: No. of person who become customers because of the e-mail
- Bounce rate: No. of unread or mail which don't reach to the audience.
- Unsubscribe rate: People that unsubscribe themselves from the marketer's e-mail list

CHECK THE PROGRESS

VI. Email marketing is useful to enhance business through emails where businessmen can approach infinite numbers of customers with just a single click. (True/False)

VII. ……………. marketing allow to keep a proper check on the results of email campaigns on the real time basis.

a. Affiliate Marketing
b. Blog Marketing
c. Email Marketing
d. Viral Marketing

VIII. …………….. software also permit us to track open, click-through and conversion rates, making it simple to spot how an email can be improved.

IX. There are thousands of free email marketing benchmarking reports available online, we just have to find these benchmarks by searching online. (True/False)

12.A.11 BLOGS

The Word Web-Blog is a combination of two words ‘web’ and ‘Log’. The blog first appeared in 1990s and at that time it’s just online diaries, where people provide personal information of their accounts of their life. Now companies used blogs as an information tool. It is used to provide information and updates to their customers. Now it is known as developing marketing strategy. All companies adopt this technique for marketing of their product or services. Blogging has more popular in these days and many people chosen it as their business.

12.A.12 BLOG MARKETIG

Starting of a new blog is not a work of just minutes. It will take some time for planning and management. Below are the steps for blog marketing:

- **Planning for Blogs**: First of all we need to do planning before the implementation of blog. It Includes planning regarding type of content that we want to share with customers and what time we have to update blogs, is it be daily, weekly or monthly.
- **Creation of blog:** After planning about the blog you need to create a blog. Insure that you can use same logo for your blog that you can use for website. The name of the blog should be attractive for customers.
- **Filling Blog With Posts:** Insure that you fill the blog with several posts because customers don’t like to read just one or two posts. So you need to add more than ten posts quickly and need to update them on regular intervals.
- **Market your Blog:** You have to try to reach your followers . For this purpose you need to integrate social media into your blogs. It increase the reach of content.
- **Reply To Comments**: Blogs are social so remember to reply the comments of peoples. If peoples ask questions than you have to answer these questions. And you need to think about the opinions given by peoples about content.
- **Encourage E-Mail Signups:** you need to encourage customers to signups with e mails.

12.A.13 PROS OF BLOG MARKETING

Blog marketing is helpful for businesses to reach their customers. They provide content to their customers with the help of blog marketing. Blog marketing has many benefits some of these are discuss below:

- **Less Cost**: Blog marketing is cost effective tool for businesses. At the cost of domain name and web hosting you can easily reach to customers.
- **Easy to use:** Blog marketing is easy to use. It is convenient tool for the business to promote their products or services.
- **Increase Popularity:** If businesses gives some offers and new content to their customers, it will increase the traffic on the website and gives the opportunity to customers for purchasing.
- **Improve Rankings:** Google always ranks new content so, blogging is helpful in rankings of the business, these rankings make businesses popular among customers.
- **Helping to gain trust:** With the help of blogging business prove that it is expert in their work. And business also give valuable information to their customers. Then customers feel good about business and they get ready to spend on products or services.
- **Build market:** Blogs helps the business to build their market. It is helpful to gain the trust of customers and it takes feedback from customers so in this way Blog provide customer services.
- **Increase revenues:** Blog is cost effective tool and it is helpful to increase the revenue of the business.

12.A.14 CONS OF BLOG MARKETING

- **Time consuming:** Blog is a time consuming work. Because creation of new content and to update blog takes maximum amount of time .
- **Lack of content**: It is difficult to think about new content. Bloggers face lack of content when they try to blogging.
- **No quick results:** Blog cannot give quick results. It takes some time to build the audience for blogs. We cannot think it gives us quick results.
- **Need to be marketed:** we use blog to market the business. For marketing we try to find out targeted market and the whole team for blogs.

12.A.15 EFFECTIVE BLOGGING STRATEGY

Blogging is a marketing tool for businesses in modern era. The followings are the main areas of concern to make blogging effective:

- **Plan**: Before starting blogging it is necessary to makes plans regarding the blogs like what type of content to be present and when to be present (time of update).
- **Research**: Before uploading any content through blogs it is necessary to research about Athe content and also research about the peoples who are interested to read this content.
- **Write**: After these two steps the next thing is to write the content. It should be in simple language or it should be attractive for the readers.
- **Upload**: After writing the content the next thing is to upload it. It is necessary that bloggers update the blog on regular intervals and the content should be attractive for readers.
- **Promote**: Promotion of the blog is very important for making blogs effective. You have to integrate blogs with your social media for this purpose.
- **Measure:** At the end we need to measure the effects of blogs. It should be the feedback given by the readers. These feedbacks help us to be better in the future. And it gives the idea to the business about the popularity of their content.

12.A.16 ELEMENTS OF BUSINESS BLOG

Now blogs are used by business to reach their customers and to show them that business is expert in their own works. Blogging helps the business to increase the brand name. It increases the followers of the business and establish themselves as an expert. The followings are the main elements of business blogs:

- **Good content:** It is necessary that business upload good content for their customers because good content increase the attention of customers. It increase the number of followers of the business .
- **It should have images and videos**: If in the blogs business have no images and videos then people don’t like it. In these days peoples are very visuals so we have to use the images, videos in the blogs.

- **Attractive**: Attractive content is a key element of business blog. In case business not able to provide good content then it fails to seek the attraction of peoples.
- **Shareable**: The information given by the business in their blogs should be shareable. It is necessary that information or content shared by business have good for the readers.
- **Post related things**: The business has need to post the related things in the blogs. It helps to catch attention of customers and with the help of these related posts readers continue reading your blogs and they don't get bored.

CHECK THE PROGRESS

X. The blog first appeared in 1990s and at that time it's just online diaries, where people provide personal information of their accounts of their life. (True/False)

XI. Blog marketing is easy to use as it is convenient tool for the business to promote their products or services. (True/False)

XII. Google does not ranks new content so, blogging is helpful in rankings of the business, these rankings make businesses popular among customers. (True/False)

XIII. The information given by the business in their blogs should be shareable. (True/False)

12.A.17 LET US SUM UP

Marketing is a concept to anticipate the needs and wants of targeted customers and satisfy them in order to create the new customers and retain the old customers. There are different ways of marketing that an organization opts for in order to perform more efficiently. Digital Marketing is one of the most efficient and effective way of marketing these days. With direct response and relatively lower cost of communication, mobile is changing the face of the marketing and advertising horizon. Mobile phones are central to the lives of most people in developed countries and are growing in importance in developing countries. Since their mainstream adoption in the 1990s, they have remained primarily communication devices. We use mobile phones to talk to other people and we carry mobile phones with us so that other people can talk to us. Email marketing is useful to enhance business through emails where businessmen can approach infinite numbers of customers with just a single click. Business uses different ways to send Email, as Email marketing is one of the best and most effective way of communication. E-mail marketing allow to keep a proper check on the results of email campaigns on the real time basis. Senders and receivers can communicate directly with each other. The Word Web-Blog is a combination of two words ‘web’ and ‘Log’. The blog first appeared in 1990s and at that time it’s just online diaries, where people provide personal information of their accounts of their life. Now companies used blogs as an information tool. It is used to provide information and updates to their customers. Now it is known as developing marketing strategy. All companies adopt this technique for marketing of their product or services. Blogging has more popular in these days and many people chosen it as their business.

12.A.18 KEY WORDS

MMS ADVERTISING	In this direct message are send that contain media things like pictures, video, or sound.
NEAR FIELD COMMUNICATION	It is a sort of invention that can send information to a purchaser's cell using a short-range remote association.
EMAIL MARKETING	It is one of the best and most effective way of communication.
CONVERSION RATE	Number of persons who become customers because of the e-mail.
BOUNCE RATE	Number of unread or mail which don’t reach to the audience.
BLOG	It is used to provide information and updates to their customers.

12.A.19 ANSWERS TO CHECK THE PROGRESS

I. Mobile Marketing
II. True
III. RFID
IV. True
V. False
VI. True
VII. Email Marketing
VIII. Email marketing
IX. True
X. True
XI. True
XII. False
XIII. True

12.A.20 TERMINAL QUESTIONS

- What is mobile marketing? Discuss the process for developing the Mobile strategy.
- Explain the Mobile evolution. What are the benefits of mobile?
- What is significance of mobile marketing? Highlight the mobile marketing strategies.

- What is Email marketing? What are the benefits of email marketing?
- Highlight the barriers for email marketing.
- What is blog marketing? Discuss the pros and cons of blog marketing.
- What are the elements of business blog? How to make the blog marketing effective?

.

COURSE: FUNDAMENTALS OF DIGITAL MARKETING

UNIT 12.B-- SOCIAL MEDIA: FACEBOOK, INSTAGRAM, TWITTER, WHATSAPP

STRUCTURE

12.B.0 OBJECTIVES

After studying the Unit, you would be able to

- Learn about the social network and significance of the social media marketing
- Best use of Instagram as marketing tool
- Learn how Instagram is important for marketing activities
- Have the knowledge how twitter emerged and marketers started using it in marketing
- Understand the tools of Twitter
- Learn about the WhatsApp and its best use in the marketing

12.B.1 SOCIAL NETWORK

The primary objective of SMM is to improve correspondence with clients, increment brand perceivability and arrive at increasingly (potential) clients. SMM is being finished by making quality substance that the clients of social systems share with their companions (through transmitting data electronically, for example Electronic Word of Mouth) by methods for intelligent alternatives on social systems, the clients find the opportunity to be heard, either by posing inquiries or by making objections.

We can make social media marketing more effective or efficiencies by proper planning. In social media marketing planning, we need to know about how much to invest in social media management tools, how to create engagement content, etc. Other thing which relates to social media marketing planning is how to drive traffic on website. Content is most necessary part to attract the consumer because attractive content makes traffic on website. All the customers like to see interesting things so marketers attract audience by creating good content. The different kinds of social media management tools are available in the market.

12.B.2 IMPORTANCE OF SOCIAL MEDIA FOR MARKETERS

- **Improve Brand Awareness:** Social media is one of the most peaceful and beneficial advanced marketing technique that can be utilized to build marketers' business perceivability. To begin, make social media profiles for business and begin organizing with others.
- **Cost Effective:** For a publicizing procedure, social media marketing is potentially the most financially savvy way. Making a record and joining is free for practically all social systems. On the off chance that marketer chooses to utilize paid promoting on social media, consistently begin little to perceive what they ought to anticipate. Being practical is significant as it causes accomplish a more prominent rate of profitability and hold a greater spending plan for other marketing.
- **Engage Customer:** Social media is a decent path for drawing in and associating clients. The more marketers speak with the group of spectators, the more possibilities they have of change. Set up a two-path correspondence with intended interest group so their desires are known well.
- **Improve Brand Loyalty:** Marketers having social media must make it simpler for clients to discover and interface with them. By interfacing with clients through social media,

marketers are increasingly likely to upsurge client maintenance and client dependability. Since building up a dedicated client base is one of the primary objectives of practically any business.

In today’s world where competition has become very stiff there is a need felt by every business to advertise its product to the fullest and reach its target consumers. Today the old techniques of marketing have taken a backseat and new marketing methods are being used. The use of social media for marketing has increased. Almost every business house is using social media for its products and service promotion. There are many social media platforms which are used by business houses for marketing like Instagram, Twitter, Facebook, YouTube, Google, Pinterest, etc.

12.B.3 INSTAGRAM

As the growth of social media has increased tremendously, which is helpful in building the relationships with the users. Social media has changed the behavior of the consumers of seeking the information, evaluation of the products and giving the feedback regarding the products or services to the companies. Moreover, the studies shows that the different videos and photos helps the brands or companies to attract more consumers, as a result this will helps in acquiring more attention of the consumers. As from the past years, Instagram has been at the top of all other social media platforms to advertise the products because now, everyone is using the Instagram.

12.B.4 INSTAGRAM MARKETING

We all know that Instagram marketing is the most competitive, but of course it is not impossible. Like, if anyone has huge number of audience or followers with him, then more and more brands will come to him for advertising or for promoting of the products.

Some of the related points to Instagram-

- Instagram sees more than one billion of the monthly users with more than 500 million stories.
- In the U.S, there are about 130 million Instagram users.
- Here on the Instagram, every user follows at least one business from which 83% users said that they are helped by discovering new products and services over Instagram.

12.B.5 HOW TO CREATE AN INSTAGRAM ACCOUNT

- Download the Instagram app form the play store or through the link on the google.
- Now, start with sign up the account or create a new account or sign in, if you already have an account.
- Enter the full name or business name so that profile ca be easily recognized.
- Now the username you have now is your unique username which no one can create again because the username is assigned only once.
- It is also important to pick the easy username that can be easily recognized.
- Optimize Instagram account with the right profile from which you are known or your business is related with. For the business, you can consider right logo from which your business can easily recognized.
- Fill up your bio with at least 150 characters. It is concise summary of you or your business.

12.B.6 STRATEGIES OF INSTAGRAM MARKETING

- **Organic Content:** Organic content is a way through which users upload their photos and videos which are unpaid. But to make audience engaged, the brands will collaborate with the users having higher followers and who are providing organic content.
- **Instagram Ads:** Instagram ads are more effective than the organic content if the users are looking for hard sales. So, the ads can be run on the Instagram like story ads, image ads, carousel ads, videos ads etc. Video and image ads can be run on the Instagram feed. There are some shoppable posts which directly advertise the products by adding tags on the images. For example, if a model wears any jewelry, the links would be provided with the image.
- **Influencer Marketing:** This is the most popular form of marketing where different influencers promote the products in their posts, stories, videos. Influencers can be from any industry like from tech to fashion to finance to food.

12.B.7 SOME OTHER INSTAGRAM MARKETING BASIC STRATEGIES

- **Firstly, find routine audiences,** this could be users produced by pixel track, an email record, a follower’s directory or another means. Record of customers who have linked with you previously perform most excellently.

- **When your previous custom viewers, similar viewers as previous must be your next target**, Facebook/Instagram's search router is great at discovering similar viewers that will react liking to your advertisement.
- **Conventional pictures of users are important, as it** accumulate photos RATHER than working on Instagram.
- **Create attractive offers,** Instagram's image character attracts desirable buyers but they might not buy or view that if you haven't present something persuasive.
- **Hashtags are mandatory, as** they motivate users and grandly connect with niche Instagram's user.

12.B.8 ADVANTAGES OF INSTAGRAM MARKETING

If you are not using Instagram for business marketing than here are few good reasons to startup marketing on Instagram. Since its foundation, Instagram has established as an influential digital marketing platform for businesses. It is beneficial for companies or brands thinking to develop their existence and for promotion of their products.

- **Increases the brand reputation:** Instagram helps in increasing the brand reputation. As we can see large portion of people are present on this platform, so it's easy to advertise the product on the Instagram to gain more customers towards the brand. If more and more people are engaging with you brand, you can build a strong brand reputation.
- **More traffic:** So, the need for the traffic is more important. So, choosing the right keywords which the customers may type or search for, would help to increase the traffic to your page. So, analyzing of social media platforms is must for reaching to the loyal customers.
- **Improves the ranking of search engine:** By using the social media platforms effectively, this will improve the ranking of the pages. Search engine optimization helps in optimizing which further improves the traffic.
- **Having more conversion rates:** If the conversion rates are high, this will help in having more opportunities because every post, blog, image or video helps the viewers to come to your website. Moreover, if the viewers share your webpage with others, it will automatically give a positive impression for your website.

12.B.9 BENEFITS OF INSTAGRAM MARKETING

- It will help in generating the income. It acts as an income source for all. You can earn online easily on the Instagram by promoting the different products or brands over the Instagram.
- It also helps in the brand awareness which means more and more people will know about business if you are running any business or if you have a business account.
- You can also conduct the social listening by doing conversations or making videos on some social topics which are beneficial for the public too.
- You can also share the story of your brand so that people would know about your brand and show some interest towards your brand. Effective stories really helpful for the brands.
- You can also have the feedback from customers about brand and services which you are providing to them. Specific keywords to be used in content is important, so that customers get easy access to your business.

12.B.10 INSTAGRAM STATISTICS

- ✓ Business profiles = 25 MILLION+
- ✓ Instagram is used by 71% U.S marketers.
- ✓ Instagram users following business accounts is 80%.
- ✓ New products and services are discovered by 85% of users daily.
- ✓ Instagram assists 80% of users choose which service or product to buy.
- ✓ About 30% users by some product or service(s)on their mobile APP.
- ✓ Daily about 200 million users check out one business account.
- ✓ Instagram is used by about 2 million advertisers all over the world.
- ✓ Funded videos get 3 times more comments than funded photos.

- **Millions of Instagram Users:** According to the stats of Instagram, there are currently about eight hundred million active users. Out of these million users about five hundred million day to day users and thirty eight percent only surf the account numerous times a day. With that load of accessible users, there is no limit to accomplish a top business with a fanatical Instagram plan.
- **Any size of business can thrive:** Among so many active users to pick from there is no limit for business to get promotion. That goes for large, well-known MNCs as well as smaller single-man handed business, even the well setup MNCs can’t get well success achievement in a day, but if their promotion team wants to get their business

on top, they can achieve so by being active online and sustain a regular update of at least one post a day.

- **Direct money can be made from Instagram:** Instagram has progressed and updated over the years therefore it has better prominence of earning money by product promotion. The most up-to-date feature is called shoppable posts which allow businesses to pin tags to the products and services in their photos along with links that comprise a product explanation, cost and the facility to "shop now", which directs the user to retailer's online store. Due to this new feature, attracting actual sales has been simplified for business and with seventy two percent of Instagram users admitted to buy product and services through this social media platform.
- **Stories create your business significant:** Instagram is a big platform to demonstrate prospective customers that you are more than just an anonymous firm. This can be made by numerous app's features, but you can truly build an intuition with live posts and stories. The relevant method to utilize live stories is to let people see behind-the-scenes snippets of your company and employs. FOR examples the videos that illustrate how products are prepared, videos of organization workers intermingle among themselves and live question and answers conference between you and your spectators.

  Instagram live posts are also superb mode to construct connection, faith and reliability with followers and presenting them labor operations of your business. If clients notice you as more than a unit looking to earn money, then they may get more expectation and hopes from your product.

- **Efficiently connecting with customers:** What is superior way of making your presence known to customers? How regarding chance to connect with them each day? The reality is that public like to express their opinions and desires something, therefore, IG is platform on which users like, comment and share their nearly all wanted posts. Companies' promotion and business marketing all depends upon no of likes and comments they get.
- **Mobility is king:** Contrasting to Facebook and Twitter, that started as browser-based sites, IG was formed primally to be an APP. Since 90 percent of time mobile time is spent on APPs, company can be the leader of promotion and make their products posts reachable to customers wherever they are through IG. Smartphone client lean to

Instagram as it has a cleaner and manageable presentation than the muddled view that Facebook produces, and with Instagram having TEN times higher active users than on Facebook marketing is more efficiently done.

- **Competitors can be analyses regularly:** Through Instagram, keen eye and regular updates on opposite companies can be kept. You can keep noting how many followers they have, what are their strategies, what measures they are taking to attract consumers towards themselves, what they are posting and how often, etc. It will help you or your company to use the information you assembled to enhance your own personal strategy for marketing.

CHECK THE PROGRESS

I. The primary objective of SMM is to improve correspondence with clients, increment brand perceivability and arrive at increasingly (potential) clients. (True/False)

II. Marketers can make social media marketing more effective or efficiencies by proper…………..

III. …………….. sees more than one billion of the monthly users with more than 500 million stories.

a. WhatsApp c. Instagram
b. YouTube d. Twitter

IV. ………… content is a way through which users upload their photos and videos which are unpaid.

V. Instagram ads are more effective than the ……………… if the users are looking for hard sales.

VI. About ……….. users by some product or service(s)on their mobile APP.

a. 20% c. 50%
b. 10% d. 30%

12.B.11 DRAWBACKS OF INSTAGRAM FOR BUSINESS

Most likely the first Instagram drawback is that Instagram is not a medium that just focus on businesses. Generally, users are using Instagram to surpass their free time and bond with other users. There is only a little number of users who have purpose to buy the product or service they spot on their timeline. Due to which, using Instagram as the only medium to advertise your business directs to the following disadvantages.

- **The Biggest Disadvantage is its Advantage:** IMAGES and VIDEOS are main content that is advantage of Instagram which can be turn out to be major disadvantage Instagram. This implies that you have to be very aware of the images or videos you post as it acts as content and will attract followers. Dull and boring pictures will make audience that follow you to skip your page in their timeline and not read the messages you want to convey in your status.
- **Everyone Is Not on Instagram:** As everyone doesn’t use Instagram to socialize it may come out to disadvantage for marketing. Instagram works only on an iPhone and Android app therefore if you are using Blackberry, or Linux it would not install on it. This problem will inhibit you to reach your target addressees.
- **Limitations on Contents:** Instagram has 60 seconds long limit of posting the video therefore you have to explain about your product or service in 60 seconds video only. Therefore, it comes as big drawbacks as you won’t be able to further give explanations of product in your post. Also, the ultimate length of captions on Instagram is 138-150 characters. For commercial purpose, the ideal length of the caption is 125 Characters which is very less. Also, using too many hashtags’ threats to get your profile banned therefore you can only use 5-7 hashtags only.
- **Difficulty in advertising:** Targeting and attracting local market is really problematic on Instagram as it has fewer choices to target users. Also, big and financially stable companies or business can only afford the commercial on Instagram as it is very expensive to sponsored by Instagram.

12.B.12 TWITTER

Twitter is an American online micro blogging and social networking service. On Twitter users interact through ‘tweets’ which are like messages. Initially, tweets were restricted to 140 characters but now it has been increased to 240 characters except for some languages. Twitter Inc. was founded in 2006 by Jack Dorsey, Noah Glass, Biz Stone and Evan Williams and is based in San Fransciso, CA. It was launched on 15th July, 2006.

Twitter is the one of the best digital platform or social media platform to start business and also there are lot of things which moves very quickly on this platform. If we look at the average watching or seeing of one tweet it is only about 16 minutes and there are more than 9000 new tweets have been tweeting every single second on this platform.

Twitter is now being used by brands and other business houses to promote their products and services. They are now actively engaging with people on the social media. Twitter has more than 300 million monthly active users and this makes it a preferable platform for marketers. A company can make its twitter account by signing up on twitter for free. A company has to make its twitter handle upload its details and then it can send tweets.

12.B.13 TWITTER MARKETING

Twitter is used by businesses for marketing. For marketing on Twitter firstly a company should have a Twitter handle. For this the company needs to sign up on Twitter. Following steps need to be followed:

- Go to Twitter.com and sign up, using your name, email id, password and username.
- Click on create my account.
- Build up timeline and follow people.
- Confirm the email id.
- A profile picture relevant to the brand should be uploaded.
- The business website should also be added to the profile so that people are able to easily trace the website and visit it.
- Add bio. The maximum character limit is 160 to tell the world about the business, be suffice. Something personal or quirky about the company or about the products and service can also be stated or the company can talk about their upcoming projects or events.
- Save the information.

Creating a twitter account is easy but it is not an easy task to grow the Twitter account and to turn it into an actual tool that builds up the brand. For successful Twitter marketing a company needs to engage with its audience and interact with them. Twitter marketing is a powerful tool. What makes Twitter different from other social media sites is that Twitter thrives off communication. Some of the main ways in which business houses use Twitter include:

- Sharing content
- Undertaking promotional activities
- Customer interaction
- Handling their grievances

- Twitter polls
- Providing expertise

12.B.14 FEATURES OF TWITTER MARKETING

- **Flexibility**: Marketing on Twitter is absolutely free and completely depends upon the user that how you want to operate it. There is completely your choice that what kind of content you want to post. This is really a good start kick for the small businesses as medium of advertising and promotion.
- **Ease of access:** Twitter is simplest social media platform and is easy to operate. Twitter is completely user-friendly.
- **Advertising**: Twitter like other social platforms provides advertisement support to the business to promote themselves and increasing their interactions.
- **Analytics**: One more feature of twitter is that it is inbuilt tracking platform that shows you your account activity. It is same like of google, Instagram and face book's analytics which show your activity in form of graph and reports to show you the important metrices.

12.B.15 HOW TO USE TWITTER FOR MARKETING

- **Customize and brand the profile:** When people look at a company's profile, they should be able to know that it's the company's profile. It means a company should customize their Twitter profile with their logo, header image, and any other recognizable and memorable details. Using the following a company can customize its profile:
  - Handle: Twitter handle is the username of the company (e.g., @hm) this should include the company's name so that the followers, customers can easily find the company's account on the platform.
  - Header: It is the background image on a profile. A company can choose to create a unique image for its header, it can use its logo or any other image related to its product or service.
  - Profile picture: Company's Twitter profile picture represents the company's twitter account it is the identity of the company it shows every move of the company, interaction and tweet. It is the image that is placed above the company's bio and might include a picture of company's founder or director.

- ➢ Bio: A company’s bio provides brief information about the company. It tells the users what the account is about what kind of content the people will find in this account. The company can include in its bio about their upcoming projects, product releases, engagements, objectives, their proud achievements, upcoming events, etc. A link to the company’s website can also be given in the bio for the ease of finding about the company and to generate leads. The character limit for the bio is about 160 so, the information should be precise and the company should use creativity to best describe its business and to attract people.

- **Twitter chats:** If a company wants to get more active followers on twitter, then the company should try to interact with its followers and the target consumers through twitter chats. There should be an interactive approach by the company when they Tweet something to start a conversation. The tweets should also be replied so that the audience feels connected with the company and it will add to the goodwill of the company.
- **Plan ahead:** It is very essential for the company to plan in advance about its upcoming events. Planning should be done in advance to make the upcoming events successful. For example, before tweeting about the launch of a product or before an occasion or event hash tags (#) can be used for the trend. The company should start planning about the campaigns prior to the start of the event. The hints about the new releases can also be given by the company to engage the audience.
- **Twitter video:** Another way of using twitter for marketing is using videos for advertisement. Twitter’s native video feature can be used for advertisement of the products and services. Videos can be recorded and uploaded to twitter live stream. Another option is to use Periscope, a live streaming app owned by Twitter. It integrates into Twitter to show the live streams in followers’ feed. The recording can be watched even after the live stream is over. There are different ways by which videos can be added on twitter the content of the video should be interesting so that it is liked and shared by the audience.
- **Follow people:** Following people and to engage in conversation with them is an important thing for marketing through Twitter. The business should follow influential people, journalists, etc. So that their content is able to reach many people at large. One thing that should be kept in mind is the number of followers following the company and the people the company is following. This ratio should be appropriate as

if the followers are less than the following then it puts out a negative impact. Business handles should not unnecessarily follow hundreds of people at once only those people should be followed who will help in promoting the brand and help in increasing the followers. Influential people should be followed. Some business houses also get paid promotions done on social networking sites through influencers the people who have huge fan following.

- **Advertise on Twitter:** By advertising on Twitter a company can easily reach its target audience. The company can advertise through visual ads, paid promotions, promoted tweets, etc.

  Promoted tweets make the tweets to appear in the twitter streams or twitter search results of specific users. It is a great option for getting more traffic on a specific webpage. The company will have to pay a certain sum of monthly fee as long as it wants to promote a tweet.

  If a company is using different tweets to achieve one objective, then the company can use twitter ads to achieve its goals in an efficient manner. Different companies have different objectives it includes app installs, video advertisement, website conversations, etc. So, the company has to decide between different objectives. This decision will impact the price the company will have to pay for running its ads.
- **Driving traffic to website:** Twitter can be used to drive traffic to the company's website. There are numerous ways in which this can be done. The company's URL can be included in the company's profile. The URL can be put in the bio of the company's twitter profile. In the tweets different links can be added.
- **Verified Twitter account:** The Company's account should look authentic for this purpose the company should get its account verified by Twitter. Twitter generally verifies the accounts if the account relates to a person who is into film and fashion industry, is a business personality, sportsperson, politician, content creator, etc. If Twitter accepts the company's application and verifies the profile a badge with a blue checkmark will appear next to the company's handle. Being verified will prevent the audience from following fake accounts and being confused by impersonator accounts or accounts with the same content. Business having a verified twitter account makes it more trustworthy and legitimate and differentiates it from a spam account.
- **Focus on increasing Twitter followers:** It is essential for the businesses to more the followers the company has more are the people looking at the have a large following

so as to reach more audience and target consumers. The company’s content and interacting with it. There are a number of ways by which a company can increase its followers:

- ➢ The content posted by the company should be sharable.
- ➢ Hash tags should be used for making it trendy.
- ➢ The company can use strategies like giveaways, contests, questions, surveys, etc.
- ➢ The social media influencers can be looked into for promoting the business.
- ➢ The links of the company’s twitter profile should be displayed on the company’s website.

- **Twitter analytics:** By using twitter analytics the company can know about the performance of its twitter campaigns and individual tweets. The monthly review about the company’s tweet and be seen and the performance can be analysed. By using analytics, the company can see whether it has been able to reach its target or not.
- **Twitter polls:** By conducting polls on Twitter the company can know about the popularity of the brand and its different products. It can know about the taste and preferences of its target consumers and can really know what actually the people want. It is a great way by which the company can interact with its customers and make them realize that the company does care about the opinions and needs of people.

12.B.16 TIPS FOR TWITTER MARKETING

- Use keyword targeting in Twitter Ads.
- Implement hashtags.
- Content sharing schedule should be organised.
- Twitter campaign.
- Create interesting content keeping in mind the different types of followers.
- Businesses can advertise about the offers offered by them through ads and video advertisement.
- Keeping an eye on the competitors twitter account.
- Celebrity endorsed advertisements can be done.
- Checking the DM’s and replying them.

- External tools can also be employed by the company to monitor its twitter activities. There are many tools available in the market for different purposes.

12.B.17 TOOLS FOR TWITTER MARKETING

There are many twitter marketing tools which can be used for better marketing and making the work easy. Some of the marketing tools are:

- **Sendible:** It is an all-around social media management tool that can be used to manage accounts from a plethora of different platforms including Twitter. With this tool the tweets can be scheduled and direct messages can be answered. Everything can be managed from a centralised dashboard. One of the most important features of this tool is the ability to schedule social media posts ahead of time. It helps in monitoring the brand's status on Twitter by integrating analytics from the brand's account and giving updates about the mentions of the brand.
- **Commun.it:** It is another social media management tool. This is the only tool which supports Twitter exclusively and is focused on engagement. It allows managing the mentions of the company in a much more efficient way by marking them as replied to, reviewed to and more. This section also allows monitoring the mentions of the brand. The posts can also be scheduled.
- **TweetDeck:** It is a twitter management tool built directly into twitter itself. It can be accessed with company's twitter account by navigating to tweetdeck.twitter.com or downloading the Mac app. This tool is essentially an extension of the Twitter dashboard and allows managing the company's main account and several additional accounts more efficiently.
- **Awario**: It is a different type of social media management tool than the other. Its main focus is on social listening and social monitoring. One of the most powerful features is its ability to spot potential leads. One of the ways it does this is by notifying about any tweets that ask for recommendations of a product or service that's similar to the business's product.
- **Social Sanp:** It helps in growing the audience and social reach by making your content more shareable. It starts with social share buttons which can placed throughout the site. This allows controlling the way the tweets appear on twitter. Customization can be done of the tweets image, title and description. Followers can

be encouraged to share the content on twitter by using Social Snap's click-to-tweet tool.

- **Hashtagify:** This tool shows the statics on hashtags from Instagram and twitter. It can be previewed by entering a hashtag on the homepage. You can search for a hashtag you are interested in using and the tool generates stats for it. These include the popularity of the hashtag, whether or not it's trending, related hashtags, etc. Individual hashtags can also be monitored.
- **Paper.li:** It is a tool that allows collecting and curating content with ease. A newspaper can be curated with the content from twitter and other social media platforms. Once the content has been gathered it can deliver to the people in the company's email list.

12.B.18 BENEFITS OR MODES OF TWITTER MARKETING

- **Direct messages**: like its other main competitors, twitter also allows direct messaging features for private conversation between two users but here is a difference that the two accounts who want to chat must follow each other. It sometimes becomes hurdles for the marketers because they have to follow each other but it surely creates genuine connections.
- **Tags promoting engagements**: The @ is a key which becomes a connection link between the tweets, by using this you can mention someone and it appears as a notification to the receiver so that they can easily view it.
- **Multiple posts are expectable, not annoying**: As we all know that Twitter is a fast-moving social media app, you never know what can be happened in just 1 or 2 hours so for being in updates the business have to be regular active and have to keep posting as there is no restrictions on the maximum posts or tweets to be send.
- **Tweet deck:** In simple language twitter deck, be said as the shortcut to be in the selected conversation. It is a free tool that helps monitoring accounts and hashtags you follow. It is an effective way to be in the conversation while avoiding other non-usable accounts.

  It is very useful for the businesses as the companies can watch their competitor's activity all at once so that they can makes strategies to beat all at once. For example, if you see a user who is complaining about a competing brand of your company because you're tracking their company, so you can reply and recommend your brand as a viable solution that will solve their problems.

- **Power of influencers for your campaigns**: Twitter is a huge platform where there are lots of marketers and influencers. The companies knew the powers of influencers, so they can use them to gain popularity and increasing their sales. There are approx. 40% chances of more sales when the users see the tweets of products from the brand companies and influencers. 49% users are those who responds because they rely on the recommendations from influencers and 40% are those who directly made purchase just by watching tweets from the influencers.
- **Low-cost advertising:** Advertising on twitter is really cost effective, it does not cost much so the marketers can grab the opportunities to Ad from the twitter. Twitter Ads guides marketers all kind of experiences through campaign creation process by suggesting various parameters to maximize impact.
- **Branded hashtags for community management:** Hashtags are the approximately used in all the social media platforms but is mostly on twitter. In twitter these hashtags are used to make a simple topic trending by using the hashtags again and again. The marketers can use this hashtag to promote their brands or products and can reach at trending level by regular deploying these hashtags.
- **Viral:** Here on twitter anyone can go viral. Here on Twitter the message or tweet spread like wildfire which sometimes became impossible to control. These viral messages can be funny, relatable or can be news. Those tweets can be generated from any of twitter account no matter how many followers a person have or not. Brands can use this technique to go viral with their tweets which can be worldwide reachable.

Till now we have discussed a lot about how companies can use twitter as a great marketing tool but we can’t forget that there are various companies who were masters at engaging users on Twitter. Yes, they are the worldwide companies who knows the importance of twitter marketing.

There are lots of companies using twitter but here we will be discussing top 3:

- **PLAYSTATION:** PlayStation is a video game brand which was produced by the Sony interactive Entertainment and the first console of the PlayStation was released in December 1994. This video giant is also an active user of twitter and is on second list with 21 million followers. On twitter they post wide variety of posts including links, videos and tweets. PlayStation keep themselves engage with their fans and also retweet them.

- **STARBUCKS:** We all are aware that Starbucks is known for its coffees and roastery reserves. It is an American multinational company whose headquarter is in Seattle, Washington. In Sep. 2020, the company had 32,600 stores in approx. 83 countries. As we talk about the twitter, Starbucks is an active user of twitter and have nearly 11 million genuine followers. Starbucks is quite active on twitter and keep posting tweets and pictures of their coffee to promote them.

  Main motive of using twitter by Starbucks are:

  - ➢ Satisfying customers
  - ➢ Listening and learning new from them
  - ➢ Bringing in-store experience to twitter
  - ➢ Focus on customer service

- **SAMSUNG MOBILE US:** Samsung Galaxy is a series of mobile computing devices that were designed and developed by the Samsung electronics. It launched on 29 June 2009 and is dealing in smartphones, tablets, phablet and smartwatches. There are various reasons why Samsung is at top 3 with 4.8 million followers which are:

  - ➢ Wide variety of content
  - ➢ Lots of retweets
  - ➢ Optimal utilization of hashtags

12.B.19 WHATSAPP MARKETING

WhatsApp marketing is similar to courier marketing in that it involves promoting a whole through WhatsApp. This channel assists brands in reaching a large audience and establishing strong customer relationships.

“Do you have WhatsApp?” we are sure someone has asked you that hundreds of times, and we are guessing you said yes. As a result, this app's popularity has skyrocketed. you'll realize ideas and tips to require advantage of this tool to the utmost and to try to WhatsApp promoting with success.

WhatsApp was founded in 2009 by January Koum and Brian Acton, two former Yahoo employees. Since then, the application's popularity has grown, especially in Western countries, with over 1200 million users registered in 2017. As a reward for this impressive distance, WhatsApp provides you with a promotional deal.

12.B.20 WHATSAPP MARKETING STRATEGY PROCESS

The following would be the process for the effective WhatsApp marketing strategy:

- Outline the objectives and key performance indicators (KPIs).
- Determine your target demographic.
- Download the company software.
- create a complete persona
- Create a contact list.
- design your contact profile

CHECK THE PROGRESS

VII. IMAGES and VIDEOS are main content that is advantage of Instagram which can be turn out to be major disadvantage Instagram. (True/False)

VIII. As everyone doesn't use Instagram to socialize it may come out to disadvantage for marketing. (True/False)

IX. A company can not make its twitter account by signing up on twitter for free. (True/False)

X. In Bio the maximum character limit is ………… to tell the world about the business, be suffice.

a. 100 c. 120

b. 200 d. 160

XI. A company's …….. provides brief information about the company as it tells the users what the account is about what kind of content the people will find in this account.

XII. By conducting polls on Twitter, the company can not know about the popularity of the brand and its different products. (True/False)

XIII. …………….. main focus is on social listening and social monitoring.

a. Commun.it c. Awario

b. TweetDeck d. Social Sanp

XIV. ……………….. helps in growing the audience and social reach by making your content more shareable.

a. Commun.it b. TweetDeck

c. Awario d. Social Sanp

XV. ……………… is a video game brand which was produced by the Sony interactive Entertainment.

XVI. WhatsApp was founded in 2009 by January Koum and Brian Acton, two former Yahoo employees. (True/False)

12.B.21 LET US SUM UP

The primary objective of SMM is to improve correspondence with clients, increment brand perceivability and arrive at increasingly (potential) clients. SMM is being finished by making quality substance that the clients of social systems share with their companions (through transmitting data electronically, for example Electronic Word of Mouth) by methods for intelligent alternatives on social systems, the clients find the opportunity to be heard, either by posing inquiries or by making objections. Social media has changed the behavior of the consumers of seeking the information, evaluation of the products and giving the feedback regarding the products or services to the companies. Moreover, the studies shows that the different videos and photos helps the brands or companies to attract more consumers, as a result this will helps in acquiring more attention of the consumers. As from the past years, Instagram has been at the top of all other social media platforms to advertise the products because now, everyone is using the Instagram. Most likely the first Instagram drawback is that Instagram is not a medium that just focus on businesses. Generally, users are using Instagram to surpass their free time and bond with other users. There is only a little number of users who have purpose to buy the product or service they spot on their timeline. Twitter is the one of the best digital platform or social media platform to start business and also there are lot of things which moves very quickly on this platform. If we look at the average watching or seeing of one tweet it is only about 16 minutes and there are more than 9000 new tweets have been tweeting every single second on this platform. Twitter is now being used by brands and other business houses to promote their products and services. WhatsApp marketing is similar to courier marketing in that it involves promoting a whole through WhatsApp. This channel assists brands in reaching a large audience and establishing strong customer relationships.

12.B.22 KEY WORDS

INFLUENCER This is the most popular form of marketing where different influencers

MARKETING promote the products in their posts, stories, videos.

TWITTER It is the one of the best digital platform or social media platform to start business and also there are lot of things which moves very quickly.

BIO A company’s bio provides brief information about the company.

SENDIBLE It is an all-around social media management tool that can be used to manage accounts from a plethora of different platforms including Twitter.

TWEETDECK It is a twitter management tool built directly into twitter itself.

12.B.23 ANSWERS TO CHECK THE PROGRESS

I. True
II. Planning
III. Instagram
IV. Organic
V. Organic content
VI. 30%
VII. True
VIII. True
IX. False
X. 160
XI. Bio
XII. False
XIII. Awario
XIV. Social Sanp
XV. PlayStation
XVI. True

12.B.24TERMINAL QUESTIONS

- What do you mean by Social Media Marketing?
- Discuss some of the social networking websites and their advantages.
- What is social network and how social media marketing is important for the marketers?
- What is Instagram marketing? Discuss the strategies of Instagram marketing.
- Highlight the fact and figures for the Instagram marketing.
- What is Twitter marketing? Explain the features of twitter marketing.
- What are the important elements of twitter marketing? Explain the key tips for twitter marketing.
- What are the tools and strategies for the twitter marketing?
- What is WhatsApp marketing and how it is significant for the company to promote the products?

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University
(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM
ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-II
Course: OPERATING SYSTEMS (MSCS-2-01T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

M.Sc. (Computer Science) Programme Coordinator:

Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:

Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Faculty of School of Business Management & Commerce:

Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Faculty of School of Social Sciences and Liberal Arts:

Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA
A State University Established by Govt. of Punjab vide Act No. 19
of 2019 and Approved Under section 2(f) of UGC

PROGRAMME COORDINATOR

Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE COORDINATOR

Dr. Monika Pathak (Assistant Professor)
School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Course: Operating Systems	
Course Code: MSCS-2-01T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Understand the structure of computing systems, from the hardware level through the operating system level and onto the applications level.
CO2	Understand basics of operating system viz. system programs, system calls, user mode and kernel mode.
CO3	Learn the working with CPU scheduling algorithms for specific situation, and analyze the environment leading to deadlock and its rectification.
CO4	Explore the memory management techniques viz. caching, paging, segmentation, virtual memory, and thrashing.
CO5	Apply Methods for Handling Deadlocks, Deadlock Prevention, and Recovery from Deadlock.

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

M.Sc. (Computer Science)

Semester II

MSCS-2-01T: Operating Systems

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

SECTION A

UNIT- I: Introduction and System Structures: Computer-System Organization, Computer-System Architecture, Operating-System Structure, Operating-System Operations, Process Management, Memory Management, Storage Management, Protection and Security, Computing Environments, Operating-System Services, User and Operating-System Interface, System Calls, Types of System Calls, System Programs.

UNIT II: Process Management: Process Concept, Process Scheduling, Operations on Processes, Multi-threaded programming: Multithreading Models, Process Scheduling: Basic Concepts, Scheduling Criteria, and Scheduling Algorithms.

Unit III: Deadlock: System Model, Deadlock Characterization, Methods for Handling Deadlocks, Deadlock Prevention, Deadlock Avoidance, Deadlock Detection, Recovery from Deadlock.

UNIT IV: Memory Management: Basic Hardware, Address Binding, Logical and Physical Address, Dynamic linking and loading, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Demand Paging, Page Replacement algorithms

SECTION B

UNIT V: File Systems: File Concept, Access Methods, Directory and Disk Structure, File-System Structure, File-System Implementation, Directory Implementation, Allocation Methods, Free-Space Management.

UNIT VI: **Introduction to Linux:** Linux's shell, Kernel, Features of Linux, Using file system: Filenames, Introduction to different types of directories: Parent, Subdirectory, Home directory; rules to name a directory, Important directories in Linux File System,

UNIT VII: Linux Commands: cal, date, echo, bc, who, cd, mkdir, rmdir, ls, cat cp, rm, mv, more, gzip, tar, File ownership, file permissions, chmod, Directory permission, change file ownership,

UNIT VIII: Shell Scripting: Creating and Executing Shell Programs, Using variables: Assigning a value to a variable, Accessing the value of a variable, Positional Parameters and other Built-In Shell Variables; Special Characters, Conditional Statements : if Statement, case Statement; Iteration Statements : for Statement, while Statement, until Statement

Suggested Readings

1. A Silberschatz, P.B. Galvin, G. Gagne, Operating Systems Concepts, 8th Edition, John Wiley Publications, 2009

2. A.S. Tanenbaum, Modern Operating Systems, 3rd Edition, Pearson Education, 2014

3. G. Nutt, Operating Systems: A Modern Perspective, 2nd Edition Pearson Education, 2000

4. S. Das, Unix Concepts and Applications, 4th edition, McGraw Hill Education, 2017

M.Sc. (Computer Science)
SEMESTER-II
OPERATING SYSTEM

UNIT 1: INTRODUCTION AND SYSTEM STRUCTURES

STRUCTURE

1.OBJECTIVE

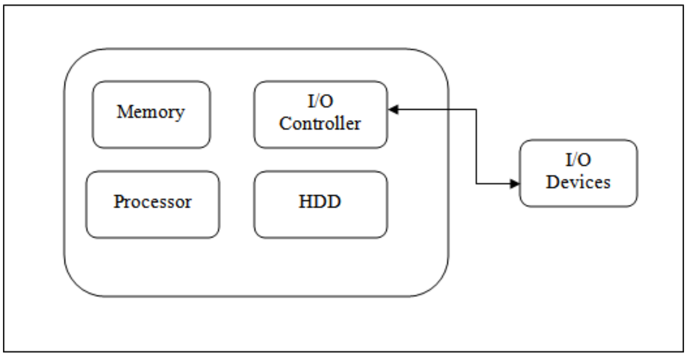
To understand the following

- Computer-System Architecture
- Operating-System Structure, Operating-System Operations
- Operating-System Services, User and Operating-System Interface
- System Calls, Types of System Calls, System Programs.

1.1.Operating System

- An operating system is a program which manages all the computer hardware.
- It provides the base for application program and acts as an intermediary between a user and thecomputer hardware.
- The operating system has two objectives such as:
    - Firstly, an operating system controls the computer's hardware.
    - The second objective is to provide an interactive interface to the user and interpretcommands so that it can communicate with the hardware.
- The operating system is very important part of almost every computer system.

Managing Hardware



- The prime objective of operating system is to manage & control the various hardware resourcesof a computer system.
- These hardware resources include processer, memory, and disk space and so on.
- The output result was display in monitor. In addition to communicating with the hardware the operating system provides on error handling procedure and display an

error notification.

- If a device not functioning properly, the operating system cannot be communicate with thedevice.

Providing an Interface

- It provides a stable and consistent way for applications to deal with the hardware without theuser having known details of the hardware.
- If the program is not functioning properly, the operating system again takes control, stops theapplication and displays the appropriate error message.
- Computer system components are divided into 5 parts
    - Computer hardware
    - Operating system
    - Utilities
    - Application programs
    - End user
- The operating system controls and coordinate a user of hardware and various applicationprograms for various users.
- It is a program that directly interacts with the hardware.
- The operating system is the first encoded with the Computer and it remains on the memory alltime thereafter.

System Goals

- The purpose of an operating system is to be provided an environment in which an user canexecute programs.
- Its primary goals are to make the computer system convenience for the user.
- Its secondary goals are to use the computer hardware in efficient manner.

1.2.VIEW OF OPERATING SYSTEM

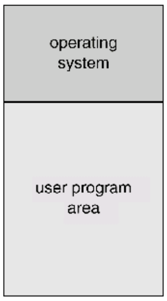
- **User view:**The user view of the computer varies by the interface being used. The examplesare -windows XP, vista, windows 7 etc. Most computer user sit in the in front of personal computer (pc) in this case the operating system is designed mostly for easy use with some attention paid to resource utilization. Some user sit at a terminal connected to amainframe/minicomputer. In this case other users are accessing the same computer through the other terminals. There user are share resources and may exchange the information. The operating system in this case is designed to maximize resources utilization to assume that all available CPU time, memory and I/O are used efficiently and no individual user takes more than his/her fair and share.The other users sit at workstations connected to network of other workstations and servers. These users have dedicated resources but they share resources such as networking and servers like file, compute and print server. Here the operating system is designed to compromise between individual usability and resource utilization.

- **System view:** From the computer point of view the operating system is the program which is most intermediate with the hardware. An operating system has resources as hardware and software which may be required to solve a problem like CPU time, memory space, file storage space and I/O devices and so on. That's why the operating system acts as manager of these resources. Another view of the operating system is it is a control program. A control program manages the execution of user programs to present the errors in proper use of the computer. It is especially concerned of the user the operation and controls the I/O devices.

1.3. TYPES OF OPERATING SYSTEM

1. **Mainframe System:** It is the system where the first computer used to handle many commercialscientific applications. The growth of mainframe systems traced from simple batch system where the computer runs one and only one application to time shared systems which allowed for user interaction with the computer system
   a. **Batch /Early System:** Early computers were physically large machine. The common input devices were card readers, tape drivers. The common output devices were line printers, tape drivers and card punches. In these systems the user did not interact directlywith the computer system. Instead the user preparing a job which consists of programming data and some control information and then submitted it to the computer operator after some time the output is appeared.

Memory Layout for a Simple Batch System



The output in these early computer was fairly simple is main task was to transfer control automatically from one job to next. Theoperating system always resides in the memory. To speed up processing operators batched the jobs with similar needs and ran then together as a group. The disadvantages of batch system are that in this execution environment the CPU is often idle because thespeed up of I/O devices is much slower than the CPU.

b. Multiprogrammed System: Multiprogramming concept increases CPU utilization by organization jobs so that the CPU always has one job to execute the idea behind multiprogramming concept. This set of job is subset of the jobs kept in the job pool. The operating system picks and beginning to execute one of the jobs in the memory. In this environment the operating system simply switches and executes another job. When a job needs to wait the CPU is simply switched to another job and so on. The multiprogramming operating system is sophisticated because the operating system makes decisions for the user. This is known asscheduling. If several jobs are ready torun at the same time the system choose one among them. This is known as CPU scheduling. The disadvantages of the multiprogrammedsystem are

- It does not provide user interaction with the computer system during the program execution.
- The introduction of disk technology solved these problems rather than reading the cards from card reader into disk. This form of processing is known as spooling.

SPOOL stands for simultaneous peripheral operations online. It uses the disk as a huge buffer for reading from input devices and for storing output data until the output devices accept them. It is also use for processing data at remote sides. The remote processing is done and its own speed with no CPU intervention. Spooling overlaps the input, output one job with computation of other jobs. Spooling has a beneficial effect onthe performance of the systems by keeping both CPU and I/O devices working at much higher time.

c. **Time Sharing System:** The time sharing system is also known as multi user systems. The CPU executes multiple jobs by switching among them but the switches occurs so frequently that the user can interact with each program while it is running. An interactive computer system provides direct communication between a user and system. The user gives instruction to the operating systems or to a program directly using keyboard or mouse and wait for immediate results. So the response time will be short. The time sharing system allows many users to share the computer simultaneously. Since each action in this system is short, only a little CPU time is needed for each user. The system switchesrapidly from one user to the next so each user feels as if the entire computer system is dedicated to his use, even though it is being shared by many users. The disadvantages oftime sharing system are:

   - It is more complex than multiprogrammed operating system
   - The system must have memory management & protection, since several jobs are keptin memory at the same time.
   - Time sharing system must also provide a file system, so disk management is required.
   - It provides mechanism for concurrent execution which requires complex CPU scheduling schemes.

2. **Personal Computer System/Desktop System:** Personal computer appeared in 1970's. They are microcomputers that are smaller & less expensive than mainframe systems. Instead of maximizing CPU & peripheral utilization, the systems opt for maximizing user convenience & responsiveness. At first file protection was not necessary on a personal machine. But when other computers 2nd other users can access the files on a pc file protection becomes necessary.The lack of protection made if easy for malicious programs to destroy data on such systems. These programs may be self replicating& they spread via

worm or virus mechanisms. They can disrupt entire companies or even world wide networks. E.g : windows 98, windows 2000, Linux.

3. **Microprocessor Systems/ Parallel Systems/ Tightly coupled Systems:** These Systems have more than one processor in close communications which share the computer bus, clock, memory & peripheral devices. Ex: UNIX, LINUX. Multiprocessor Systems have 3 main advantages.
   a. **Increased throughput:** No. of processes computed per unit time. By increasing the no. of processors move work can be done in less time. The speed up ratio with N processors is not N, but it is less than N. Because a certain amount of overhead is incurred in keeping all the parts working correctly.
   b. **Increased Reliability:** If functions can be properly distributed among several processors, then the failure of one processor will not halt the system, but slow it down. This ability to continue to operate in spite of failure makes the system fault tolerant.
   c. **Economic scale:** Multiprocessor systems can save money as they can share peripherals,storage & power supplies.

   The various types of multiprocessing systems are:

   - **Symmetric Multiprocessing (SMP):** Each processor runs an identical copy of the operating system & these copies communicate with one another as required. Ex: Encore's version of UNIX for multi max computer. Virtually, all modern operating systemincluding Windows NT, Solaris, Digital UNIX, OS/2 & LINUX now provide support forSMP.
   - **Asymmetric Multiprocessing (Master – Slave Processors):** Each processor is designed for a specific task. A master processor controls the system & schedules & allocates the work to the slave processors. Ex- Sun's Operating system SUNOS version 4provides asymmetric multiprocessing.

4. **Distributed System/Loosely Coupled Systems:** In contrast to tightly coupled systems, the processors do not share memory or a clock. Instead, each processor has its own local memory. The processors communicate with each other by various communication lines such as high speed buses or telephone lines. Distributed systems depend on networking for their functionalities. By being able to communicate distributed systems are able to share computational tasks and provide a rich set of features to the users. Networks vary by

the protocols used, the distances between the nodes and transport media. TCP/IP is the most common network protocol. The processor is a distributed system varies in size and function. Itmay microprocessors, work stations, minicomputer, and large general purpose computers. Network types are based on the distance between the nodes such as LAN (within a room, flooror building) and WAN (between buildings, cities or countries). The advantages of distributed system are resource sharing, computation speed up, reliability, communication.

5. **Real time Systems:** Real time system is used when there are rigid time requirements on the operation of a processor or flow of data. Sensors bring data to the computers. The computer analyzes data and adjusts controls to modify the sensors inputs. System that controls scientific experiments, medical imaging systems and some display systems are real time systems. The disadvantages of real time system are:
   - **a.** A real time system is considered to function correctly only if it returns the correct result within the time constraints.
   - **b.** Secondary storage is limited or missing instead data is usually stored in short term memory or ROM.
   - **c.** Advanced OS features are absent.Real time system is of two types such as:
   - **Hard real time systems:** It guarantees that the critical task has been completed on time.The sudden task is takes place at a sudden instant of time.
   - **Soft real time systems:** It is a less restrictive type of real time system where a critical task gets priority over other tasks and retains that priority until it computes. These have more limited utility than hard real time systems. Missing an occasional deadline is acceptable

     e.g. QNX, VX works. Digital audio or multimedia is included in this category.

   It is a special purpose OS in which there are rigid time requirements on the operation of a processor. A real time OS has well defined fixed time constraints. Processing must be done within the time constraint or the system will fail. A real time system is said to function correctlyonly if it returns the correct result within the time constraint. These systems are characterized by having time as a key parameter.

1.4. Functions of Operation System

The various functions of operating system are as follows:

1. **Process Management**

- A program does nothing unless their instructions are executed by a CPU.A process is a programin execution. A time shared user program such as a complier is a process. A word processing program being run by an individual user on a pc is a process.
- A system task such as sending output to a printer is also a process. A process needs certain resources including CPU time, memory files & I/O devices to accomplish its task.
- These resources are either given to the process when it is created or allocated to it while it is running. The OS is responsible for the following activities of process management.
- Creating & deleting both user & system processes.
- Suspending & resuming processes.
- Providing mechanism for process synchronization.
- Providing mechanism for process communication.
- Providing mechanism for deadlock handling.

2. **Main Memory Management**

The main memory is central to the operation of a modern computer system. Main memory is a large array of words or bytes ranging in size from hundreds of thousand to billions. Main memory stores the quickly accessible data shared by the CPU & I/O device. The central processor reads instruction from main memory during instruction fetch cycle & it both reads & writes data from main memory during the data fetch cycle. The main memory is generally theonly large storage device that the CPU is able to address & access directly. For example, for theCPU to process data from disk. Those data must first be transferred to main memory by CPU generated E/O calls. Instruction must be in memory for the CPU to execute them. The OS is responsible for the following activities in connection with memory management.

- Keeping track of which parts of memory are currently being used & by whom.
- Deciding which processes are to be loaded into memory when memory space becomesavailable.
- Allocating &deallocating memory space as needed.

3. **File Management**

File management is one of the most important components of an OS computer can store information on several different types of physical media magnetic tape, magnetic disk & opticaldisk are the most common media. Each medium is controlled by a device such as disk drive or tape drive those has unique characteristics. These characteristics include access speed, capacity, data transfer rate & access method (sequential or random).For convenient use of computer system the OS provides a uniform logical view of information storage. The OS abstracts from the physical properties of its storage devices to define a logical storage unit the file. A file is collection of related information defined by its creator. The OS is responsible for the followingactivities of file management.

- Creating & deleting files.
- Creating & deleting directories.
- Supporting primitives for manipulating files & directories.
- Mapping files into secondary storage.
- Backing up files on non-volatile media.

4. **I/O System Management**

One of the purposes of an OS is to hide the peculiarities of specific hardware devices from the user. For example, in UNIX the peculiarities of I/O devices are hidden from the bulk of the OSitself by the I/O subsystem. The I/O subsystem consists of:

- A memory management component that includes buffering, catching & spooling.
- A general device- driver interfaces drivers for specific hardware devices. Only the device driver knows the peculiarities of the specific device to which it is assigned.

5. **Secondary Storage Management**

The main purpose of computer system is to execute programs. These programs with the data they access must be in main memory during execution. As the main memory is too small to accommodate all data & programs & because the data that it holds are lost when power is lost. The computer system must provide secondary storage to back-up main memory. Most modern computer systems are disks as the storage medium to store data &

program. The operating system is responsible for the following activities of disk management.

- Free space management.
- Storage allocation.
- Disk scheduling

Because secondary storage is used frequently it must be used efficiently.

1.5. NETWORKING

A distributed system is a collection of processors that don’t share memory peripheral devices or a clock. Each processor has its own local memory & clock and the processor communicate with one another through various communication lines such as high speed buses or networks. The processors in the system are connected through communication networks which are configured in a number ofdifferent ways. The communication network design must consider message routing & connection strategies are the problems of connection & security.

1.6. PROTECTION OR SECURITY

If a computer system has multi users & allow the concurrent execution of multiple processes then the various processes must be protected from one another’s activities. For that purpose, mechanisms ensure that files, memory segments, CPU & other resources can be operated on by onlythose processes that have gained proper authorization from the OS.

1.7. SYSTEM CALLS

System calls provide the interface between a process & the OS. These are usually available in the form of assembly language instruction. Some systems allow system calls to be made directly from ahigh level language program like C, BCPL and PERL etc. systems calls occur in different ways depending on the computer in use. System calls can be roughly grouped into 5 major categories.

1.7.1. Process Control

□ **End, abort:** A running program needs to be able to has its execution either normally (end) orabnormally (abort).

- **Load, execute:** A process or job executing one program may want to load and executes another program.
- **Create Process, terminate process:** There is a system call specifying for the purpose of creating a new process or job (create process or submit job). We may want to terminate a job or process that we created (terminates process, if we find that it is incorrect or no longer needed).
- **Get process attributes, set process attributes:** If we create a new job or process we should able to control its execution. This control requires the ability to determine & reset the attributes of a job or processes (get process attributes, set process attributes).
- **Wait time:** After creating new jobs or processes, we may need to wait for them to finish theirexecution (wait time).
- **Wait event, signal event:** We may wait for a specific event to occur (wait event). The jobs orprocesses then signal when that event has occurred (signal event).

1.7.2. File Manipulation

- **Create file, delete file:** We first need to be able to create & delete files. Both the system callsrequire the name of the file & some of its attributes.
- **Open file, close file:** Once the file is created, we need to open it & use it. We close the file when we are no longer using it.
- **Read, write, reposition file:** After opening, we may also read, write or reposition the file (rewind or skip to the end of the file).
- **Get file attributes, set file attributes:** For either files or directories, we need to be able to determine the values of various attributes & reset them if necessary. Two system calls get fileattribute & set file attributes are required for their purpose.

1.7.3. Device Management

- **Request device, release device:** If there are multiple users of the system, we first request the device. After we finished with the device, we must release it.
- **Read, write, reposition:** Once the device has been requested & allocated to us, we can read,write & reposition the device.

1.7.4. Information Maintenance

- **Get system data, set system data:** Other system calls may return information about thesystem like number of current users, version number of OS, amount of free memory etc.
- **Get process attributes, set process attributes:** The OS keeps information aboutall itsprocesses & there are system calls to access this information.

1.7.5. Communication

There are two modes of communication such as:

a. **Message passing model:** Information is exchanged through an inter process communication facility provided by operating system. Each computer in a network has a name by which it is known. Similarly, each process has a process name which is translated toan equivalent identifier by which the OS can refer to it. The get hostid and get processed systems calls to do this translation. These identifiers are then passed to the general purpose open & close calls provided by the file system or to specific open connection system call. Therecipient process must give its permission for communication to take place with an accept connection call. The source of the communication known as client & receiver known as server exchange messages by read message & write message system calls. The closeconnection call terminates the connection.

b. **Shared memory model:** processes use map memory system calls to access regions of memory owned by other processes. They exchange information by reading & writing data inthe shared areas. The processes ensure that they are not writing to the same locationsimultaneously.

1.8. SYSTEM PROGRAMS

System programs provide a convenient environment for program development & execution. Theyare divided into the following categories.

a. File manipulation: These programs create, delete, copy, rename, print & manipulate filesand directories.

b. **Status information:** Some programs ask the system for date, time & amount of availablememory or disk space, no. of users or similar status information.

i. **File modification:**Several text editors are available to create and modify the contents of filestored on disk.

ii. **Programming language support:** compliers, assemblers & interpreters are provided to the user with the OS.

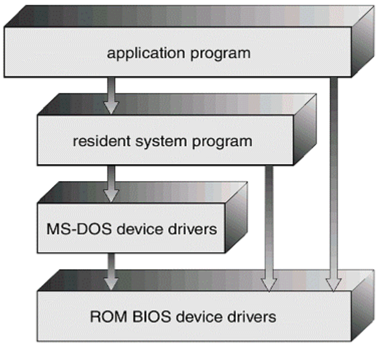
iii. **Programming loading and execution:** Once a program is assembled or compiled, it mustbe loaded into memory to be executed.

iv. **Communications:** These programs provide the mechanism for creating virtual connectionsamong processes users 2nd different computer systems.

v. **Application programs:** Most OS are supplied with programs that are useful to solve common problems or perform common operations. Ex: web browsers, word processors & text formatters etc.

1.9. SYSTEM STRUCTURE

1. **Simple structure:** There are several commercial system that don't have a well-defined structure such operating systems begins as small, simple & limited systems and then grow beyond their original scope. MS-DOS is an example of such system. It was not divided intomodules carefully. Another example of limited structuring is the UNIX operating system.

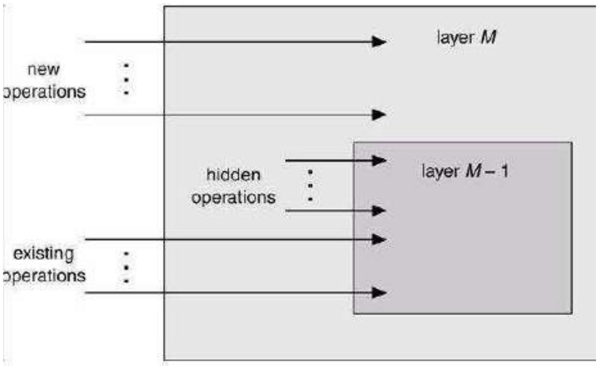


2. **Layered approach:** In the layered approach, the OS is broken into a number of layers (levels) each built on top of lower layers. The bottom layer (layer o) is the hardware &

top

most layer (layer N) is the user interface. The main advantage of the layered approach is modularity.

- The layers are selected such that each users functions (or operations) & services of only lower layer.



This approach simplifies debugging & system verification, i.e. the first layer can be debugged without concerning the rest of the system. Once the first layer is debugged, its correct functioning is assumed while the 2nd layer is debugged & so on.

If an error is found during the debugging of a particular layer, the error must be on that layer because the layers below it are already debugged. Thus the design & implementation of the system are simplified when the system is broken down into layers.

Layers	Functions
5	User Program
4	I/O Management
3	Operator Process Communication
2	Memory Management
1	CPU Scheduling
0	Hardware

Each layer is implemented using only operations provided by lower layers. A layer doesn't need to know how these operations are implemented; it only needs to know what these operations do.

The layer approach was first used in the operating system. It was defined in six layers.

The main disadvantage of the layered approach is:

- The main difficulty with this approach involves the careful definition of the layers, because a layer can use only those layers below it. For example, the device driver for thedisk space used by virtual memory algorithm must be at a level lower than that of the memory management routines, because memory management requires the ability to usethe disk space.

- It is less efficient than a non layered system (Each layer adds overhead to the system call & the net result is a system call that take longer time than on a non layered system).

1.10. Operating System Services

An operating system provides an environment for the execution of the program. It provides some services to the programs. The various services provided by an operating system are as follows:

- **Program Execution:** The system must be able to load a program into memory and to run that program. The program must be able to terminate this execution either normally or abnormally.
- **I/O Operation:** A running program may require I/O. This I/O may involve a file or a I/O device for specific device. Some special function can be desired. Therefore the operating system must provide a means to do I/O.
- **File System Manipulation:** The programs need to create and delete files by name and read and write files. Therefore the operating system must maintain each and every files correctly.
- **Communication:** The communication is implemented via shared memory or by the technique of message passing in which packets of information are moved between the processes by the operating system.
- **Error detection:** The operating system should take the appropriate actions for the occurrences of any type like arithmetic overflow, access to the illegal memory location andtoo large user CPU time.
- **Research Allocation:** When multiple users are logged on to the system the resources must be allocated to each of them. For current distribution of the resource among the various processes the operating system uses the CPU scheduling run times which determine which process will be allocated with the resource.
- **Accounting:** The operating system keep track of which users use how many and which kindof computer resources.
- **Protection:** The operating system is responsible for both hardware as well as software protection. The operating system protects the information stored in a multiuser

computer system.

1.11. PRACTICE EXERCISE

a. Why is the operating system important?

b. What's the main purpose of an OS? What are the different types of OS?

c. What are the benefits of a multiprocessor system?

d. What is a bootstrap program in OS?

e. What is the main purpose of an operating system?

f. What are the different operating systems?

g. What is kernel?

h. What is monolithic kernel?

i. What do you mean by a process?

M.Sc. (Computer Science)
OPERATING SYSTEM

UNIT 1: INTRODUCTION AND SYSTEM STRUCTURES

STRUCTURE

OBJECTIVES

To understand the following

- Different types of Process Scheduling
- Operations on Processes
- Multi-threaded programming and Model
- CPU Scheduling
- Scheduling Concepts
- Scheduling Criteria
- Scheduling Algorithms

INTRODUCTION TO PROCESS

Process Basics

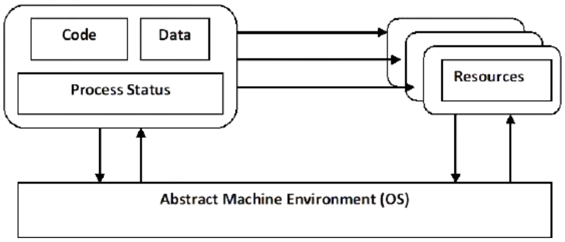
The fundamental activity of an operating system is the creation, management, and termination of processes. Now the question comes to mind, what is a process?

A process is a program under execution or the “animated” existence of a program or an identifiable entity executed on a processor by the operating system.

A process may be defined as an instance of a program in execution. It is also known as a task. An operating system manages each hardware resource attached to the computer by representing it as an abstraction. Abstraction hides unwanted details from the users and programmers allowing them to have a view of the resources in the form, which is convenient to them. A process is an abstract model of a sequential program in execution. The operating system can schedule a process as a unit of work. A process can be identifying in an operating system by its following components:

The object program (or code) to be executed.
The data on which the program will execute.
The status of the process execution.
The stack is associated with a process to store parameters and calling addresses.

A process may be represented schematically as in the figure.



Process Abstraction

Process Description

To control and manage the processes, the Operating system records the information about them in the primary process table. The primary process table is used to keep one entry per process in the operating system. Each entry contains at least one pointer to a process image. The Process Image contains **User Data**: It contains program data that can be modified etc., **Code:** The sequence of instructions (program) to be executed, **Stack:** Each process has one or more stacks associated with it. A stack is used to saved parameters of the process and calling addresses for process, system calls, and **Process Control Block (PCB)** of process in which data needed by the operating system to control the process (attributes and information about the process) is stored.

Process Control Block

The data structure that stores information about a process is called Process Control Block. When a process is initialized, the corresponding process control block of the process is created. Information in a process control block is updated during the transition of process states.

A process control block is a location in the main memory, where various information of a process regarding memory, process, and I/O management is stored. Each process has a single process control block. When a process is completed the process control block is unloaded from the memory. The information stored in a process control block is:

Pointer	**Process State**
Process Number	

Program Counter
Registers
Memory Limits **List of open files**
. . . .

Figure: Process Control Board (PCB)

Each process has a **priority**, implemented in terms of numbers. The higher priority processes have precedence over lower priority processes. The priority field is used for storing the priority of a process.

A new process can be created from the existing process. The existing process is called the parent of the newly created process. The field, **link** (pointer) to the parent process stores the address of the process control block of the parent process in the main memory. The field, link (pointer) to the child process stores the address of the process control block of the child process in the main memory.

Process State field stores the information about the recent state of the process. The state of the process may be new, waiting, ready, running, and so on.

Process Number filled to store the numeric value which is an identifier of the process.

The PCB stores information regarding the programming environment of a process. The programming environment information includes the value of the registers, stack, and program counter.

A **program counter** is a special register that saves the address of the next instruction to be executed for this process.

A PCB also stores information regarding memory management, such as the number of memory units allocated to the process and the addresses of the memory chunks allocated.

Registers: It includes a general-purpose register, stack pointers, index register, and accumulators, etc. Many register and accumulators etc. Some register and type of register depend upon the computer architecture.

The PCB control block stores file management information. An example of the file management information stored in the process control block is:

- The number of files opens.
- List of the open files.
- The access right of the files opens, such as read-only or read-write.

Role of PCB

The process control block is the most important data structure in an OS. Each Process Control Block contains all of the data about a process that is required by the Operating System. The blocks are read as well as modified by virtually every unit in the Operating System, including those which involved scheduling, interrupt processing, resource allocation, and performance monitoring and analysis of the process.

One can say that the set of process control blocks defines the state of the OS. This brings up an important design issue. Many routines within the OS will need access to information in process control blocks. The provision of direct access to these tables is not difficult. Each process is equipped with a unique ID, and this can be used as an index into a table of pointers to the process control blocks.

The difficulty is not access but rather protection. There are two problems :

1. A bug in a single routine, such as an interrupt handler, could damage process control blocks, which could destroy the system’s ability to manage the affected.
2. A design change in the structure or semantics of the process control block could affect many modules in the OS.

PROCESS SCHEDULERS

Several types of schedulers can be used in an OS. Schedulers are classified according to the type and duration of processes. Schedulers are classified as:

- Long-term Scheduler/High-Level Scheduler
- Medium-term Scheduler/ Intermediator Scheduler
- Short-term Scheduler/Low-Level Schedular

Long-term Scheduler

Long-term scheduling identifying which programs is to be admitted to the system as new processes. Once a new process is accepted, it may enter the scheduling queues in one of two places:

- If all resources are initially fully available to the new process, they may be admitted to the tail of the ready queue.
- If all resources are not immediately available, the new process may be entered in the blocked-suspended queue until those resources are provided.

Long-term Scheduler plans the CPU scheduling for batch jobs. Processes, which are resource-intensive and have a low priority, are called batch jobs are executed in a group or bunch. An example of a batch job is a user request for printing a bunch of files.

Medium-term Scheduler

Medium-term scheduling is part of the swapping role of an operating system. Medium-term Scheduler plans CPU scheduling for the processes that have been waiting for the completion of another process or an I/O task that requires a long time. A process is suspended or blocked marked as waiting if it is waiting for the completion of a long time I/O task. These processes are removed from the main memory and stored in the swapped-out queue in the secondary memory to create space in the main memory. The swapped-out queue is implemented in the secondary memory for storing the waiting processes that have been swapped out of the main memory. After completion of the I/O operation, the suspended or blocked processes are resumed and placed in the ready queue in the main memory. If the process is waiting for the completion of a short-term I/O task, the process is not swapped out of the main memory and is not handled by the medium-term scheduler. The success of the medium-term schedules is based on the degree of multiprogramming that it can maintain, by keeping as many processes “runnable” as possible.

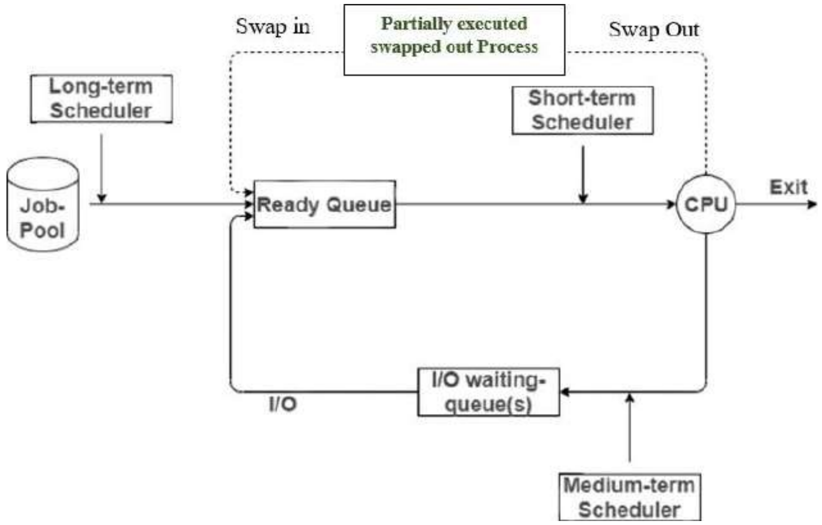


Figure: Working of medium-term scheduler

Short-term Scheduler

Short-term Scheduler plans the scheduling of the processes that are in a ready state. Short-term schedulers retrieve a process from the ready queue and allocate CPU time to it. The process state is changed from ready to run. If an interrupt or time-out occurs the scheduler places the running process back into the ready queue and marks the running process as ready. **The figure below shows how short, medium, long term schedulers work.**

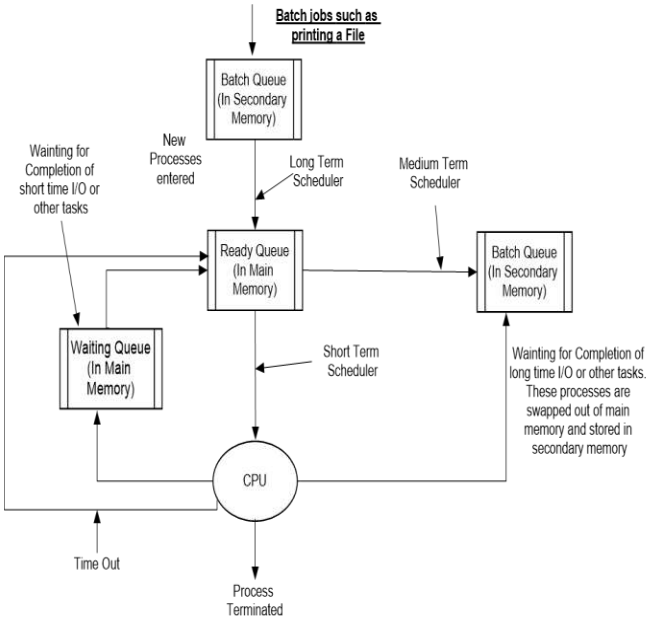

Batch jobs such as printing a File
Batch Queue (In Secondary Memory)
New Processes entered
Long Term Scheduler
Medium Term Scheduler
Wainting for Completion of short time I/O or other tasks
Ready Queue (In Main Memory)
Batch Queue (In Secondary Memory)
Waiting Queue (In Main Memory)
Short Term Scheduler
Wainting for Completion of long time I/O or other tasks. These processes are swapped out of main memory and stored in secondary memory
CPU
Time Out
Process Terminated

Figure: Role of various type schedulers

If a new process is a batch job, it is placed in the batch queue; otherwise, it is ready into the ready queue. The long-term scheduler selects a batch job on a first-come-first-served basis and sends it to the ready queue for execution.

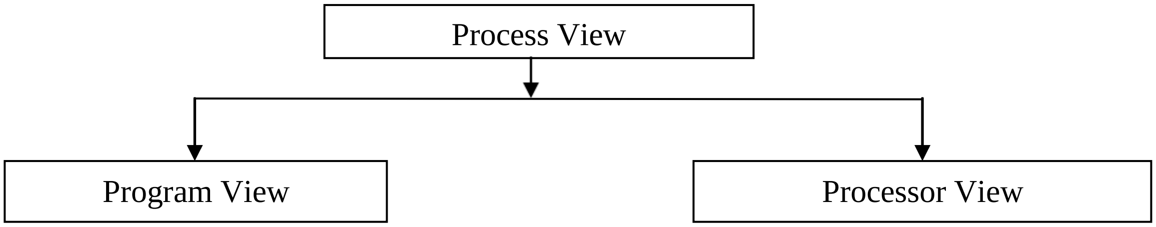
The short-term scheduler selects a process from the ready queue for execution. If a process waits for a short time for completion of an I/O task it is placed in the waiting queue, otherwise, the waiting process is swapped out of the main memory and placed in the swapped-out queue, which is implemented in the secondary memory.

Sr. No.	Long Term	Short Term	Medium Term
1.	It is a job scheduler.	It is a CPU scheduler.	It is swapping.
2.	Speed is less than short term schedular	Speed is very fast.	Speed is in between both.
3.	It controls the degree of Multiprogramming.	Less control over the degree of Multiprogramming.	Reduce the degree of Multiprogramming.
4.	Absent or minimal in time sharing system	Minimal in a time-sharing system	The Time-sharing system uses a medium-term scheduler.
5.	It selects processes from pool and load them into memory for execution.	It selects from among the processes that are ready to execute	The process can be reintroduced into memory and its execution can be continued.
6.	Process state is (new to Ready.)	Process state is (Ready to Running)	Process state is (waiting)
7.	Select a good process, a mix of I/O bound and CPU bound	Select a new process for a CPU quite frequently.	Select suspended Process

OPERATION ON PROCESSES

Process State

The principal function of a processor is to execute machine instructions residing in the main memory. We can view the process from two points of view: Program View and Processor View.



Different Process View

Program View: Its execution involves a sequence of instructions within that program. The behavior of individual process can be characterized by a list of the sequence of instructions – a ***trace*** of the process

Processor View: It executes instructions from the main memory, as dictated by changing values in the program counter register. The behavior of the processor can be characterized by showing how the traces of various processes are interleaved.

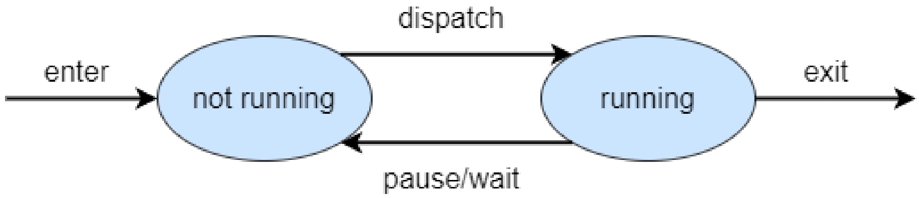
The Simple Two-State Process Model

The operating system's principal responsibility is controlling the execution of processes. This includes determining the interleaving pattern for execution and allocating resources to processes. The first step in designing an OS to control processes is to describe the behavior that we would like the processes to exhibit.

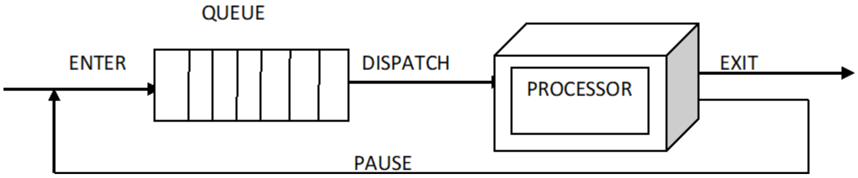
We can construct the simplest possible model by observing that, at any time, a process is either being executed by a processor or not. In this model, a process may be in one of two states: **Running or Not Running**, as shown in the figure Two-State Process Model **State transition Diagram**. When the OS creates a new process, it creates a Process Control Block for the process and enters that process into the system in the Not Running state. The process exists, is known to the OS, and is waiting for an opportunity to execute.

From time to time, the currently running process will be interrupted and the dispatcher portion of the OS will select some other process to run. The former process moves from the Running state to the Not Running state, and one of the other processes moves to the Running state.

From this simple model, we can already begin to appreciate some of the design elements of the OS. Each process must be represented in some way so that the OS can keep track of it. That is, there must be some information relating to each process, including the current state and location in memory; this is the Process Control Block. Processes that are not running must be kept in some sort of queue, waiting their turn to execute.



Two-State Process Model State transition Diagram



Two-State Process Model Queuing Diagram

Process Creation

When a new process is to be added to those currently being managed, the OS builds the data structures that are used to manage the process and allocates address space in the main memory to the process. These actions constitute the creation of a new process. Four common events leading to the creation of a process. In a batch environment, a process is created in response to the submission of a job. In an interactive environment, a process is created when a new user attempts to log on. In both cases, the OS is responsible for the creation of the new process. An OS may also create a process on behalf of an application. For example, if a user requests that a file be printed, the OS can create a process that will manage the printing. The requesting process can thus proceed independently of the time required to complete the printing task.

Traditionally, the OS created all processes in a way that was transparent to the user or application program, and this is still commonly found with many contemporary operating systems. However, it can be useful to allow one process to cause the creation of another. For example, an application process may generate another process to receive data that the application is generating and to organize those data into a form suitable for later analysis. The new process runs in parallel to the original process and is activated from time to time when new data are available. This arrangement can be very useful in structuring the application. As another example, a server process (which may be a print server or file server) may create a new process for each request that it handles. When the OS creates a new process at the explicit request by another process, this work is known as **process spawning**.

Parent Process

When one process spawns another, the former is referred to as the **parent process**.

Child Process

The spawned process is referred to as the **child process**.

Typically, the “related” processes need to communicate and cooperate. Achieving this cooperation is a difficult task for the programmer.

Process Termination

Any computer system must provide a means for a process to indicate its completion. A batch job should include a Halt instruction or an explicit OS service call for termination. In the former case, the Halt instruction will generate an interrupt to alert the OS that a process has been completed. For an interactive application, the action of the user will indicate when the process is completed. When the parent process terminates, the OS may automatically terminate all its children. For example, in a time-sharing system, the process for a particular user is to be terminated when the user logs off or turns off his or her terminal. On a personal computer or workstation, a user may quit an application (e.g., word processing or spreadsheet). All of these actions ultimately result in a service request to the OS to terminate the requesting process.

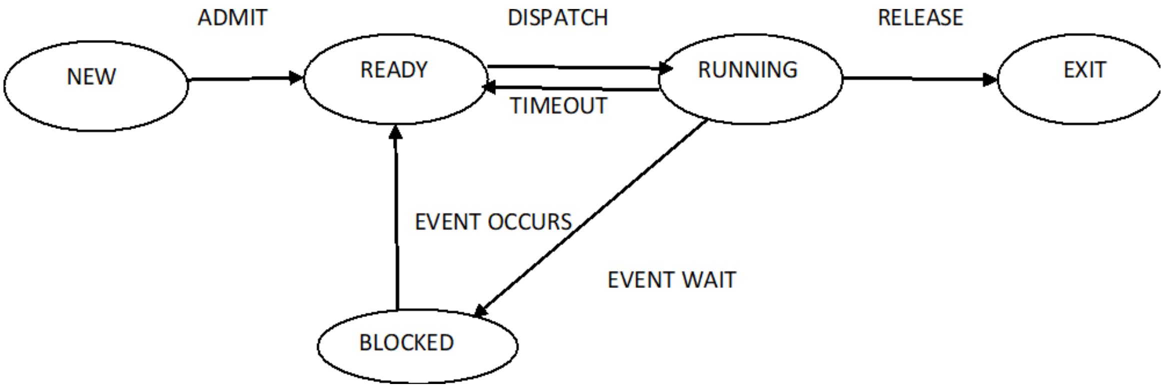
Five State Process Model

A process is a program in execution. In the five-state process model, there are five states of a process. A process may be in any one of the states during its lifetime. First of all, a process arrives into the system for its execution, then it becomes ready for execution and, then it gets the attention of the processor, and at last, the process is terminated after the completion of its execution. When the parent process terminates, the operating system may automatically terminate all its children.

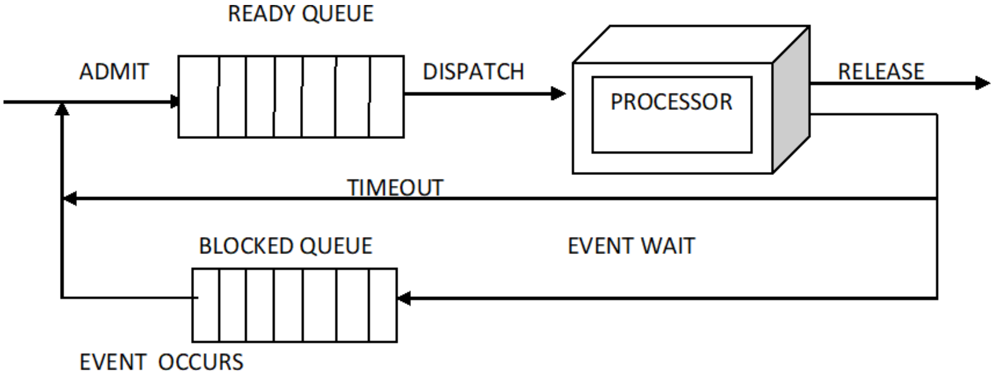
The queue is a first-in-first-out list and the processor operates in a **round-robin** fashion on the available processes (each process in the queue is given a certain amount of time, in turn, to execute and then returned to the queue, unless blocked). Some processes in the Not Running state are ready to execute, while others are blocked, waiting for an I/O operation to complete.

Thus, using a single queue, the dispatcher could not just select the process at the oldest end of the queue. Rather, the dispatcher would have to scan the list looking for the process that is not blocked and that has been in the queue the longest.

A more natural way to handle this situation is to split the Not Running state into two states: Ready and Blocked. We have added two additional states that will prove useful. The five states in this **new** diagram are as follows:



(i) **Five-State Process Model** State Transition Diagram



(ii) **Five-State Process Model** Queuing Diagram

So, the process either be in any of one of the below-mentioned states -:

1. **NEW**: The process is said to be in the NEW state when it arrives into the system for its execution. In other words, the process enters the JOB POOL of the system. A process that has been newly created but has not admitted yet into the pool of executable processes by the OS. Typically, a new process has not yet been loaded into the main memory, although its process control block has been created.

2. **READY**: The process is supposed to be in the READY state when it is ready to get the attention of the CPU In other words, the process enters the READY QUEUE of the system. A READY process is that, which is prepared to execute when given the opportunity.

3. **RUNNING**: A process is supposed to be running, when it gets the attention of the CPU for its execution. In other words, the process is being executed by the CPU in this stage.

4. **WAITING**: A process is said to be waiting when the process is blocked for some time due to some reason. In other words, the process switches (jumps) from the RUNNING stage to the BLOCKED stage for some time. A process that cannot execute until some event happens, the event may be the completion of an Input or Output operation. *Waiting* is a frequently used alternative term for *Blocked* as a process state. There are various reasons for the blocking of a process like I/O Required, Time Slice Elapses, Higher Priority Job Arrives, etc.

5. **TERMINATED**: A process is supposed to be in the TERMINATED state when the process completes its execution successfully. A process which released from the pool of executable processes by the Operating System, either because it paused or because it was abandoned for some reason.

There are two queues now: ready queue and blocked queue

When the process is admitted in the system, it is placed in the ready queue and when a process is removed from the processor, it is either placed in the ready queue or a blocked queue (depending on circumstances). If event time out occurs, then it moves to ready queue, and if event wait occurs then it moves to blocked queue. When an event occurs, all the processes waiting on that event are moved from the blocked queue onto the ready queue.

The New and Exit states are useful constructs for process management. The New state refers to a process when it has been just defined. For example, if a new user starts to log onto a system or a new batch job is submitted to the OS for execution, the OS can define a new process in two stages. First, the OS performs the necessary housekeeping chores. An identifier is associated with the process. Any tables that will be needed to manage the process are allocated and built. At this point, the process is in a New state. This means that the OS has performed the necessary actions to create the process but has not committed itself to the execution of the process. For example, the OS may limit the number of processes that may be in the system for reasons of performance or main memory limitation. While a process is in the new state, information concerning the process that is needed by the OS is maintained in control tables in the main memory. However, the process itself is not in the main memory. That is, the code of the program to be executed is not in the main memory, and no space has been allocated for the data associated with that program.

While the process is in the New state, the program remains in secondary storage, typically disk storage Similarly, a process exits a system in two stages. First, a process is terminated once it reaches its usual completion point, when it aborts due to an unrecoverable error, or when another process with the appropriate authority causes the process to abort. Termination moves the process to the exit state. At this point, the process is no longer eligible for execution. The tables and other information associated with the job are temporarily preserved by the OS, which provides time for auxiliary or support programs to extract any needed information. For example, an accounting program may need to record the processor time and other resources utilized by the process for billing purposes. A utility program may need to extract information about the history of the process for purposes related to performance or utilization analysis. Once these programs have extracted the needed information, the OS no longer needs to maintain any data relating to the process and the process is deleted from the system. Memory committed to existing processes. This limit assures that there are not so many active processes as to degrade performance.

MULTI-THREADED PROGRAMMING

A thread is a single sequential flow of control within a program. A process is defined sometimes as a ***heavyweight process*** and a thread is defined as a ***lightweight process***. A thread belongs only to one process. It is a unit of computation associated with a particular heavyweight process, using many of the associated process's resources. It has a minimum internal state and a minimum of allocated resources. Threads can share the same resources (files, memory space, etc) which are in that process. Thread can be created faster as compare to the process. Threads are widely used in real-time operating systems and modern operating systems. Each thread has its control block, with a state (Running/Blocked/etc.), saved registers, instruction pointer. Threads improve application performance through parallelism. This concept is useful in a server-client environment. A process can have a single thread or multiple threads. In multiple threads, each thread is associated precisely to one process and it always inside of the process, which means no thread can exist outside a process.

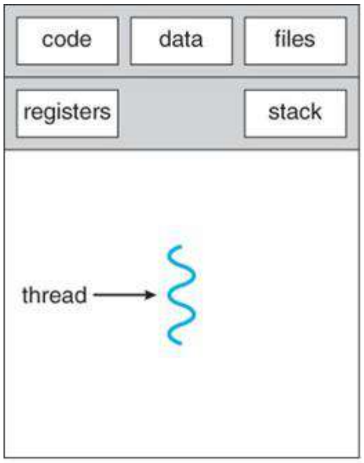


Figure: Representation of Single Thread Process

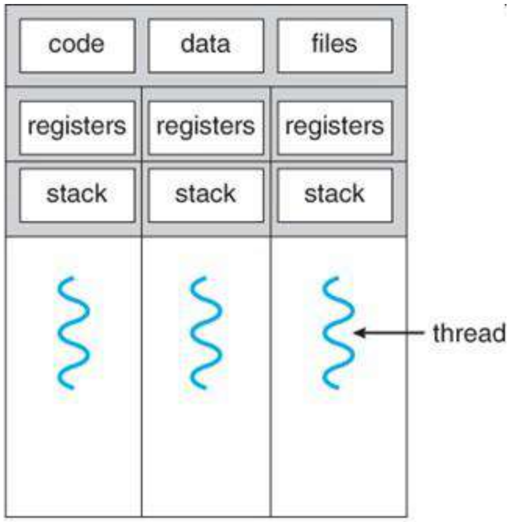


Figure: Representation of Multi-Thread Process

Advantages of Threads

- Thread minimizes context switching time.
- Threads help in the parallel execution of an application on shared-memory multiprocessors.

- The benefits of multi-threading can be greatly increased in a multiprocessor architecture.
- It is more cost-effective to create and context switch threads.
- Threads are dependent on each other so it has efficient communication.
- Threads improve application performance through parallelism.

Threads can be created at the user level and the kernel level so it is implemented by two ways:

(i) User Level (ii) Kernel Level

(i) User Level

In the user-level thread, all of the work of thread management is done by the application and the Kernel does not aware of its existence. In user space, the thread library contains code for creating and destroying threads, message passing, data, scheduling of thread execution, etc.

Characteristic of User Level Threads

- It is generally fast to create and easy to manage.
- User-type threads can run on any OS.
- A multithreaded application cannot take advantage of multiprocessing.
- Scheduling can be application-specific.

(ii) Kernel Level

In Kernel-level thread, all of the work of thread management is done by the Kernel. Kernel threads are directly supported by the operating system. The kernel maintains context information for the process as a whole and individual threads within the process. In Kernel space, thread library contains code for creating and destroying threads, message passing, data, scheduling of thread execution, etc.

Characteristic of Kernel Level threads

- In the kernel-level thread, scheduling is on a thread basis.
- It is generally slow to create.
- The kernel can schedule multiple threads from the same process on multiple processes.

- Kernel routines themselves can be multithreaded.
- Context switching between threads is time-consuming.

Comparison between process and thread

Process	Thread
The process is termed the heavyweight process	Thread is termed lightweight process
Processes are independent of one another	Threads are not independent of one another
In multiple process implementations, each the process executes the same code but has its memory and file resources	All threads can share the same set of open files, child processes.
It takes more time to create a new process.	It takes less time to create a new thread.
It takes more time to terminate a process	It takes less time to terminate a thread
While context switching needs the interface with the operating system.	In the thread, switching does not need to call an OS and cause an interrupt to the kernel.
It takes more time to switch between two processes	It takes less time to switch between two threads

Difference between User-level thread and Kernel level thread

User Level Thread	Kernel level thread
It is maintained at the user level.	It is maintained at the Kernel level.
User-level threads are quicker to create	Kernel level threads are not quickly created
It is easy and speedily managed	It is slowly managed.
It runs on any Operating System	These are specific to Operating System.
Implemented by a thread library at the user level	OS support directly to Kernel threads.

Daemon

A Daemon is a system process. It is created at boot time and keeps executing in the background. A daemon is created to perform specific tasks. It gets activated automatically when a request is received for performing a particular task. After completion of the task, it again goes back to the background.

MULTI-THREADED MODELS

Some operating system allows the facility of both level threads user-level thread and Kernel level thread. An example of this combined approach is Solaris. Multithreading permits the accomplishment of multiple parts of a program run in parallel. These parts of a program are called threads. It was also known as the lightweight process. By using multitasking and multithreading, CPU utilization increased.

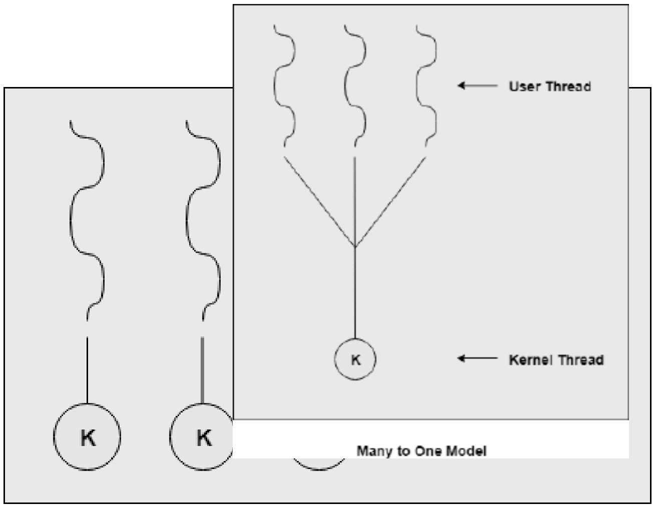
Multithreading models are 3 models namely:

one-one model, many to one model, and many to many models which are described below:

One to One Model

There is 1 to1 association between both the thread's user-level thread and kernel-level thread. This model offers more concurrency than the M:1 model. It also permits another thread to complete its work when a thread makes a blocking system call. A major disadvantage of this is when a user-level thread is created then the requirement of corresponding kernel thread. Therefore a lot of kernel-level threads are required which is a burden on the system, but there is a limit on the number of threads in the system.

A diagram that reveals the one-to-one model is given below –



One to One Model

Many to One Model

The many to one model, in this model many of the user threads maps to a single kernel thread. This model is relatively efficient as compare to the 1:1 model as in this the user space was managed by the thread management.

A disadvantage of this model is that when a thread blocking system calls then it blocks the whole process. Another disadvantage is that multiple threads cannot run simultaneously because one thread can access the kernel at a single time.

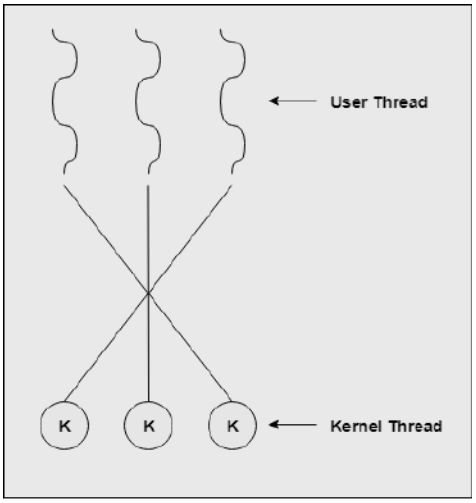
A given below diagram that shows the many to one model:-

Many to Many Model

In the many-to-many model, any number of user threads correspond with any number of kernel-level threads. There is no limit. So there is no disadvantage as compared to other models. These threads can execute simultaneously on a multiprocessor.

CPU SCHEDULING

Scheduling refers to the set of policies and mechanisms that an OS supports for determining the order of execution of pending jobs and processes. A scheduler is an OS Module that determines the next pending job to be admitted into the system for execution or the next ready process to be dispatched to RUN state.



Many to Many Model

Introduction

In the multiprogramming OS, the method or procedure for switching the CPU among multiple processes is called **CPU scheduling**. The CPU scheduler is a part of an OS, which is responsible for CPU scheduling. When a process performs an I/O-related task, it does not use certain resources, such as CPU and these resources remain idle.

CPU scheduling enables processes to utilize idle resources by assigning them to other processes. Whenever CPU becomes idle the CPU scheduler chooses a process among the processes which are in the ready queue and sends the process to the CPU for execution.

For example, process A is running and needs to perform an I/O-related task. Process A does not need the CPU while performing the I/O-related task. Process scheduler changes the state of process A from running to waiting and enables process B to use the CPU.

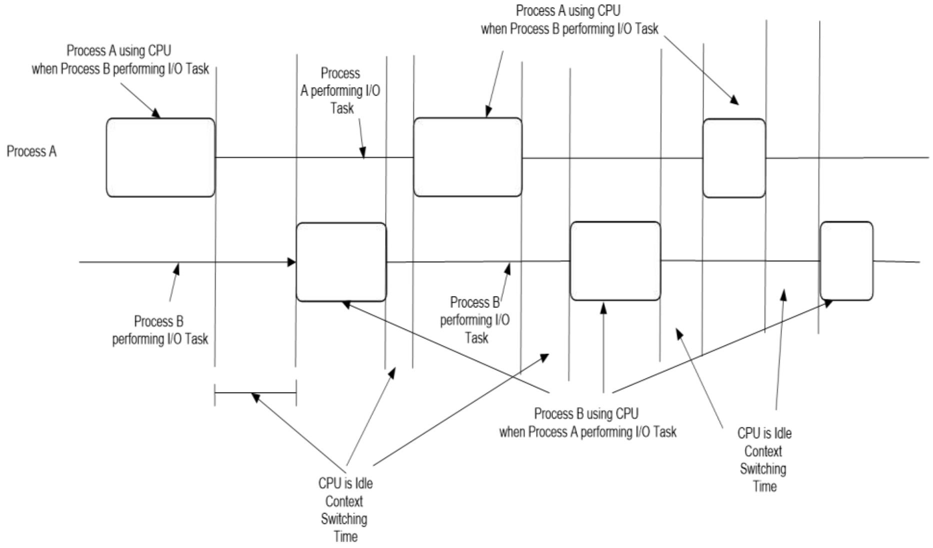


Figure: CPU Scheduling

For example, in the process of CPU scheduling, two processes, A and B, share the CPU times. When process A uses the CPU, process B performing I/O operations, and when the CPU is accessed by process B, Process A performs I/O operations.

Goals of Scheduling

(i) To optimize the utilization of system resources.

(ii) To ensure that more critical processes get priority over others processes.

(iii) To provide as fair a deal as possible to all jobs and processes which are pending.

When a process, in the ready state, is allocated the CPU in the place of a partially done process, the value of the various field of the partially done process, such as process state and I/O status is updated in the Process Control Block of the partially done process. The OS marks the partially done process as ready and sends it back to the ready queue. The OS reads the data from the process control block of the ready process and allocates CPU to it and marks the process state running. This is called **context switching**.

In other words, “Context switching” refer to the process of transferring control of the CPU from the currently running process to another process from the ready queue.

For example, a time-out has occurred for process A. Process A will be released. Process B is to be allocated the CPU. The steps performed by the **dispatcher** in context switching are:

- It retrieves and analyzes information about process A regarding program counter,memory, and registers.
- It updates the PCB of process A by writing the new values of the various fields of PCB. This includes changing the state of process A from running to ready or waiting.
- It moves process A to the appropriate queue. If the state of process A is changed into a waiting state, it is inserted in the waiting queue. If process A is transformed into the ready state, it is inserted into the ready queue.
- It retrieves and analyzes information about process B from the PCB of process B regarding program counter, memory, and registers.
- It allocates the CPU time to process B
- It restores the environment values, such as program counter, memory, and registers of process B.

The context switching is pure overhead. The extent of this overhead depends on the size of the process content. The larger the process content, the higher will be the context-switching overhead.

CPU SCHEDULING CRITERIA

Factors for measuring the performance of an operating system

There are certain factors on which the performance of an operating system depends. The efficiency and overall performance of the operating system can be measured in terms of the following factors:

(a) **CPU utilization**: CPU utilization refers to the usage of the processor during the execution of a process.CPU utilization may vary from 40% to 90%.CPU should remain as busy as possible. So, CPU utilization should be the maximum for the better efficiency of a system.

(b) **Throughput**: Throughput may be defined as the total number of processes completed per unit of time. It is the measure of work done by the CPU. It is expressed in terms of the number of jobs done in a given unit of time. It is important to know that the value of throughput does not depend only on the capability of the system but also on the nature of jobs, so, it may vary accordingly. If the jobs are CPU bound then the throughput will be less and if the jobs are I/O bound, the throughput will be high.

If the jobs are long and heavy, the throughput may be one or two processes per hour. On the other hand, if the jobs are short and light, the throughput may be 100 processes per hour. However, it should be as maximum as possible.

(c) **Turnaround time**: Turnaround time is the time which is the difference between the time of submission or entered and the time of ending of the job. It is a metric for batch systems. Turnaround time is the sum of the following components:

□ The time spends in waiting for entry into the system.
□ Total time spent in the ready queue.
□ Total time spent in the device queue.
□ Time spent in the execution of the process.
□ Time spend in doing the Input/Output operation.

However, turnaround time should be as minimum as possible.

(d) **Response time:** Response time is defined as the difference between the time of submission of the job for processing and the time when it gets the first response of the system. It is considered the best metric for interactive systems.

(e) **Waiting time:** Waiting time may be defined as the sum of intervals for which a process has to wait in the ready queue. It should be minimum for the better efficiency of the system. The scheduling strategies try to minimize the waiting time for the processes.

Average Turnaround Time: The turnaround time of a process is the total time elapsed from the time the process is submitted to the time the process is completed. It is calculated as

Turnaround time (TAT) = Process finish time (T_1) – Process Arrwal time (T_0)

Average Turnaround Time =

The lower the average turnaround time, the better it is.

Average Waiting Time: Waiting time of a process is defined as the total time spent by the process while waiting in a ready state or suspended state.

Waiting time (WT) = Turnaround time (TAT) – Actual Execution time (□t)

Average Waiting Time =

The lower the average waiting time, the better it is.

For the better performance and efficiency of a system the CPU utilization should be ***maximum***, Response time should be ***minimum***, Waiting time should be ***minimum***, Throughput should be ***maximum*** and the Turnaround time should be ***minimum***.

DISPATCHER

The dispatcher is a module of an OS that gives control of the CPU to the process selected by the scheduler. The dispatcher is responsible for the context switching. The time taken by the dispatcher for halting the running process and starting the process selected by the scheduler for executions is called **dispatch latency**.

SCHEDULING STRATEGIES

Scheduling strategies refer to the various algorithms used to choose a particular process from among the various processes in the ready queue.

There may be several processes waiting for the attention of the CPU in the ready queue. The operating system chooses a particular process from the ready queue and allocates the processor to it. So, the operating system requires a mechanism to decide, which process should be chosen next for execution.

There are two types of jobs:

□ **CPU Bound Jobs:** The jobs, that spend more time performing CPU operations and less I/O operations are called CPU-bound jobs/processes.

□ **I/O Bound Jobs:** The jobs that spend more time performing I/O operations and less time doing CPU operations are called I/O bound jobs/processes.

CPU Scheduling can be **preemptive** or **non-preemptive**.

In **preemptive scheduling**, the scheduler removes the running process from the CPU before its completion so that another process can run. In other words, in preemptive scheduling, the running process only gives up the control of the CPU voluntarily.

In **non-preemptive scheduling,** nothing can remove a process from utilizing CPU time until it completes or a time-out occurs.

There are several strategies used for *CPU Scheduling,* called *Scheduling Strategies* like:

1. FIRST COMES FIRST SERVE (FCFS)
2. SHORTEST JOB FIRST (SJF) (Preemptive and Non-preemptive)
3. PRIORITY SCHEDULING (Preemptive and Non-preemptive)
4. ROUND ROBIN SCHEDULING (RR SCHEDULING)

The detailed description of the scheduling strategies is as follows:

First Come First Serve (FCFS)

It is the simplest of all the scheduling algorithms. It is purely non-preemptive scheduling. The key concept of this algorithm is:

"allocate the CPU to the processes in the order in which they arrive".

According to this algorithm, the process which arrives first will get the CPU before any other process. Same way, the process which arrives as a second, will get the CPU after the first process and so on. It assumes the Ready Queue as the FIFO QUEUE. When a process completes its execution, the CPU is allocated to that process which is the first process in queue i.e the FRONT of the queue. If a new process arrived then it enters in REAR of the queue.

When a process starts running then it is removed from the queue. This algorithm is NON-PREEMPTIVE by default. This means once the CPU is assigned to a process, that process keeps the CPU till the end of its execution.

Advantages:

The code for FCFS scheduling is simple to write and understand.

It is suitable for Batch systems.

It is considered to be a fair policy, as the job which arrives first will get the CPU first.

Disadvantages:

If shorter jobs arrive after the longer jobs, then the waiting time will be large.

It is not suitable for Time-sharing systems.

Low CPU utilization, because all the other processes wait for one long job to get off the CPU**.**

This algorithm is never recommended whenever performance is a major issue.

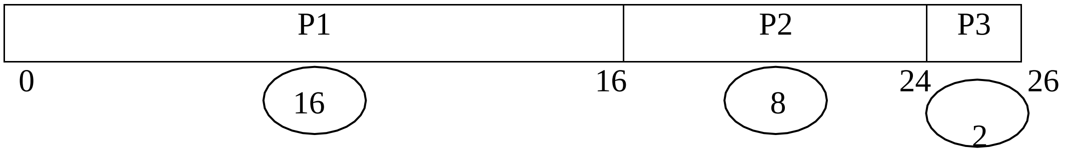
Example:

Consider the following snapshot of processes that arrive at a different time with the CPU-burst in milliseconds.

Note: Ignoring the arrival time in non-preemptive scheduling, the lowest number has the highest priority and time slice 2ms.

Process ID	**Arrival Time (T_0) ms**	**Priority**	**Next CPU Burst Time (Δt) ms**
P1	0	2	16
P2	1	3	8
P3	2	1	2

The Gantt chart of execution of the processes according to FCFS is the following:



Waiting time for P1 = 0 ms

Waiting time for P2 = 16 ms

Waiting time for P3 = 24 ms

Table depicting performance of FCFS

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	16	16	16	0
P2	0	8	24	24	16
P3	0	2	26	26	24
				66	40

As mentioned in question ignore the arrival time in non-preemptive scheduling so, here arrival time mentioned 0

Total waiting time = 0+16+24 = 40 milliseconds

Average waiting time = 40/3 = 13.33 milliseconds.

Total turnaround time = 16+24+26 = 66 milliseconds

Average turnaround time = 66/3 = 22 milliseconds.

However, if the jobs arrive in a different order, then the waiting time and turnaround time can be reduced. So, the waiting time and turnaround time depend upon the order of the jobs in which they arrive.

Shortest Job First (SJF)

The key concept of this algorithm is

“allocate the processor to the job which has the least CPU burst time”.

The ready queue has all the processes which require the processor for their execution. According to this algorithm, the processor is allocated to that job that has the smallest CPU burst time amongst all the processes in the ready queue.

If two processes have the same CPU burst, then the processor is allocated to the process which arrives first. This algorithm can either be PREEMPTIVE or NON-PREEMPTIVE.

Advantages

- This is considered to be an optimal algorithm as it helps to achieve the minimum waiting time.
- The shorter jobs have to wait for less time as compared to the longer jobs.

Disadvantages

- There is a need for the mechanism to know about the CPU burst of all the processes in advance.
- If the shorter jobs arrive again and again, the longer jobs may wait for a longperiod.

Example

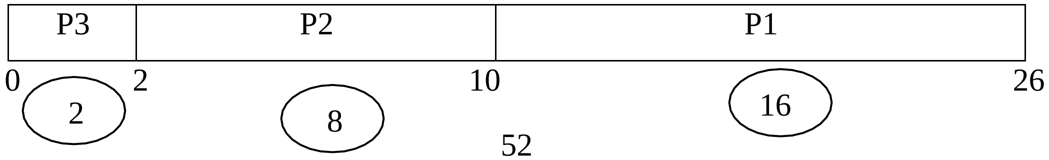
NON-PREEMPTIVE SJF

Consider the following snapshot of processes that arrive at a different time with the CPU-burst in milliseconds.

Note: Ignoring the arrival time in non-preemptive scheduling, the lowest number has the highest priority and time slice 2ms.

Process ID	**Arrival Time (T_0) ms**	**Priority**	**Next CPU Burst Time (Δt) ms**
P1	0	2	16
P2	1	3	8
P3	2	1	2

The Gantt chart of execution of the processes according to NON-PREEMPTIVE SJF is the following:



Waiting time for P1 = 10 ms

Waiting time for P2 = 2 ms

Waiting time for P3 = 0 ms

Table depicting performance of non-preemptive SJF

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	16	26	26	10
P2	0	8	10	10	2
P3	0	2	2	2	0
				38	12

Total waiting time	= 10+2+0	= 12 milliseconds
Average waiting time	= 12/3	=4 milliseconds.
Total turnaround time	= 26+10+2	= 38 milliseconds
Average turnaround time	=38/3	= 12.6 milliseconds.

PREEMPTIVE SJF: It is also termed as Shortest Remaining Time Next (SRTN) and Shortest Remaining Time First(SRTF).

Consider the set of the following processes with the CPU-burst in milliseconds.

Process ID	**Arrival Time (T_0) ms**	**Priority**	**Next CPU Burst Time (Δt) ms**
P1	0	2	16
P2	1	3	8
P3	2	1	2

Here, first of all, P1 arrives and the processor is allocated to it. After one second P2 arrives, since P2 has a smaller CPU burst as compare to P1, therefore the processor is preempted from P1 and is allocated to P2. Same way, after one second, P3 arrives whose CPU burst is less than the P2, so the processor is preempted from P2 and allocated to the P3. After the execution of P3, the processor is again allocated to P2 and so on.

The Gantt chart of execution of the processes according to PRE EMPTIVE SJF is the following:

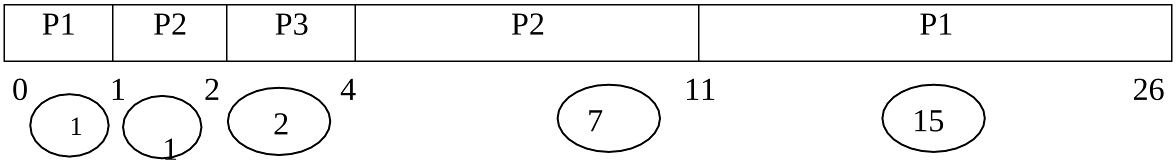


Table depicting performance of preemptive SJF

Process ID	Arrival Time (T_0) ms	Next CPU Burst Time (Δt) ms	Finish Time (T_1) ms	Turnaround Time TAT=T_1-T_0 (ms)	Waiting Time= TAT-Δt (ms)
P1	0	16	26	26	10
P2	1	8	11	10	2
P3	2	2	4	2	0
				38	12

Total waiting time	**= 10 + 2 + 0**	**= 12 milliseconds**
Average waiting time	=12/3	= 4 milliseconds.
Total turnaround time	= 26 + 10 + 2	= 38 milliseconds
Average turnaround time	=38/3	= 12.6 milliseconds.

In the case where it is not possible to know the CPU time for each process, this is estimated using predictors:

– $P_n = aO_{n-1} + (1-a)P_{n-1}$ where

- O_{n-1} = previous service time
- P_{n-1} = previous predictor
- a is within [0,1] range

– If a = 1 then P_{n-1} is ignored

– P_n is dependent upon the *history* of the process evolution

Priority Scheduling

Priority scheduling is the scheduling mechanism in which each process in the system is assigned a priority. The processor is allocated to the processes according to their priority. The key concept of this algorithm is

“***Allocate the processor to the process which has higher priority”***.

The ready queue is assumed to be a priority queue in which each process is assigned a priority. First of all, the processor is allocated to the process having higher priority and then to the process having lower priority and so on. Priority scheduling can be of two types:

PREEMPTIVE or NON-PREEMPTIVE

Example:

NON-PREEMPTIVE PRIORITY SCHEDULING

Consider the set of the following processes with the CPU-burst in milliseconds.

Note: Ignoring the arrival time in non-preemptive scheduling, the lowest number has the highest priority and time slice 2ms.

Process ID	**Arrival Time (T_0) ms**	**Priority**	**Next CPU Burst Time (Δt) ms**

P1	0	2	16
P2	1	3	8
P3	2	1	2

The Gantt chart for the execution of the programs is as follows:

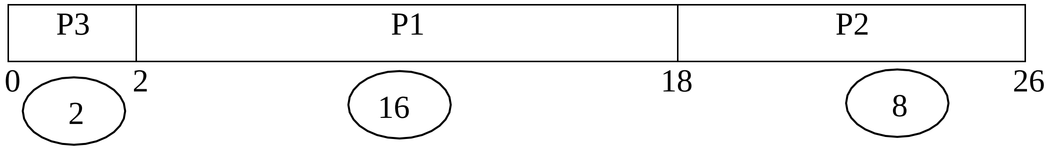


Table depicting performance of non-preemptive priority scheduling

Process ID	Arrival Time (T_0) ms	Next CPU Burst Time (Δt) ms	Finish Time (T_1) ms	Turnaround Time TAT=T_1-T_0 (ms)	Waiting Time= TAT-Δt (ms)
P1	0	16	18	18	2
P2	0	8	26	26	18
P3	0	2	2	2	0
				46	20

Waiting time for P1 = 2 ms

Waiting time for P2 = 18 ms

Waiting time for P3 = 0 ms

Total waiting time = 2 +18 + 0 = 20 milliseconds

Average waiting time =20/3 = 6.6 milliseconds.

Total turnaround time = 18 + 26 + 2 = 46 milliseconds

Average turnaround time =46/3 = 15.3 milliseconds.

PREEMPTIVE PRIORITY SCHEDULING

Consider the set of the following processes with the CPU-burst in milliseconds.

Process ID	Arrival Time (T_0)	Priority	Next CPU Burst

	ms		**Time (Δt) ms**
P1	0	2	16
P2	1	3	8
P3	2	1	2

The Gantt chart for the execution of the programs is as follows:

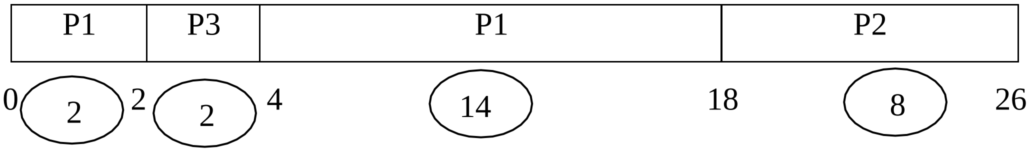


Table depicting performance of preemptive priority scheduling

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	16	18	18	2
P2	1	8	26	25	17
P3	2	2	4	2	0
				46	19

Total waiting time = 2 + 17 + 0 = 19 milliseconds
Average waiting time =19/3 = 6.3 milliseconds.
Total turnaround time = 18 + 25 + 2 = 45 milliseconds
Average turnaround time =45/3 = 15 milliseconds.

The problem in Preemptive Priority Scheduling (Starvation or Blocking)

The main problem with priority scheduling is that low-priority processes can wait for a long time due to high-priority process arrivals. If an Operating system has not any precautions and just chooses the process with the highest priority, low priority processes wouldn't get CPU, as long as high priority processes are runnable. This problem is known as **Starvation.**

Solution of Starvation (AGING)

Aging is a method of slowly increasing the priority of processes that are waiting in the system for a long time. The simplest solution is dynamic priorities. On one hand, the operating system can reduce the priority of a running process for each time quantum it used the CPU and on the other hand, it could increase the priority of other processes (which finally leads to the first situation because the boost should only be temporary) which didn't get the CPU for a certain amount of time. Whether the operating system uses one or another solution, processes have a base priority that remains unchanged, and a real priority that is used for scheduling. Often the real priority is limited to a specific range so that important process still gets the processor when they need it.

Another way to avoid this problem is static priorities. The OS has to keep a record of how long a process has used the CPU. If it reaches a certain limit, the next highest priority process is allowed to run. This is not practically good.

Round Robin Algorithm (RR Algo)

The Round Robin scheduling algorithm is designed especially for the Time Sharing systems. It can be considered as FCFS scheduling along with the preemption. The processor is allocated to a process for a fixed amount of time called, **TIME SLOT** or **TIME QUANTUM**, or, **TIME SLICE**. It is a purely preemptive algorithm. Likewise FCFS, the processor is allocated to the processes in the order in which they arrive, but the ed from the process after the time slice is over, and the processor is allocated to the next process in the ready queue. A time quantum generally varies from 10milliseconds to 100 milliseconds. The ready queue is assumed to be a circular queue. Theshort-term scheduler goes on allocating the processor to the processes in the ready queue for a fixed amount of time.

However, if a process has its CPU burst less than the time quantum, then the process releases the CPU voluntarily (itself). Otherwise, the processor is allocated to the process in the ready queue for a fixed amount of time (time quantum) and after that CONTEXT SWITCHING take place, and the processor is allocated to the next process in the ready queue, keeping the former process at the end of the ready queue. The RR algorithm is PREEMPTIVE by default.

The performance of the Round Robin algorithm depends on the value of time quantum or slice

- If the value of time quantum is large, then this algorithm becomes the same as FCFS.
- If the value of the time quantum is small, then the number of the context switching will be increased considerably, this is not desirable at all. It affects the system throughput adversely.
- Thus, the size of the time quantum should neither be very large nor too small forbetter efficiency.

Example:

Consider the set of the following processes with the CPU-burst in milliseconds.

Note: Ignoring the arrival time in non-preemptive scheduling, the lowest number has the highest priority and time slice 2ms.

Process ID	Arrival Time (T_0) ms	Priority	Next CPU Burst Time (Δt) ms
P1	0	2	16
P2	1	3	8
P3	2	1	2

The Gantt chart of execution of the processes according to the RR scheduling algorithm is the following. Let the time quantum be 2 milliseconds-:

P1	P2	P3	P1	P2	P1	P2	P1	P2	P1

0 2 4 6 8 10 12 14 16 18

26 2 2 2 2 2 2 2 2 2 8

Table depicting performance of Round Robin scheduling

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	16	26	26	10
P2	1	8	18	17	9
P3	2	2	6	4	2
				47	21

Waiting time for P1 = 10 ms

Waiting time for P2 = 9 ms

Waiting time for P3 = 2 ms

Total waiting time = 10 + 9 + 2 = 21 milliseconds

Average waiting time = 21/3 = 7 milliseconds.

Total turnaround time = 26 + 17 + 4 = 47 milliseconds

Average turnaround time = =47/3 = 15.6 milliseconds.

Comparison between FCFS and Round Robin Scheduling Algorithm

FCFS	Round Robin
FCFS is purely non-preemptive Scheduling	It is purely preemptive Scheduling
It has minimum overhead	It has a higher overhead as compare to FCFS as context switching occurs more.
Response time depends upon the size of the process	It offers a better response time
It is not designed for a time-sharing system	It is designed for a time-sharing system
The workload is simply processed in the order process arrived	It is similar to FCFS but it uses a time slice which means one process maximum uses

	CPU for time slice at once.

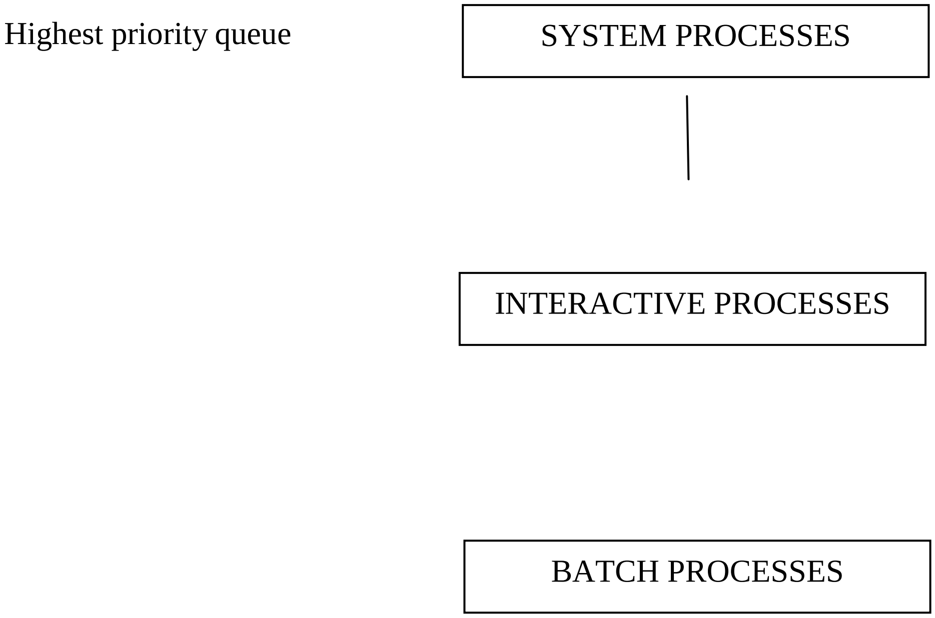
Multilevel Queue Scheduling

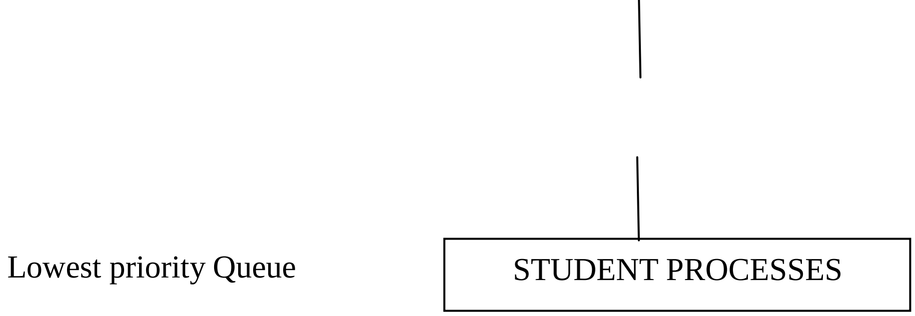
A multilevel queue scheduling algorithm is used when the processes are to be divided into different groups. Sometimes such a situation arises where two or more processes have different response time requirements, so there is a need for a different kind of schedule for both processes.

A multilevel queue scheduling division of the ready queue into numerous separate queues. The processes are permanently allocated to a particular queue, based on the properties of the process.

Sometimes there are two or more processes, among which some processes have higher priority and other having low priority. So, there is a need for a separate queue for the processes with higher priority and the processes with lower priority.

Consider an example of multilevel queue scheduling for different kinds of processes in the ready queue. There are four different queues for different processes which are made based on their priority in the system.





Lowest priority Queue

STUDENT PROCESSES

Each queue may have its scheduling criteria, based upon the nature and properties of the jobs.

Advantages

- Low scheduling overhead can be achieved by using a multilevel scheduling algorithm.

Disadvantages

- It is not considered as efficient in some of the cases, like if higher priority queues don’t become empty for a long time, then the lower priority jobs may starve.

Multilevel Feedback Queue Scheduling

It is an enhancement of multilevel queue scheduling.

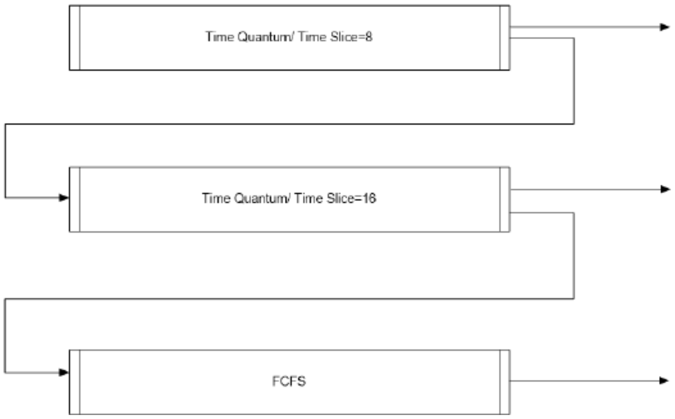
Actually, in multilevel queue scheduling, the processes are assigned a particular queue on entering the system. The processes do not move from a queue to any other queue. So, it is an inflexible approach.

Multilevel feedback queue scheduling permits a process to move between the queues. For example, if a process CPU-bound process which means it uses too much CPU then the process is moved (shifted) to a lower priority queue. Correspondingly, a lower priority job can also be shifted to a higher priority queue if it is waiting for a long time, etc.

Advantages

- It allows a process to move to any other queue.
- It is more flexible as compared to any other scheduling algorithm.

- A lower priority job, which is waiting for a long time, can be shifted to a higher priority queue.
- A higher priority job, affecting negatively the efficiency, can be moved to a lower priority queue.



Multilevel Queue Scheduling

Disadvantages

- However, it is considered to be a complex scheduling algorithm.
- While moving the processes from a queue to any other queue increases the CPU overhead.

Illustration

With reference to the following set of processes

Process ID	Arrival Time (T_0) ms	Next CPU Burst Time (Δt) ms	Priority
P1	0	29	1
P2	6	14	5

P3	8	10	3
P4	10	8	1
P5	13	6	2

Determine Average Waiting Time and Turn Around Time, for the following scheduling.

(a) First Come First Serve (FCFS)

(b) Shortest Job First (SJF) (Non-preemptive)

(c) Shortest Job First (Preemptive) or Shortest Remaining Time Next (SRTN)

(d) Priority Scheduling (Preemptive)

(e) Priority Scheduling (Non-preemptive)

(f) Round Robin (RR)

Note: Time Slice – 5 ms, the highest number has the highest priority and ignores the arrival time in non-preemptive scheduling.

(a) First Come First Serve (FCFS)

In this scheduling, jobs or processes are scheduled to run in the same order as those that have arrived in the system. It's a pure non-preemptive algorithm. Process P1. will execute first because its ID is 1, then P2, P3, P4, and P5. So its Gantt Chart is given below:

P1	P2	P3	P4	P5
29	14	10	8	6

0 29 43 53 61 67

Calculating Waiting Time and Turnaround Time

In this problem, in a non-preemptive algorithm arrival time is ignored so, waiting time for all processes is when they start to run and turnaround time is when they complete the process.

Note Arrival time 0 in this problem because in non-preemptive algorithm arrival time is ignored.

Table depicting performance of FCFS

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	29	29	29	0
P2	0	14	14	43	29
P3	0	10	10	53	43
P4	0	8	8	61	53
P5	0	6	6	67	61
				253	186

Average Waiting Time = (0+29+43+53+61)/5=37.2 ms

Average Turnaround Time = (29+43+53+61+67)/5=50.6ms

(b) Shortest Job First (SJF)

In shortest job first, job to be dispatch will be the one, which happens to be the shortest amongst the pending jobs. This is non-preemptive. So a job, once scheduled, is permitted to complete its next burst. In this problem, Arrival time is ignored for non-preemptive scheduling so, P5 has the shortest next burst time hence it will run first then P4, P3, P2 & P1 will run. So its Gantt chart is

P5	P4	P3	P2	P1

0 6 14 24 38 67

6 8 10 14 29

Calculation of Waiting Time & Turnaround Time

In this problem, in a non-preemptive algorithm arrival time is ignored so, waiting time for all processes is when they start to run and turnaround time is when they complete the process.

Table depicting performance of non-preemptive SJF scheduling

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	29	67	67	38
P2	0	14	38	38	24
P3	0	10	24	24	14
P4	0	8	14	14	6
P5	0	6	6	6	0
				149	82

Note: Arrival time is 0 in this problem because non-preemptive algorithm arrival time is ignored.

Average Waiting Time = (38+24+14+6+0)/5=16.4 ms

Average Turnaround Time = (67+38+24+14+6)/5=29.8 ms

(c) Shortest Remaining Time Next (SRTN)

This is a preemptive algorithm, where the next job/process to be dispatched will be one that happens to be shortest amongst the pending jobs, at the time of making the decision. However, if a process/job arrives later, whose next burst time to be less than the remaining burst time of the currently running process, the currently running process will be preempted by the new process. The preempted process will be later re-dispatched when its remaining burst happens to be shortest amongst the pending processes.

In this problem, at 0 arrival time, only one process is there i.e. P1. So, P1 will execute first. After 6ms, a new process P2 arrives which has a 14ms burst time and P1 has 23ms (29-6) remaining burst time. So, P1 will be preempted by the P2 because P2 has the shortest burst time at 6ms. After 2ms i.e. 8ms a new process P3 arrives which has 10ms burst time and P2 has 12ms (14-2) remains burst time P3 has the shortest burst time at 8ms. So, P2 will be preemptive by P3. After 10ms a new process P4 arrives which has an 8ms burst time and P3 has 8ms (10-2) remains burst time. So there is a tie between P3 and P4. To break tie FCFS algorithm is adopted so, P3 will continue to execute. At time 13ms a new process, P5 arrived with 6ms burst time. At that time P3 has (10-(13-8)) i.e. 5ms remaining burst time which shortest than among all-time so P3 will continue. After completion of P3 at 18ms time. There is no new process arrived. Now remains burst time for a process are P1 is 23ms, P2 is 12ms, P4 is 8ms and P5 is 6ms. So amongst P5 is shortest than P4, P2 and P1 will execute. Gantt chart of this is given below:

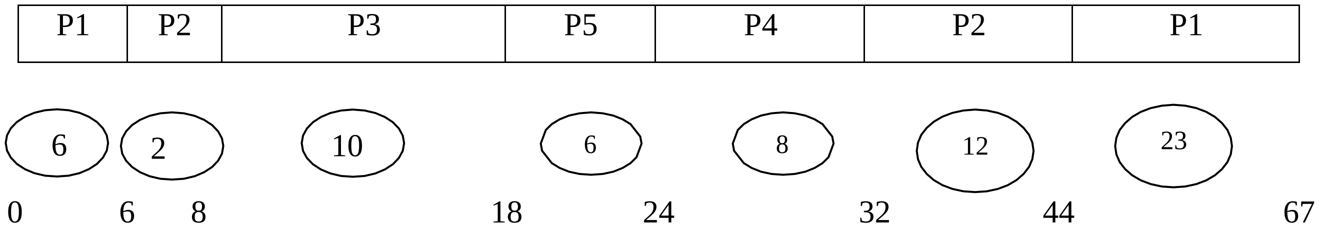


Table Depicting Performance of SRTN Algorithm

Process ID	Arrival Time (T_0) ms	Next CPU Burst Time (Δt) ms	Finish Time (T_1) ms	Turnaround Time TAT=T_1-T_0 (ms)	Waiting Time= TAT-Δt (ms)
P1	0	29	67	67	38
P2	6	14	44	38	24
P3	8	10	18	10	0
P4	10	8	32	22	14
P5	13	6	24	11	5

				148	81

Average Waiting Time =(38+24+0+14+5)/5=16.2 ms

Average Turnaround Time = (67+38+10+22+11)/5= 29.6 ms

(d) Priority Scheduling (Preemptive)

At the time of schedules, a dispatcher dispatched that process which has the highest priority amongst the processes which are waiting in the ready queue. When a process Pi is executing, if another process Pj arrived and has higher priority, then Pi will be preempted by Pj.

In this problem, at time 0ms only one process P1 is there so it is executed till a new process has arrived i.e. 6ms. After 6ms a new process P2 comes which has priority 5 which is greater than P1 priority i.e. 2 so, P1 will be preempted by P2. Now P2 will execute at 8ms a new process P3 arrive which have priority 3 but P2 have the highest priority at that time so, P2 will continue, after 10ms P4 arrive which have priority value 1 which lowest so, P2 will continue. After that at 13ms P5 comes whose priority is lower than P2. So P2 will continue till 20ms i.e. (6 + 14).

Now there are 4 processes at 20ms P1, P3, P4, P5 are ready and priorities are 1, 3, 1, 2 respectively. Among them, 3 is the highest priority. So, P3 will execute then P5. After that P1, P4 have the same priority so, there is a tie. To break the tie FCFS algorithm was adopted so P1 executes than P4. Gantt chart of this algorithm is:

P1	P2	P3	P5	P1	P4

0 6 20 30 36 59

6 67 14 10 6 23 8

Table depicting performance of a priority-based preemptive algorithm.

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	29	59	59	30
P2	6	14	20	14	0
P3	8	10	30	22	12
P4	10	8	67	57	49
P5	13	6	36	23	17
				175	108

Average Waiting Time = (30+0+12+49+17)/5=21.6 ms

Average Turnaround Time = (59+14+22+57+23)/5=35 ms

(e) Priority Scheduling (Non-preemptive)

At the time of scheduling, a process that has the highest priority amongst all the processes which are waiting in the ready queue. Once dispatched, a process Pi is allowed to complete its burst time, even if Pj another process of having higher priority becomes ready while running Pi.

In this scheduling, arrival time is ignored so amongst all processes P2 has the highest priority which will execute first. Priority of above processes is:

P2 > P3 > P5 > P1 > P4

Gantt chart is:

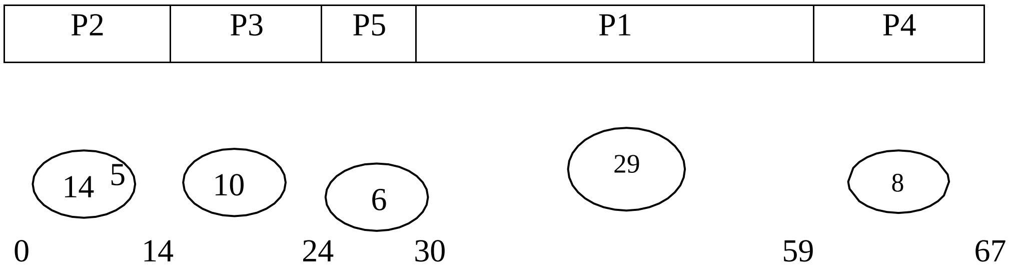


Table depicting performance of a priority-based non-preemptive algorithm.

Process ID	**Arrival Time (T_0) ms**	**Next CPU Burst Time (Δt) ms**	**Finish Time (T_1) ms**	**Turnaround Time TAT=T_1-T_0 (ms)**	**Waiting Time= TAT-Δt (ms)**
P1	0	29	59	59	30
P2	0	14	14	14	0
P3	0	10	24	24	14
P4	0	8	67	67	59
P5	0	6	30	30	24
				194	127

Note: Arrival time is 0 in this problem because non-preemptive algorithm arrival time is ignored.

Average Waiting Time = (30+0+14+59+24)/5=25.4 ms

Average Turnaround Time = (59+14+24+67+30)/5=38.8 ms

(f) Round Robin (RR)

It is purely a preemptive scheduling algorithm. A small unit of time is called a time slice or time quantum. It is the maximum time for which a process can execute at a time. New processes are added at the end of the ready queue and the head of the ready queue process Pi is dispatched for the time slice. If the process Pi has not finished its execution, then it is linked to the tail of the ready queue, and the next process in the ready queue is dispatched for the next time slice and so on.

In this problem, the time slice is 5ms so P1 will execute for 5ms but there is no more process at 5ms so, P1 again executes for the next 5ms. After 10ms P2 execute for 5ms and P3, P4, P5 so on. This process will continue till the process complete their execution. Gantt Chart is:

P1	P1	P2	P3	P4	P5	P1	P2	P3	P4	P5	P1	P2	P1

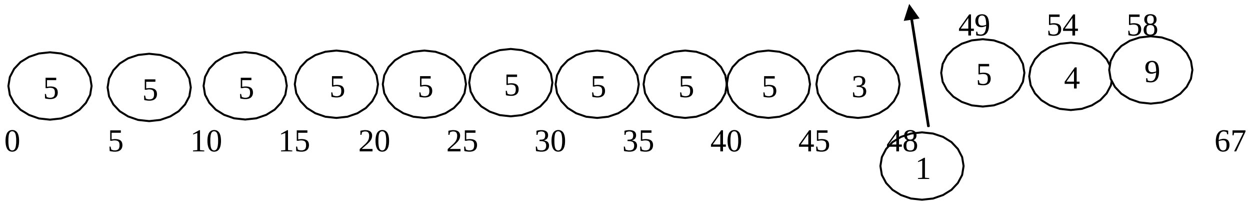


Table depicting performance of round robin algorithm

Process ID	Arrival Time (T_0) ms	Next CPU Burst Time (Δt) ms	Finish Time (T_1) ms	Turnaround Time TAT=T_1-T_0 (ms)	Waiting Time= TAT-Δt (ms)
P1	0	29	67	67	38
P2	6	14	58	52	38
P3	8	10	45	37	27
P4	10	8	48	38	30
P5	13	6	49	36	30
				230	163

Average Waiting Time = (38+38+27+30+30)/5 =32.6 ms

Average Turnaround Time = (67+52+37+38+36)/5=46 ms

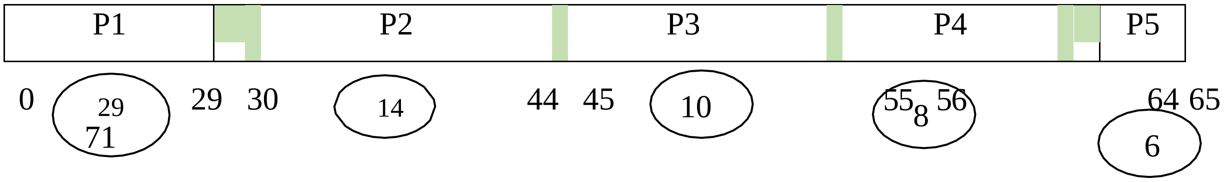
CPU Scheduling Algorithms analysis

Scheduling Algorithm	Turnaround Time	Waiting Time	Average Turnaround Time	Average Waiting Time
FCFS	253	186	50.6	37.2
SJF	149	82	29.8	16.4
SRTN	148	81	29.6	16.2
Priority(Preemptive)	175	108	35	21.6
Priority(non-	194	127	38.8	25.4

Preemptive)				
Round Robin	230	163	46	32.6

The above table depicts that SRTN scheduling gives better performance among the other scheduling.

Note: Some time context switching time is given in problem then that time is switching time and it is pure overhead. It will be increased the waiting time and turnaround time. If the context switch time is 1ms then in the above example the Gantt chart of FCFS as below:



Performance of all CPU Scheduling Algorithms

Scheduling Algorithm	CPU Utilization	Response Time	Average Waiting Time	Average Turnaround Time	Throughput
FCFS	Low	Low	High	High	Low
SJF	Medium	Medium	Medium	Medium	High
Priority	Medium	High	High	High	Low
Round Robin	High	High	Medium	Medium	Medium
Multi-Level	High	Medium	Medium	Medium	High

Points to Remember

- A program is a static entity.
- A process is an instance of a program under execution.
- The process is a dynamic entity.

- Process state is defined as the current activity of the process.
- In two states process model process has two states: Running or not running.
- In five states process model process has five states: New, Ready, Waiting Running, and Terminating.
- Each process has its Process Control Board (PCB).
- In-Process Control Board (PCB) attributes and information about the process is stored which is needed by OS to control the process.
- Schedulers are of three types: Long term Scheduler, Medium Term Scheduler, Short Term Scheduler.
- Long term scheduler is also known as a job scheduler. It selects the processes and loads them into memory. It changes process states from new to ready state.
- A short-term scheduler is also known as a CPU scheduler or dispatcher. It changes process states from ready to running state.
- The process is called the heavyweight process.
- A process can have a single thread as well as multiple threads.
- Thread is called a lightweight process.
- Threads are not independent of one another.
- Threads can be created at a user level and the kernel level.
- A context switch is switching the CPU to another process which requires saving the state of the old process and loaded the saved state for the new process.
- A Daemon is a system process that is created at boot time and keeps executing in the background.
- CPU scheduling is used to increase CPU utilization.
- In non-preemptive scheduling, once the process has been assigned to the CPU, the CPU cannot be taken away from that process until it terminates or is blocked.
- In preemptive scheduling, the CPU can be taken away from the process during execution.
- Context switching is required in preemptive scheduling.

- Throughput means how many processes the system can execute in a unit of time.

Higher the number, the better it is.

- Response Time is the time from the entry of request until the first response is produced.
- Finish time means when the process finishes its execution (T1).
- Arrival time is the time when the process arrived (T0).
- Burst Time is the estimated time the process needed CPU for execution (**Δt**).
- Turnaround time is computed by subtracting the time the process entered the system from the time it terminated (TAT= T1-T0).
- Waiting Time is time spent by the process for CPU (WT=TAT- **Δt**).
- CPU-bound process spends most of its time in CPU.
- I/O bound process spends most of its time in I/O operations.
- First Come First Serve (FCFS) is a purely non-preemptive algorithm.
- In FCFS, the CPU is allocated to the process in order of arrival.
- Shortest Job First (SJF) scheduling algorithm can be either a non-preemptive or preemptive algorithm.
- In SJF, the CPU is allocated to the process which has the smallest burst time.
- Preemptive SJF is known as Shortest Remaining Time Next (SRTN).
- In Shortest Job First (SJF) scheduling (non-preemptive and preemptive) the next CPU burst time must be known in advance.
- A priority scheduling algorithm can be either a non-preemptive or preemptive algorithm.
- In priority scheduling, the CPU is allocated to the process which has the highest priority.
- In preemptive priority scheduling, there is a big problem known as **starvation.**
- In starvation, blocking of low priority processes due to high priority jobs keep arriving one after another.
- **Aging** is a solution to starvation, after some time makes the priority of a process go up the longer it stays run-able but isn’t run.
- Round Robin is a purely preemptive algorithm.
- CPU is allocated to all processes in a queue for small-time, which is known as Time Slice, Time Quantum.

- Multilevel queue algorithms allow different algorithms to be used for various classes of processes.
- Multilevel feedback queues allow processes to move from one queue to another.

EXERCISES

1. What is a Process? Explain different states of a process.
2. Explain the Process State diagram in detail.
3. What do you mean by Process Creation and Termination?
4. What is Process Control Block?
5. Explain the following terms:

   (i) Throughput

   (ii) Waiting Time

   (iii) Turn Around Time

   (iv) Response Time

   (v) CPU utilization
6. What are the various factors for measuring the performance of an operating system?
7. Explain the Operations on Processes.
8. What do you mean by Thread and explain different types of threads?
9. Difference between Thread and Process.
10. Difference between User Level Thread and Kernel Level Thread.
11. Explain how threads can improve system performance.
12. Discuss the following terms:

    □ Process Spawning

    □ Parent Process

    □ Child Process

    □ Halt

    □ Dispatcher

    □ Ready Queue

    □ Blocked Queue

□ Daemon

13. What do you mean by CPU scheduling and why it is required?
14. What do you mean by I/O bound and CPU bound process?
15. Define the differences between preemptive and non-preemptive scheduling?
16. What are the scheduling criteria? Explain it.
17. Explain the different types of schedulers.
18. Consider the following set of processes

Process	**Arrival Time**	**Priority**	**Burst Time**
P0	0	4	20
P1	1	2	8
P2	3	1	10
P3	5	3	4
P4	8	1	2
P5	9	5	7

Calculate Average Turnaround Time, Average waiting time for the following algorithms. The time slice is 3 ms and the lowest number has the highest priority for the following scheduling.

(a) FCFS (b) SJF (Preemptive and non-preemptive)

(c) Priority (Preemptive and non-preemptive) (d) Round Robin

19. What is Multilevel queue scheduling. Why we use it?
20. Explain between long-term and short-term scheduler.
21. Explain with example the following:

(a) FCFS (b) SJF

22. Explain with example the following:

(a) Priority Scheduling (b) Round Robin Scheduling

23. Difference between FCFS and RR scheduling.

24. Consider the following set of processes

Process	**Arrival Time**	**Priority**	**Burst Time**
P1	0	3	5
P2	1	1	2
P3	1	2	8
P4	3	1	5
P5	7	4	7

Calculate Average Turnaround Time, Average waiting time for the following algorithms. The time slice is 2 ms and the highest number has the highest priority (ignore the arrival time in non-preemptive scheduling) for the following scheduling.

(a) FCFS (b) SJF (Preemptive and non-preemptive)

(c) Priority (Preemptive and non-preemptive) (d) Round Robin

25. Consider the following set of processes

Process	**Arrival Time**	**Priority**	**Burst Time**
P1	0	1	20
P2	0	5	12
P3	3	3	3
P4	4	4	18
P5	8	2	8

Calculate Average Turnaround Time, Average waiting time for the following algorithms. The time slice is 5 ms and the lowest number has the highest priority (1 ms is context switching time and ignore the arrival time in non-preemptive scheduling) for the following scheduling.

(a) FCFS (b) SJF (Preemptive and non-preemptive)

(c) Priority (Preemptive and non-preemptive) (d) Round Robin

M.Sc. (Computer Science)

SEMESTER-1

COURSE: DBMS

UNIT 1:

STRUCTURE

Objective

Introduction to Deadlock

System Model

Deadlock Characterization

Necessary Conditions For Deadlock

Deadlock Detection

Deadlocks Management

Deadlock Prevention

Deadlock Avoidance

Deadlock Dectection & Recovery

Deadlock Ignorance

Practice Excercises

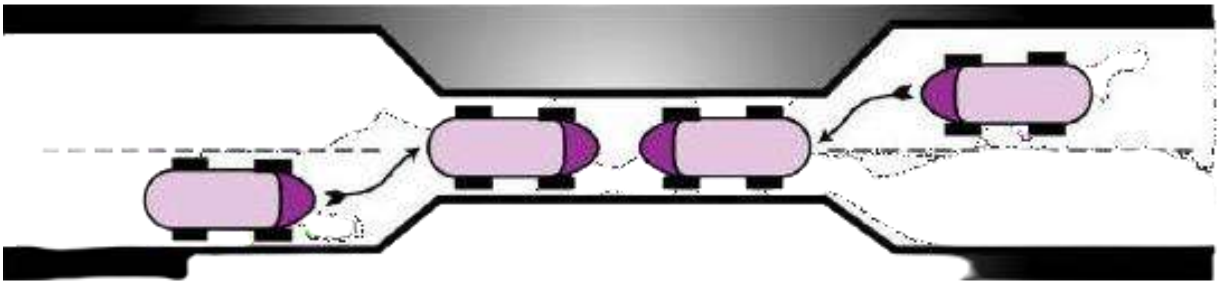
OBJECTIVES

- To understand the Deadlock Problem
- System Model
- Deadlock Characterization
- Methods for Handling Deadlocks
- Deadlock Prevention
- Deadlock Avoidance
- Deadlock Detection
- Recovery from Deadlock

INTRODUCTION TO DEADLOCK

There is a situation in which two processes wait for each other. The resources of a system may be limited or less as compared to the number of processes. In multiprogramming, numerous processes may require a predetermined number of resources. When a process requests for a resource, the Operating System checks whether the resource is available or not. If the resource is available, then it is allocated to the process. On the other hand, if the resource is not free (available), the process enters the waiting state. Sometimes it happens that the process remains to wait, for a long time because the requested resource is holding by other waiting processes. This situation leads to **DEADLOCK**. It is a condition, wherein a set of processes are waiting forever for the resources, held by each other. None of them can proceed with its execution.

In routine life, a traffic jam on the narrow bridge is an example of Deadlock, when cars came from both side, but only a single car crossed from the bridge at a time and both car drivers refuge to back the cars as shown in the figure below:

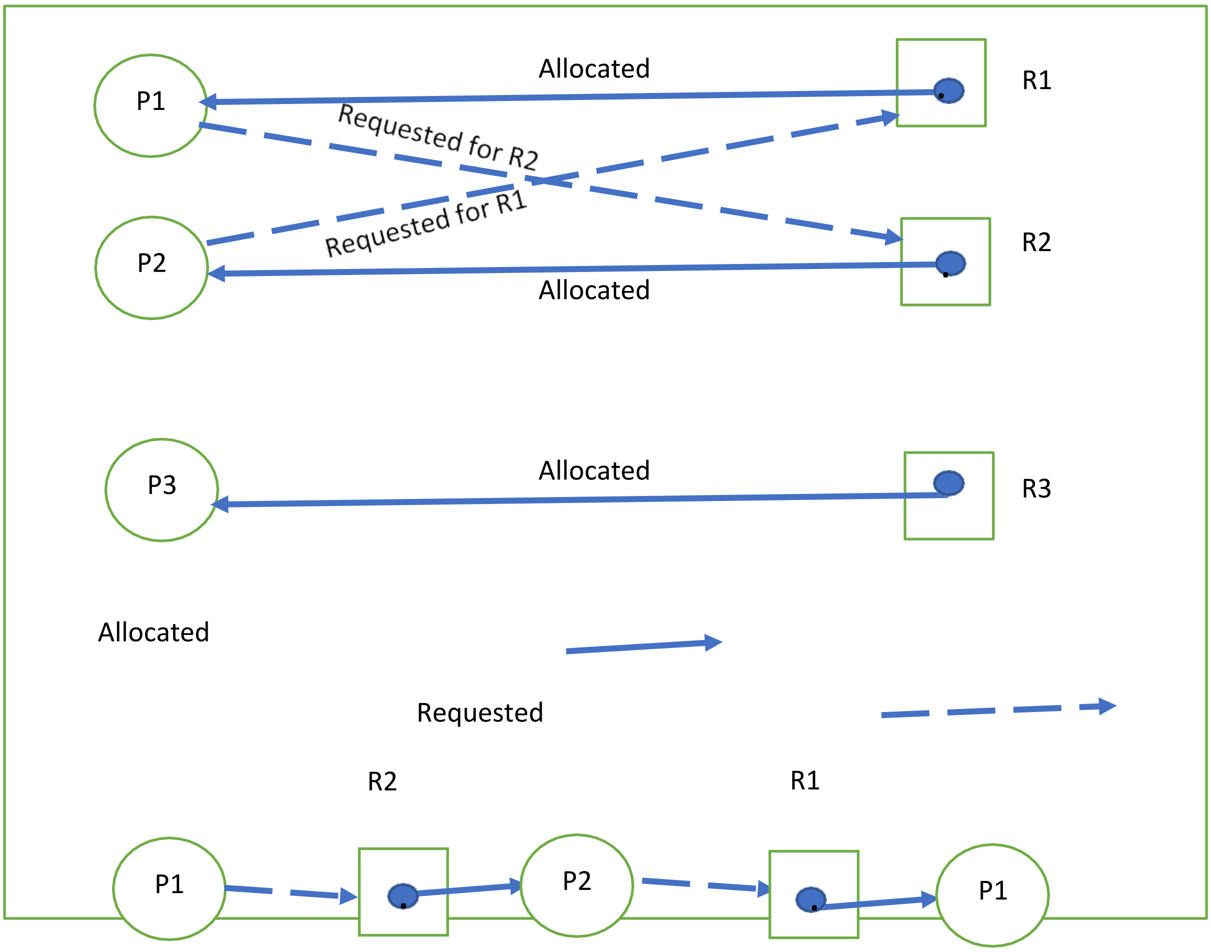


Example: Consider a scenario where there are three processes P1, P2, P3, and three resources R1, R2, R3. P1 requires two resources R1 and R2 for its execution. Also, P2 requires two resources R1 & R2 for its execution .P3 requires only R3. Initially, R1 is allocated to P1, and R2 is allocated to P2.

Here, (i) R1 is allocated to P1 and P1 is requesting R2.

(ii) R2 is allocated to P2 and P2 is requesting for R1.

(iii) R3 is allocated to P3.



SYSTEM MODEL

In the system, there are n number of processes namely p_1,p_2,p_3, …… $p_{n-1,}$ $p_{n,}$ and m number of resource types namely R_1, R_2,…..R_{m-1}, $R_{m.}$ In the System model, R resources are to be circulated among some processes P. The resources are then divided into many types, each consisting of some definite quantity of identical instances which is represented as w. The main example of resource types is *CPU cycles, memory space, Input-Output devices* such as keyboards, printers, and CD-DVD drives, directories, and files. Each resource type has instances. Instances mean the number of resource types that represent as w. Each resource type R_i has W_i instances When a system has 4 printers, then the resource type printer got four instances. Each process uses a resource as the following sequence:

- **Request:** Process request for the resources. If resources are available then the system allocates to process otherwise process waits for resources.
- **Use**: Then process used the resources when the system allocates them.
- **Release:** Process used the resources and then released them.

DEADLOCK CHARACTERIZATION

Necessary Conditions For Deadlock

The following four conditions must hold simultaneously, for a deadlock to occur:

1. Mutual exclusion

Mutual exclusion implies that a resource can be utilized exclusively by only one process at a time in a non-shareable mode. If any other process wants to use that resource, then it must have to wait, until it is released by the former process. For example, P1 holds R1 and P2 holds R2, both in mutually-exclusive mode.

2. Hold and wait

It implies that there exists a process that is holding at least one resource which is waiting for another resource or resources, which is being held by other processes. It implies that there exists a process that must be holding some resources (at least one) in a non-sharable mode and at the same time must be waiting for the other resources, which are held by other processes in a non-sharable mode. Like in the above example: P1 is holding R1 in a non-shareable mode and at the same time trying to acquire R2, which is currently held by P2 in a non-sharable mode. Similarly, P2 is holding R2 in a non-shareable mode and at the same time trying to acquire R1, which is currently held by P1 in a non-sharable mode. This situation is called hold and wait for conditions.

3. No pre-emption

It implies that the resources cannot be pre-empted forcefully from a process. After the completion of the process, a resource is released voluntarily by the process.

4. Circular wait

Circular wait implies that the processes are waiting for resources that are circularly held by another process. In other words, Pi is waiting for a resource held by Pi+1 and so on Pn-1 is waiting for a resource which is held by Pn, and at last, Pn is waiting for a resource held by Pi. For example, Let a set of processes, say (P1, P2, P3, ---, Pn). They must wait in a circular way for the resources held by each other i.e.

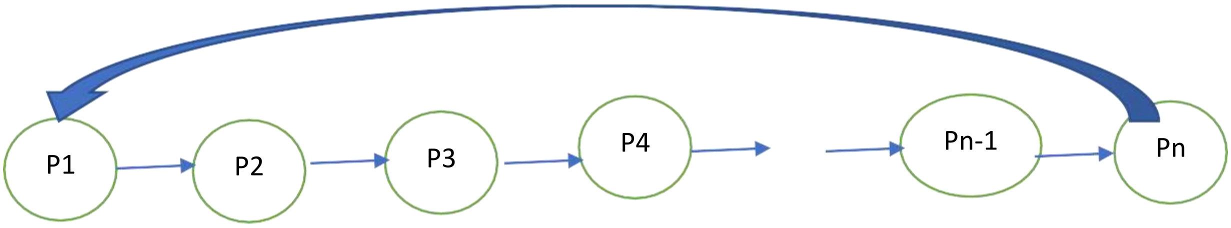
P1 is waiting for a resource which is held by P2

P2 is waiting for a resource which is held by P3

- - - - - -

Pn-1 is waiting for a resource which is held by Pn

Pn is waiting for a resource which is held by P1



Deadlock Detection

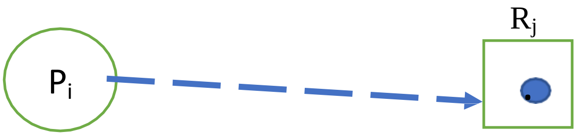
When all the four essential condition holds at the same time then deadlock occurs. Another way to detect deadlock is Resource Allocation Graph (RAG)

Resource Allocation Graph

The Resource Allocation Graph is a directed graph which is a set of vertices which is denoted by V and a set of edges which is denoted by E. Vertices (V) is divided into two types. Processes (P) and Resources (R). P = {P_1, P_2, - - -, $P_{n-1,}$ P_n} the set consisting of all the processes in the system. A process is represented by a circle, with the process name indicated with a label inside the circle. R = {R_1, R_2, - - - -, R_{n-1}, R_n} the set containing of all resources types in the system. A resource is represented by a square, with dots inside the square representing different instances of that resource.

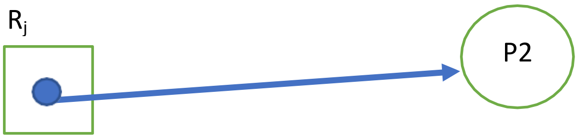
There are two types of edges: Request Edge & Assignment edge.

An edge from process P_1 to Resource R_j represents a request edge.



Request Edge

An edge from resource Rj to process Pi represents an assignment edge.



Assignment Edge

Note: The assignment edges originate from the instance within the resource symbol, which indicates which instance is assigned to the process.

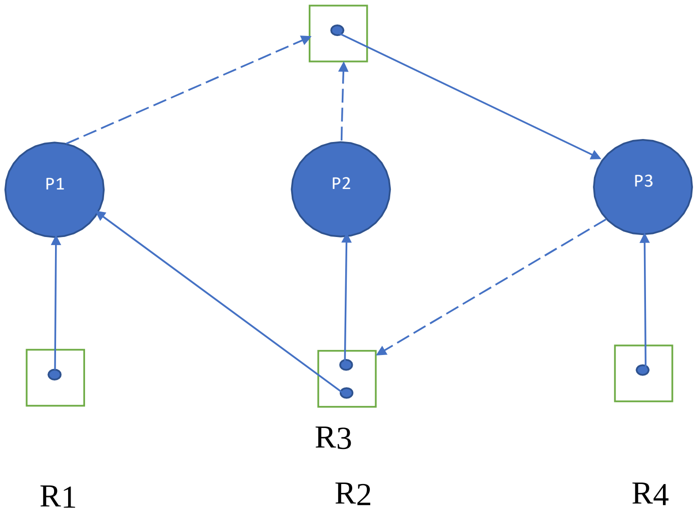
If there is no cycle in RAG, it indicates that there is no Deadlock occurred.

If there is a cycle detected in the graph and the resources involved in the cycle have only one instance per resource, then it indicates that a deadlock exists.

If there is a cycle detected in the graph and some of the resources involved in the cycle have multiple instances per resource, there may be a deadlock exists.

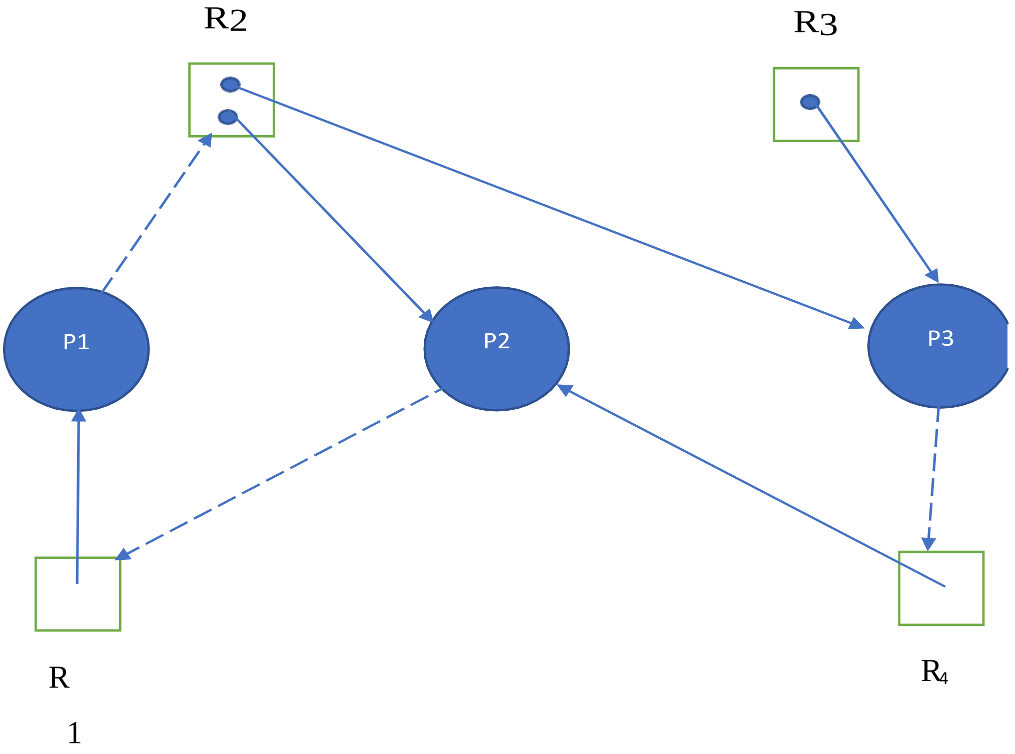
For example

Let there are 3 processes P1, P2, P3, and 4 resources R1, R2, R3, R4. R1 has one instance, R2 has 2 instances and R3 has one instance, R4 has one instance. R1, R2 is allocated to P1 and requested for R3. R2 is allocated to P2 and requested for R3. R3, R4 are allocated to P3 and requested for R2.



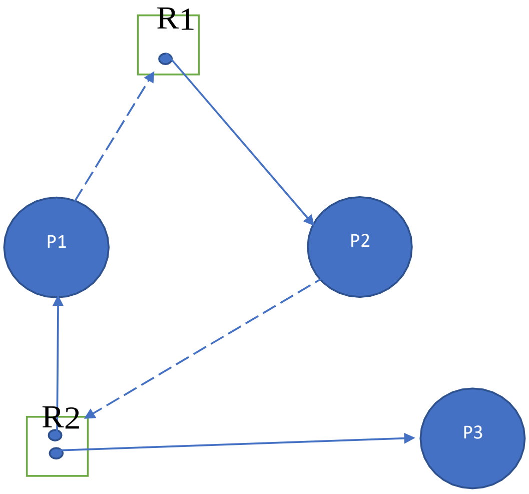
Resource Allocation Graph

Example of RAG with a cycle Deadlock



Resource Allocation Graph with Deadlock

Example of RAG with a cycle but no deadlock



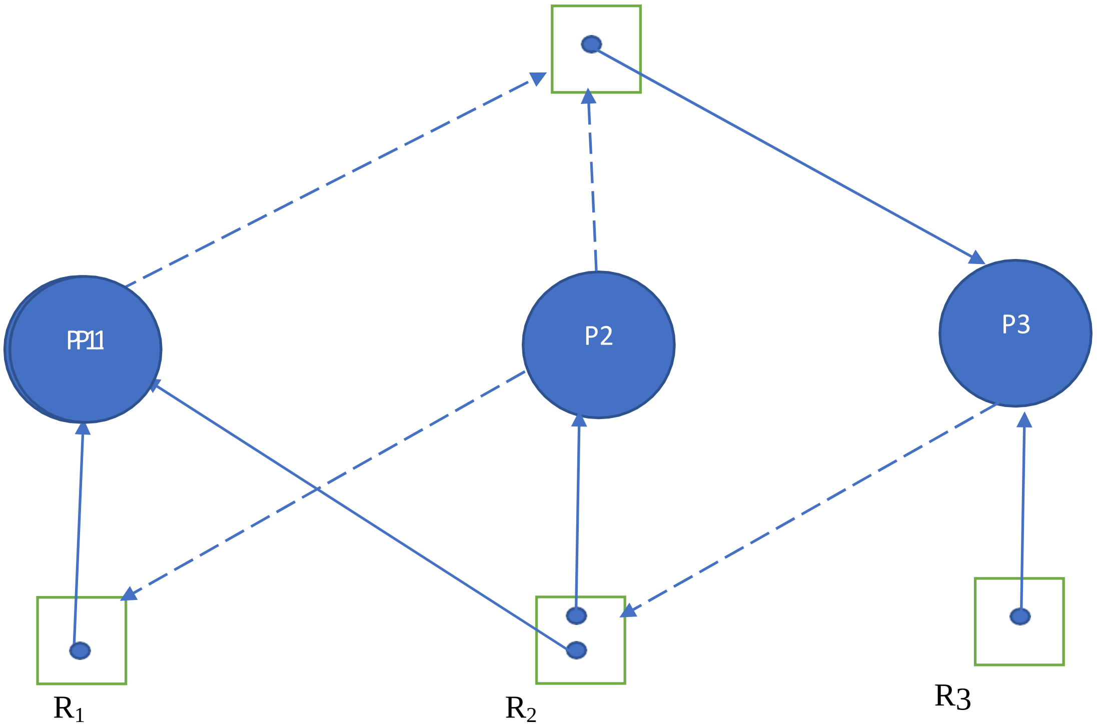
Resource Allocation Graph with Cycle but no Deadlock

Process Wait for Graph (PWFG)

PWFG can be obtained by collapsing the resources symbol in RAG. An edge from process P1 to P2 indicates that P1 waiting for a resource that is currently held by P2.

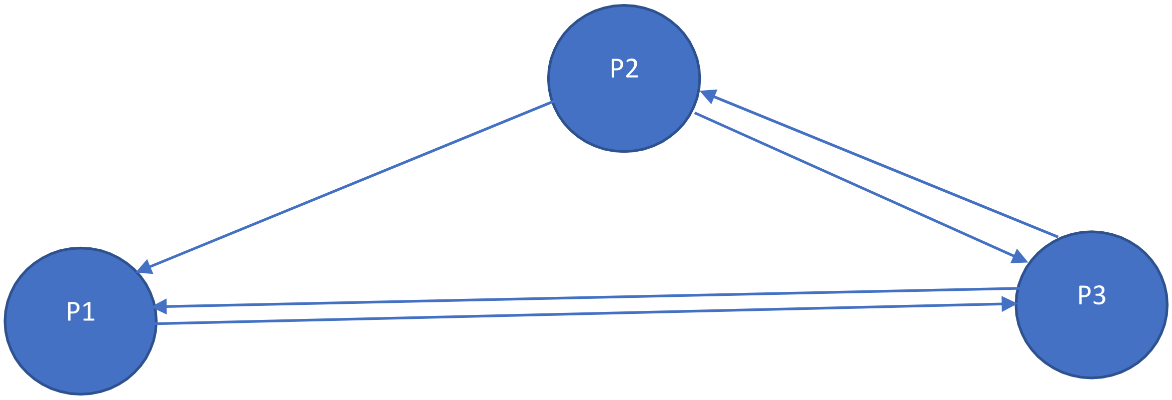
R_3

Consider the following RAG



Example of Resource Allocation Graph

Equivalent PWFG for above RAG



Process Wait for Graph (PWFG)

The PWFG contains a cycle, thus indicated that deadlock exists.

Deadlocks Management

Different OS adopts different strategies to handle deadlocks. The policies to handle deadlockscan be classified into four major groups:

- Deadlock Prevention
- Deadlock Avoidance
- Deadlock Detection and Recovery
- Deadlock Ignorance

The deadlock prevention techniques prevent the occurrence of at least one of the four conditions that cause deadlocks. By preventing are of the necessary four conditions deadlock can be prevented. The deadlock avoidance techniques acquire information in advance about which resource a process will claim at what stage of execution. The OS allocates resources in such a manner that no deadlock occurs. Deadlocks can be avoided by maintaining the system always in a safe state. The system is said to be in a safe state if all pending processes can be successfully executed in some sequence. That sequence is called a safe sequence and processes are in a safe state. The third category of deadlock management strategies is known as detection and recovery. These types of techniques allow deadlocks to occur detect deadlock when it occurs and then apply certain methods to recover from deadlock.

Finally, an OS can assume that deadlock will never happen or rarely occur and fully ignore it. This approach might be referred to as the no-policy approach. The positive point of this approach is it saves CPU time and memory space for deadlock management, unlike thoserequired for detection, prevention, or avoidance methods. This strategy has been adopted in UNIX.

Deadlock Prevention

This technique allocates resources in such a manner that at least one of the four necessary conditions of deadlock cannot occur. It denies at least one of the four conditions required for the occurrence of a deadlock. These techniques do not utilize a resource properly. Forexample, an OS adopting a deadlock prevention technique allows recourses to be preempted from a blocked process. No preemption condition is violated and deadlock does not occur.

We have already discussed that there are four conditions to be satisfied for the occurrence of a deadlock. Below is a brief discussion about how the conditions for a deadlockcan be prevented.

1. Mutual exclusion condition

The mutual exclusion condition indicates the locking of resources in exclusive, also known as the non-shareable mode that leads to blocking of resources. An OS can avoid blocking resources by locking the resources in shareable mode, if feasible. For example, two processes need to read data from a file.

The OS can allow locking of that file in shareable mode by two processes simultaneously. Locking resources in shareable mode prevents waiting for a resource. Whena process request to look at a resource in shareable mode, the OS allows the locking instantly and no time is wasted waiting to lock the resource exclusively.

Some resources cannot be shared by multiple processes. For example, two processes cannot simultaneously share a scanner. Again when a process writes data to a file, it needs to lock the file in exclusive mode.

2. Hold and Wait

An OS can avoid this by adopting a strategy that a process must request for all resources it requires at the same time. This strategy can be implemented in two ways.

Firstly, a process p requires all the resources it needs when the execution begins. On the other hand, a process can start execution with a minimum set of resources required and request the other resources when required at the time of execution. In the second approach, a process must release all the resources it presently holds before requesting any other required resources.

For example, a process p requires resources R1, R2, R3, and R4 in exclusive mode in the sequence: R3 and R1 simultaneously at the beginning of execution, then R1, R2, and R3 simultaneously, and lastly R1 and R4 simultaneously. When applying the first strategy, process P acquires all four simultaneously before it starts execution.

When applying the second strategy, the process p needs not to acquire all the four resources at the beginning. The process p1 acquires the resources R1 and R3 at the time of starting execution. Soon the process p requires R2. The process p releases all the acquired resources R1 and R3 and sends a request to the OS for acquiring the resources R1, R2, and R3. Later, the resources R2 and R3 are not required and R4 is needed. The process p releases all the acquired resources R1, R2, and R3 and sends a request to acquire R1 and R4.

3. No Preemption

An OS can adopt a policy to avoid this condition. If a process requests an unavailable resource, the process must release all the resources it has presently acquired and wait for the required resources. In other words, when a process, P requests a busy resource this policy allows other processes to preempt the resources currently held by P.

Certain resources are good candidates to be preempted. For example, CPU, and memory need a mention. While revoking the control of the resources from a process the OS updates the process control block of the process. Not all the resources can be preempted. For example, a partly updated file cannot be preempted since data will be lost.

4. Circular Wait

An OS can impose an ordering of resources for avoiding the circular wait. Resources in a system are grouped into certain categories. For example, all magnetic disks form a group and all scanners from another group. Each category is assigned a number. When a process acquires a resource belonging to a specific category, no other process belonging to a categoryhaving a lower number can be claimed.

Formally, if there are n categories of resources in a system, ranging from 0 to n-1, and if a process acquires a resource belonging to category c, then the process can only request a resource belonging to category c+1.

This helps to prevent a formation circle because a process holding the control of a resource of the category n-1 cannot claim a resource of the type 0. The disadvantage of this technique is each process needs to acquire all the required resources in a predetermined and specific order depending upon the arrangement of numbers assigned to various categories of resources

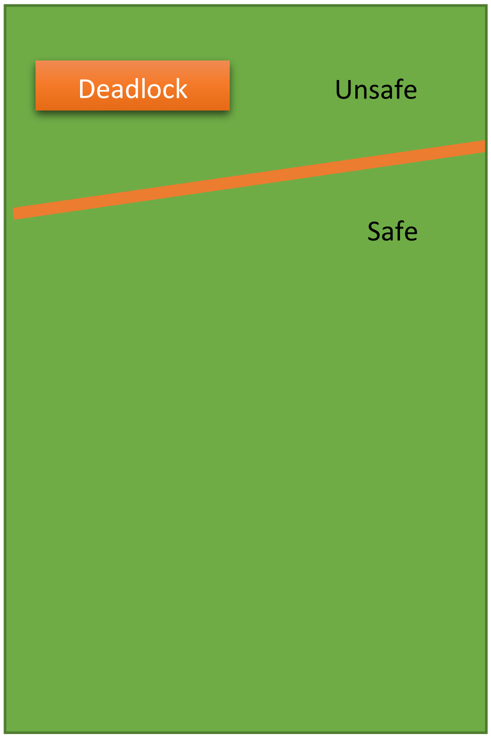
Deadlock Avoidance

In deadlock avoidance, Operating System requires additional information about which process requires which resource in advance to avoid a deadlock. The fundamental concept of deadlock avoidance is that OS only entertains those requests for resources by processes that will not lead to a deadlock. The deadlock-avoidance algorithm regularly inspects the resource-allocation state of the processes to confirm that there can never occur a circular-waitcondition. Resource-allocation *state* is explained by the total number of available resources and number of resources allocated to the processes, and the maximum demands of resources by the processes.

Safe State

When processes request resources and it is available in the system and allocated immediately then the system in a **safe state**. The system is in a safe state if there exists an order of processes $<P_1, P_2, ..., P_n>$ which satisfied their resources in that sequence. For eachP_i, the resources that P_i can still request can be satisfied by currently available resources + resources held by all the P_j, with $j < i$. That is: If P_i resources wanted are not instantly available, then P_i can wait till all P_j have finished. When P_j is finished, P_i can get needed resources, execute, return allocated resources, and terminate. When P_i terminates, $P_{i +1}$ can obtain its needed resources, and so on.

If a system is in a safe state then there is no Deadlock. If a system is in an unsafe state then there is the possibility of deadlock. To avoid deadlock, it is must confirm that a system never entered an unsafe state and it should be in a safe state.



Relationship between safe, unsafe, and deadlock state

Avoidance algorithms

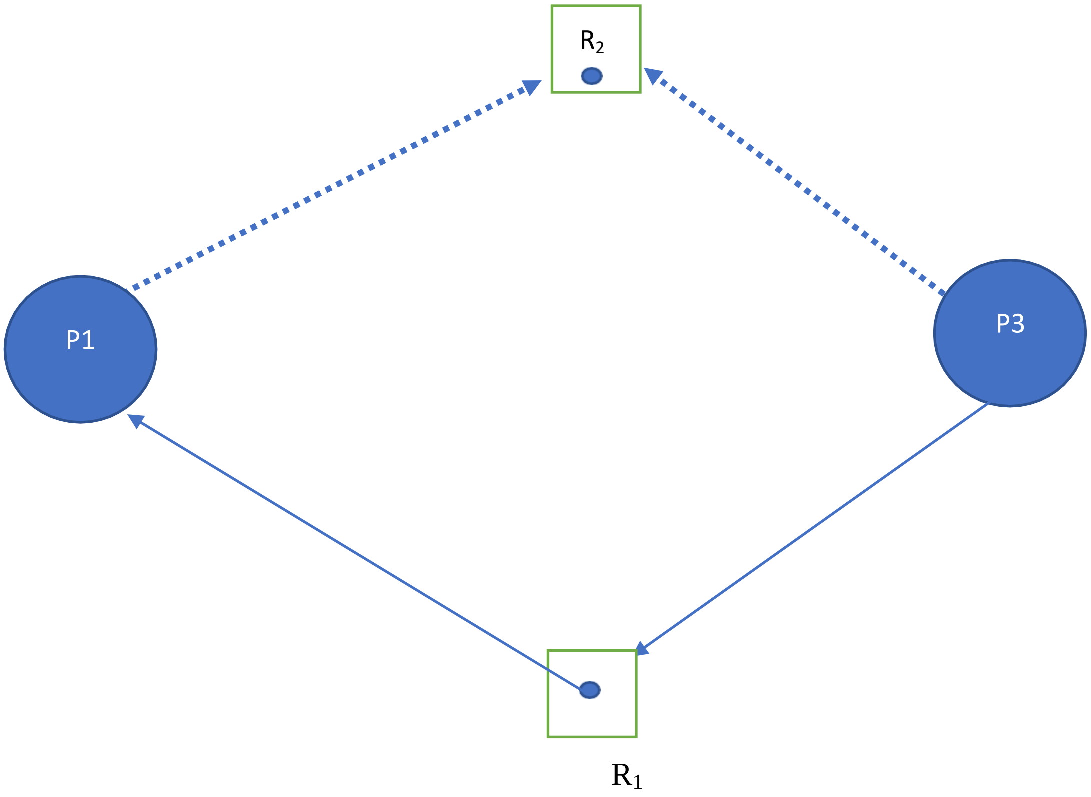
There are two algorithms to avoid deadlock

For the Single instance of a resource type, in this algorithm, we use a Resource- Allocation Graph (RAG) and the second algorithm which is deals with multiple instances of a resource type, then we use the Banker‟s algorithm.

Resource-Allocation Graph (RAG) Scheme

In Resource allocation Graph Scheme Claim edge $P_i \rightarrow R_j$ showed that process P_i may demand resource R_j; it is denoted by a dashed line. Claim edge changes to request edge when a process needs a resource. Request edge changed to an assignment edge when the resource is assigned to the process. When a resource is free by a process after execution, the assignment edge reconverts into a claim edge. Resources must be demanded in advance in the system.

Suppose that process P_i requests a resource $R_{j.}$ The request can be approved if and only if changing the request edge to an assignment edge does not result in the creation of a cycle in the Resource Allocation Graph(RAG).



(i) Unsafe state in RAG

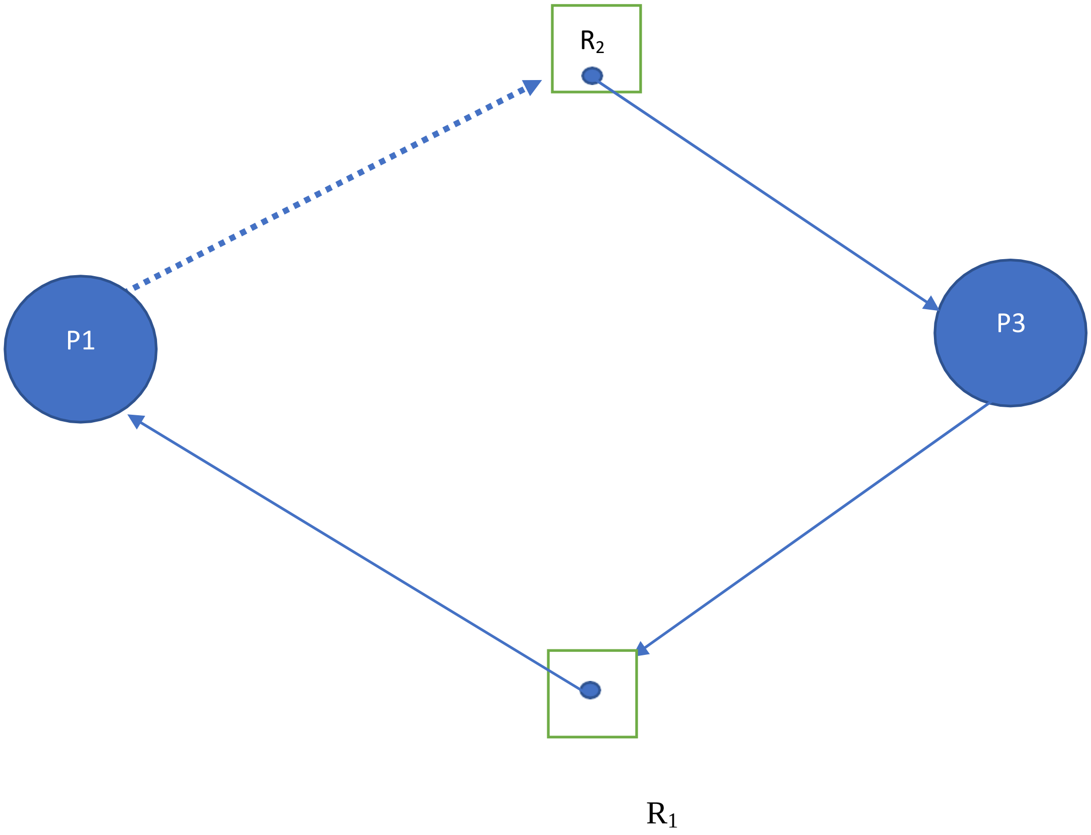


Figure showing (i) safe and (ii) unsafe state in RAG

BANKER’S ALGORITHM

The **banker’s algorithm** adopts the deadlock avoidance strategy. The name, banker‟s algorithm, depicts the similarity of the concept in the field of banking. A banker never allots the cash available in such a manner that the banker cannot fulfill the needs of its clients.

A scheduling algorithm that can avoid deadlocks is BANKER‟S ALGORITHM. It is modeled on the way that a single banker might deal with a group of customers to whom he has granted lines of credit. The banker‟s algorithm can be implemented for a single resource or multiple resources. The banker algorithm keeps track of all the resources currently available to the operating system and all the resources which are allocated to the processes. It maintains the various data structures to store the information about the number (instances) of resources. Whenever a process requests a resource, the banker‟s algorithm checks the availability of the resources and acts accordingly. Finally, it helps to check that the system is in a SAFE STATE or not.

Data Structures for the Banker’s Algorithm

Suppose *n* be the number of processes, and *m* be the number of resource types.

Available: Array of length *m*. If available [*j*] = *k*, it means that *k* instances are available of resource type R_j.

Max: *n* x *m* matrix. where n is the number of rows and m is the number of columns.*Max* [*i,j*] = *k*, it means that k instances maximum needed by process P_i of the resource typeR_j.

Allocation: *n* x *m* matrix. where n is the number of rows and m is the number of columns Allocation[*i,j*] = *k,* it means that k instances are currently allocated to process P_i of the resources R_j.

Need: *n* x *m* matrix. where n is the number of rows and m is the number of columns *Need*[*i,j*] = *k*, it means that P_i may need *k* more instances of R_j to complete its task. It can be calculated by matrix subtraction:

Need [*i,j*] = *Max*[*i,j*] – *Allocation* [*i,j*].

Safe State Algorithm

This algorithm is applied by an Operating System to determine whether a system is in a safe state or not. A system is safe if the system can allocate resources to every process in some order and still is in the safe state which means that its avoiding deadlock. In other words, a system is in a safe state only if there occurs, a safe sequence.

Let „m‟ be the total number of resources and „n‟ be the total number of processes in thesystem.

Let „Work‟ and „Finish‟ be the two arrays of length „m‟ and „n‟.

(1) Set Work = Available and also Set value FALSE for all the members of array Finish

FINISH[i] = FALSE, for i = 0 to n - 1

In other words, it is assumed that all the processes are to be executed and neither of the processes is completed yet.

(2) Find a process “i” such that following both conditions should be satisfied

Finish[i] = FALSE and Need[i] < = WorkIf no such “i” exists ,then go to step (4).

In other words, find a process that is not completed yet and whose need is less than available.

(3) Set Work = Work + Allocation$_i$ (Allocation of process i)Set

Finish[i] = TRUE.

Go to step (2).

In other words, after the completion of the execution of a process, the process will returnall the resources to the system. So, the available resources of the system will increase.

(4) Check if Finish[i] = TRUE for all “i” at that time

System is in safe state

else

System is not in safe state

Consider an example

Let there are 3 resources and 5 processes such that

Processes	Allocation			Max			Available		
	R1	R2	R3	R1	R2	R3	R1	R2	R3
P0	0	1	0	6	5	3	2	3	2
P1	2	0	0	3	3	2			
P2	3	0	2	9	0	2			
P3	2	1	1	2	2	2			
P4	0	0	2	5	3	2			

Find the Need matrix by using the formula:Need

= Max – Allocation

So, the need matrix is:

Processes	Need		
	R1	R2	R3

P0	6	4	3
P1	1	3	2
P2	6	0	0
P3	0	1	1
P4	5	3	0

Hereby considering the need of all the processes, we have to find such a process whose need is less than available. It is visibly that the need of the process P1 is less than Available, so the process P1 will be executed first. (As 1 3 2 < 2 3 2)

After the execution of process P1, the P1will return all the resources to the system. So, the available resources will increase as the following :

Available = Available + Allocation$_i$

In the same way, after the completion of a process, the Available will be updated, as the process returns the resources to the system after being executed.

So, the processes will be executed in the following sequence

< P1, P3, P4, P0, P2>

Explanation of Safety Algorithm

First, we set all processes in the false state which means that no processes are executed or completed yet, and set work is equal to available. Then we find repeatedly a process that is false state and whose need is less than equal to work. If we find such a process then we change its state to true and its resources are deallocated which is added in work. In this way, the number of available resources is increased. Then we check all the processes are true or not. If all the processes are true then the system is in a safe state. If even one process is in a false state then the system is in an unsafe state.

Resource Request Algorithm

The second part of the banker‟s algorithm, known as the resource-request algorithm determines whether granting a resource according to any claim by a process leads to anunsafe state. This algorithm is applied when a process sends a request to grant one or a set of resources.

Request$_i$ = request vector for processP$_i$ =

If request$_i$ [j] = K,

it means that P$_i$ requested the K instances of resource type R$_j$.

(1) If Request[i] <= Need [i] (Check request is genuine or not)then
Goto step (2)

Else

“Raise an error condition”

(Because the process has exceeded its maximum claim)

(2) If Request[i] <= Available (Check weather system has available resources or not)

Then

Goto step (3)Else
The process P$_i$ has to wait because the resource is not available yet.

(3) Let the system pretend that the resources have been allocated by changing the following:
Available = Availablei – Requesti ;
Allocationi = Allocationi + Requesti ;Needi =
Needi – Requesti;

Then check with the safety algorithm, if the result of the safety algorithm is safe, then thetransaction is completed and the process is allocated the requested resources.

However, if the system is in an unsafe state, then the process must wait and the old allocation state is restored.

Explanation of Resource Request Algorithm

In this algorithm, first, we check whether the request of the process is less than its need or not. If the request is more than need then it is an error because the process claims more resources from its maximum claim. Else we check current request is less than available ornot. If it is less than or equal to available then we allocate the request and update data structure such as allocation of the process by adding a request in allocation, the request is subtracted from need as well as available. Then we check the system is in a safe state or not by applying a safe state algorithm. If available is less than need then the request can not be granted and the process will wait because the resource is not available yet.

Advantages

Advantages of banker‟s algorithm, it allows mutual exclusion, hold and wait, no preemption conditions. The system assures that a process will be allocated resources without deadlock.

Disadvantages

The disadvantage of a banker‟s algorithm is the overhead of simulation and calculation before each time a process requests for a resource. Deadlock avoidance techniques cannot be applied until an OS knows the resource requirement of a process in advance. This approach may lead a process to starvation while for a long time to get the requested resources allocated by the OS.

Illustration

Consider the following table with 5 processes P_0 to P_4 and 3 resource types namely X Yand Z; X has 9 instances, Y has 5 instances, Z has 8 instances.

	ALLOCATION	MAX
	X Y Z	**X Y Z**
P_0	2, 2, 3	6, 2, 3
P_1	2, 0, 1	6, 1, 2
P_2	0, 1, 1	3, 1, 2
P_3	1, 0, 0	1, 0, 1
P_4	1, 1, 2	6, 2, 3

Using Banker‟s algorithm check whether the system is in a safe state or not. If it is a safestate then if P_1 requests (1, 0, 1) then it is a safe state or not. If it is not in a safe state then, what changes are required in available resources to make the system safe?

Total instance of resources is

X = 9, Y = 5, Z = 8

Total allocated resources to P_0 to P_4 areX = 6,

Y = 4, Z = 7

Now available is

X = 3, Y = 1, Z = 1

Now calculate Need: i.e. $Need_i = MAX_i – Allocation_i$

Process	Allocation	MAX	NEED	Available
	X Y Z	**X Y Z**	**X Y Z**	**X Y Z**
P_0	2, 2, 3	6, 2, 3	4,0,0	3,1,1
P_1	2, 0, 1	6, 1, 2	4,1,1	
P_2	0, 1, 1	3, 1, 2	3,0,1	
P_3	1, 0, 0	1, 0, 1	0,0,1	
P_4	1, 1, 2	6, 2, 3	5,1,1	

Now apply safe state algorithm to check whether the system is safe state or not.First step is set all process equal to false i.e. P_0, P_1, P_2, P_3, P_4 are FALSE. Now find such process which false and whose $need_i \leq$ Available i.e. P_2.

Now update data structure and status of processes i.e. $finish_i$

Available = Available + $Allocation_2$

$Finish_2$ = True

Now new Available = 3, 1, 1 + 0,1, 1

= 3, 2, 2

All processes are not true so again go to step - 2 to find a process that is yet false andneeds is less than equal to available.

$Need_3$ < = Available i.e. P_3

Now update data structure and status of processes i.e. $finish_i$Now

Available = Available + $Allocation_3$

New Available = 3, 2, 2 + 1, 0, 0

=> 4, 2, 2

$Finish_3$ = True

Now 3 processes are false i.e. P_0, P_1 & P_4 & Available = 4, 2, 2

All processes are not true so again go to step - 2 to find a process that is yet false andneeds is less than equal to available.

$Need_0$ < = Available i.e. P_0

Now update data structure and status of processes i.e. $finish_i$Now

Available = Available + $Allocation_0$

=> 4, 2, 2 + 2,2,3

= 6, 4, 5

$Finish_0$ = True

Now 2 process are false i.e. P_1 & P_4 and Available = 6, 4, 5

All processes are not true so again go to step - 2 to find a process that is yet false andneeds is less than equal to available.

$Need_1$ < = Available i.e. P_1

Now update data structure and status of processes i.e. $finish_i$New

Available = Available + $Allocation_1$

= 6,4,5 + 2, 0, 1

=> 8, 4, 6

$Finish_1$ = True

Now only one process i.e. P_4 is false if its need is greater than Available then system isnot safe state but $need_4$ < Available

Update Available & Finish

New Available = Available + $Allocation_4$

= 8, 4, 6 + 1, 1, 2

=> 9, 5, 8

Now finish is true so there is no false in finish and all resources are available now i.e. 7, 3, 6.

□ System is safe state & sequence is

$< P_2, P_3, P_0, P_1, P_4>$

The system is in a safe state if the process runs in the above sequence.

Now check if P_1 requests (1, 0, 1) then if it is allocated to the process then whether it isin safe-state or not? To check that the resource request algorithm follows.

In this algorithm first request is to check whether it is less than $Need_i$ or not. If it is greater than $Need_i$ its mean P_i request more resources as it claims earlier so it can‟t be given & request directly terminated. If $Request_i$ is less than $Need_i$ then we check $Request_i$ < = Available if it is true then the request is granted & we check that now system is in a safe state or not by using the safe state algorithm used earlier. If it is false then at that time $Request_i$ can not be fulfilled and the process P_i made to wait.

In this case Request P_1 is (1, 0, 1) which less than $Need_1$ and Available so we run resource request algorithm.

Set Available = Available - $Request_i$

= 3, 3, 1 - 1, 0, 1

Now Available = 2, 3, 0

Set Allocation_1 = Allocation1 + Request_1

= 2, 0, 1 + 1, 0, 1

Now Allocation_1 = 3, 0, 2

Set Need_1 = Need1 - Request_1

= 4, 1, 1 - 1, 0, 1

Now Need_1 = 3, 1, 0

Now Request is granted now we check after allocating the request whether the system is in a safe state or not. To check that we used a safe state algorithm. Now the situation of processes is given below:

Process	**Allocation**	**MAX**	**NEED**	**Available**
	X Y Z	**X Y Z**	**X Y Z**	**X Y Z**
P_0	2, 2, 3	6, 2, 3	4,0,0	2,3,0
P_1	3, 0, 2	6, 1, 2	3,1,0	
P_2	0, 1, 1	3, 1, 2	3,0,1	
P_3	1, 0, 0	1, 0, 1	0,0,1	
P_4	1, 1, 2	6, 2, 3	5,1,1	

Now by applying a safety algorithm there is no such process whose need_i < = available. So if the request of process_1 is granted then the system comes in an unsafe state. So, we revoke the request of Process_1 so that system may run safely. To overcome this problem we must need at least one A-type and at least two C-type Resource. If it will available in the available list then the system may come in a safe state.

Deadlock Detection and Recovery

Another approach to deadlock management is detection and recovery. In this approach, the system is not prevented from occurring a deadlock. The operating system periodically searches or analyzes that whether a deadlock has occurred. If the system is in deadlock, OS recovers from deadlock.

This approach of deadlock management has two phases, detection, and recovery of deadlocks. The algorithm to detect deadlocks in an OS can use a data structure similar to that of the data structures used in the context of deadlock avoidance.

A variation of the safety algorithm discussed in deadlock avoidance has been discussed below. In this algorithm, the data structures ALLOCATION, CLAIMS (Request), and AVAILABLE are the same as the data structure used in the context of deadlock avoidance. Another data structure used in this algorithm is FLAG.

The FLAG is a vector of the length equal to the number of processes in a system. FLAG is used to mark and unmark a process. For example, FLAG[X]=TRUE means the process Px is marked, and FLAG[Y]=FALSE means that the process Py is unmarked.

An OS verifies after a certain interval whether a deadlock has occurred. This algorithm finds a sequence of execution of processes that do not lead to an unsafe state. For example, there are 3 processes P1, P2, P3. Executing these processes in the sequence P2□P1□P3 does not lead the system to deadlock. The steps in this algorithm are:

1. Initialize FLAG[i] to FALSE, where 0 <= i<=s-1, and s denotes the number of processes.
2. Find a process Px such that FLAG[X]=FALSE and CLAIMS$_X$<=AVAILABLE. Goto step 6, if no such process is found.
3. Update the AVAILABLE vector because, after the termination of the process, Px, OS revokes the control of all the resources held by Px. Perform AVAILABLE = AVAILABLE + ALLOCATION$_X$.
4. Mark the process Px as FLAG[x]=TRUE.
5. Go to step 2.
6. If FLAG[i]=TRUE for all i, where 0<=i<=s-1, then all processes can be completed properly and there exists no deadlock, else the system is deadlocked with the processes which are unmarked that is FLAG[i] is set to FALSE.

The algorithm needs an order of O(m × n^2) processes to detect that the system is in a deadlock state or not.

The frequency of verifyin5r4ag the system whether a deadlock has occurred is an important issue. An OS can adopt any of the two policies. First, an OS can verify the system state whenever a process sends a request for resources. Alternatively, an OS can verify the system state after a certain interval of time.

An OS can adopt two strategies to recover from deadlock. Firstly, all the processes involved in the deadlock can be **terminated forcefully**. Alternatively, the OS can apply a **preemption technique** from a deadlock period.

Aborting deadlocked processes can be done in two ways.

An OS can abort all the deadlocked processes when a deadlock is detected.

On the other hand, an OS can apply the trial and error method. This means, one of the deadlocked processes is aborted and it is verified that whether the deadlock is over. This procedure is continued until the system reaches a safe state.

Which process is to be aborted first depends on the cost of terminating the process. If a process is 95% done, OS tries to avoid aborting that process. The number of processes to be aborted is also important in this context. For example, four processes are deadlocked. The OS can recover itself from deadlock by aborting either all of the processes A, B, and C or a singleprocess D. In this case, an OS will sacrifice a single process to recover from deadlock.

The second approach of recovery from deadlock is preempting a resource from a process. In this case, a victim is select, the process from which a resource is revoked by anOS is rolled back to a safe previous state and continued later on. In this approach, the starvation problem may occur by selecting the same process as a victim again and again.

Deadlock Ignorance

In deadlock ignorance, we assume that our system never enters in the deadlock condition. It is only a theoretical concept. Practically it is not possible because if the system is not in a safe state then the system may be in a deadlock state or we can say that if the system satisfied all the necessary four conditions simultaneously then the system must be deadlock state which can be ignored.

Comparison Between Various Deadlock Management

Deadlock Prevention	**Deadlock Avoidance**	**Deadlock Detection and Recovery**
It is easy method adversely affects Utilization	It is complex method Utilization improved at a cost	It is also a complex method It Depends on the frequency of detection
Involves violating at least one necessary conditions for deadlock	RAG Algorithm Bankers Algorithm (Multiple instances)	Maintain a wait-for graph of the and periodically invoke a “cycle-search” algorithm
Used in critical systems	Used in applications that have deterministic needs	Used in Database Applications

Points to Remember

- Deadlock: Process is said to be in a deadlock state if the process is waiting for a specific event that will not occur.
- The reusable resource can be safely used by only one process at a time.
- Four necessary conditions for deadlock are:
- (i) Mutual exclusion (ii) Hold and wait
- (iii) Circular waiting (iv) No preemption
- Resource Allocation Graph is used to detect the deadlock.
- Resource Allocation Graph is also known as RAG.
- If RAG contains a cycle then it may or may not deadlock the situation.
- In RAG, the process is represented by a circle and the resource is represented by a square.

- Process Wait For Graph is obtained from Resource Allocation Graph.
- Process Wait For Graph is known as PWFG.
- The policies to handle deadlock can be classified into four major groups
- Deadlock Prevention
  - Deadlock Avoidance
  - Deadlock Detection and Recovery
  - Deadlock Ignorance
- Deadlock prevention is a set of methods for ensuring that at least one of the necessary conditions cannot hold.
- Deadlock avoidance needs extra information about how resources are to be demanded.
- Safe order is an order of the process in which there exists one order in which all the processes can be executed without resulting in a deadlock.
- The deadlock avoidance algorithm enthusiastically examines the resource-allocation state to confirm that a circular wait condition can never exist.
- A safe state is not a deadlock state.
- Not all unsafe states are deadlocks.

- Deadlocks occur only when some process makes a request that cannot be granted immediately.
- Banker‟s algorithm is a deadlock avoidance algorithm.
- A safety algorithm is used to find the safe state of the system.

PRACTICCE EXERCISES

1. What is deadlock?

2. What are the necessary conditions for the occurrence of a deadlock?

3. How deadlock can be avoided.

4. What do you mean by RAG?

5. Explain methods for deadlock detection.

6. Explain safe state.

7. What are the different methods of handling deadlock.

8. Explain the Banker‟s algorithm with an example.

9. Explain the safe state algorithm with an example.

10. Explain the resource algorithm with an example.

11. Consider the following snapshot of a system.

Process	**Allocation**	**MAX**	**Available**
	M N O P	**M N O P**	**M N O P**
P_0	2, 2, 1, 3	7, 5, 3, 3	1, 1, 2, 1
P_1	2, 2, 1, 1	6, 2, 2, 4	
P_2	1, 2, 1, 0	4, 2, 4, 0	
P_3	1, 1, 0, 1	2, 2, 2, 3	
P_4	2, 1, 2, 0	2, 4, 3, 3	

Whether the system is a safe state or not. If process P3 request 0, 0, 1, 2 can this request granted immediately or not and also check after granting above request system is insate state or not. If it is not in a safe state then what changes are required in available resources to make the system safe.

B.Sc.(DATA SCIENCE)

SEMESTER-II

OPERATING SYSTEM

UNIT IV: MEMORY MANAGEMENT

STRUCTURE

OBJECTIVES

- Understanding Memory, Address Binding, Logical Vs Physical Memory
- Understanding Concept of Dynamic Loading and Linking, Swapping, Contiguous Memory Allocation, Segmentation, Paging, Demand Paging, Page Replacement algorithms

INTRODUCTION

The important task of allocating memory to processes, and efficiently, ensuring that processes have their instructions and data in main memory when needed, is termed as Memory Management.

Memory is one of the important resources of the computer system. The main memory is usually divided into two partitions, one for the resident operating system and the other for the user processes.

The operating system is responsible for memory management. It keeps track of the status of the memory, makes policy, allocates the memory, and then deallocates the memory from the process.

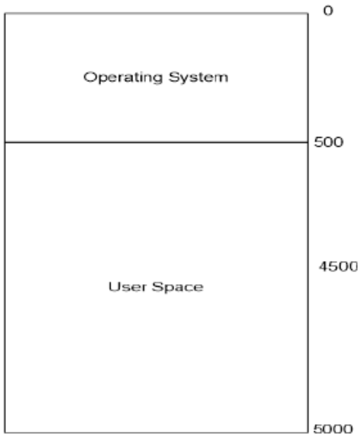


Figure: Main Memory

The functions of Memory Management

An operating system performs various activities for memory management:

1. It keeps track of the status of the memory, that whether it is free, allocated, available, or not available (full).
2. If the memory is free for the execution of the process, the Operating System

chooses a policy to allocate the memory to the process.

3. According to the chosen policy (algorithm), it allocates the memory to the process.
4. After the execution of the process, it deallocates the memory.

ADDRESS BINDING

Address binding means map from one address space to another i.e. the logical addresses to real physical addresses in memory. To execute a program, it must be loaded into the main memory at a particular location. The instructions that use addresses in a program must be bound to proper address space in the main memory. Many instructions use –fixed‖ addresses these must be *bound* to ―fixed‖ locations in the memory.

This binding of addresses of instruction and data to actual physical addresses can take place at compile-time, load time, and run time during the execution of a process.

To run a program there is a sequence of steps to execute a particular program which is as follow:

- A program is known as a source program and it is loaded into the main memory.
- A compiler or assembler or interpreter is required for compilation or assembly.
- After compilation or assembly, a code is generated by them which, is known as Object code.
- When a program is executed then, the program links with other object modules or system library which are loaded into linkage editors with the help of linker and loader.

At last, the process is stored in memory as a binary image, and system library files are dynamically loaded and dynamically linked.

DYNAMIC LOADING AND LINKING

In any program execution linking and loading plays a key role. Linking means creating an executable module of a program by linking the object codes which is created by the assembler or compiler. Other hand loaders, load these runnable modules to the main memory for execution.

Linking: It is the **procedure of linking all the modules or the sub-program or all the functions in a program for whole program execution**. It takes more than one object module and combines it into a single object file. The linker is also called a link editor. It takes object modules from the compiler or assembler and forms a runnable file for the loader.

Based on time Linking is classified into two categories – static linking and dynamic linking:

Linking of object modules is done at compile time and at load time. Compile-time linking is completed when the source code is converted into machine code is called Static linking. The load-time linking is done while the program is loaded into memory by the loader.

Loading

Loading is the **procedure of loading the program from auxiliary memory to the main memory** for execution.

Dynamic loading

It involves loading routines into memory only when required. This is done during execution. Dynamic loading reduces the memory requirements of large programs. This is especially the case if there is a large set of infrequently used routines.

Dynamic linking

It is often used for libraries. Only a ―stub‖ of the library is kept in the image of the program. When a program calls one of these routines, the routine is loaded and linked into memory. All programs share one copy of the same library routine.

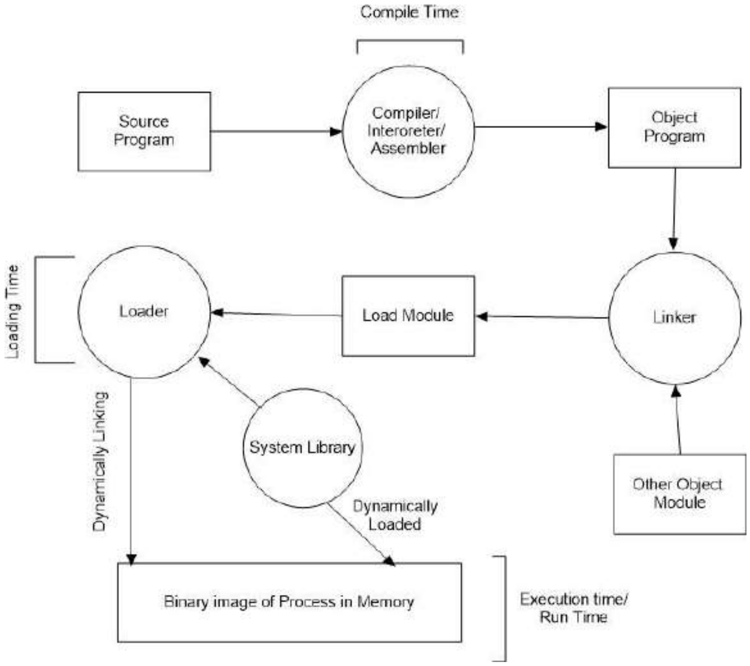


Figure Processing of a User Program

Address binding of instructions and data to memory addresses can happen at three different stages i.e. compile time, load time, execution time (as shown in figure Processing of User Program)

Compile-time: If the memory location is known a priori, **absolute code** can be generated. It must recompile code if starting location changes, after compiling the process.

Load time: It must generate **relocatable code** if memory location is not known at compile time. Final binding is delayed until load time.

Execution time: If the process can be moved during its execution from one memory segment to another, binding may be delayed. Then hardware support is needed for address maps. For example base and limit registers. Most general-purpose operating systems use the execution time-binding method.

LOGICAL VS. PHYSICAL ADDRESS SPACE

When a process is executing, the CPU would generate addresses, called **Logical Ad**

dresses. There are two registers used to define logical address space. These are the Base register and Limit register. Base register stores the starting memory address of the process and the Limit register stores the size of a process. Let a process P whose size is 5000 i.e. value of the Limit register and it has the start address is 1000. i.e. value of Base Register. A pair of **base** and **limit** registers define the logical address space as the shown figure below

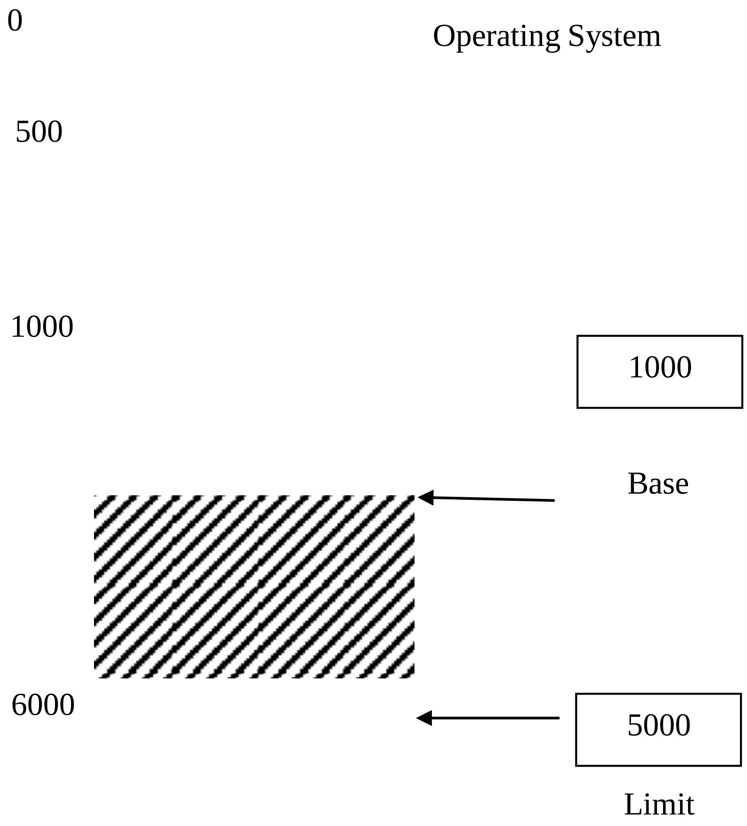


Figure Base and Limit Register

Executing processes that occupy corresponding physical addresses in the physical memory is known as **Physical Addresses.** Logical and physical addresses are the same in compile-time and load-time address-binding schemes but logical (virtual) and physical addresses differ in the execution-time address-binding scheme. A physical address is the effective memory address of instruction or data.

SWAPPING

Swapping is a mechanism in which a process can be swapped temporarily out of memory to a storage device (backing store) and another process brings in memory for execution. If a swapped-out process requires again then it is brought back into memory for continued execution.

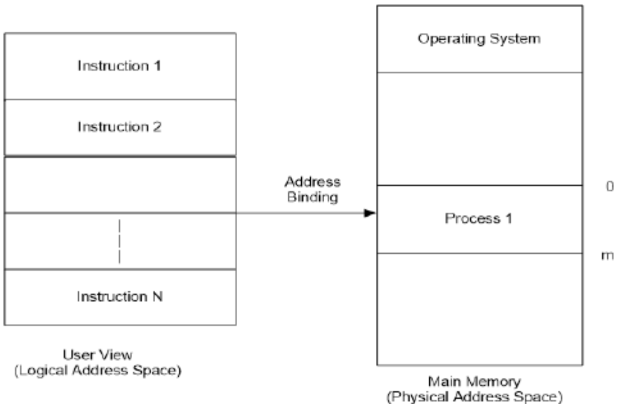


Figure Logical and Physical Address Space

The backing store is a fast disk large enough to accommodate copies of all memory images for all users. It must provide direct access to these memory images. When the process swapped out then it is called **Roll out** and when it brings in then it is called **Roll in.** The swapping variant is used for priority-based scheduling algorithms where the lower-priority process is swapped out due to the arrival of a higher-priority process. It can be loaded and executed. A major part of swap time is transfer time. The system maintains a ready queue of ready-to-run processes which have memory images on disk.

Example: Let there are n processes in the storage device and P4 in main memory which is not required. Now P2 is required so P4 is roll out and P2 is rolled in which is known as swapping as shown in the figure.

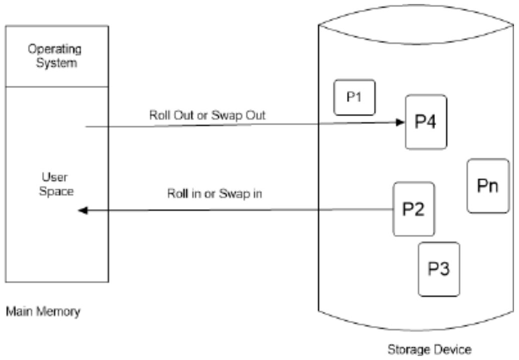


Figure: Swapping

MEMORY ALLOCATION

Memory management includes the various methods of allocating the memory to the different processes. The memory management techniques are basically of two types:

(i) Contiguous memory allocation.

(ii) Non-contiguous memory allocation.

Contiguous Memory Allocation

In contiguous memory allocation, the data and instructions of a program are sure to reside in a single contiguous memory area. It is a simple technique of allocating memory to the processes. There is no possibility of sharing data and instructions among the processes, which are loaded into entirely different portions of memory. The entire process is allocated a single contiguous memory area for its execution. Addressing is very simple as compare to the non-contiguous allocation methods.

Fragmentation

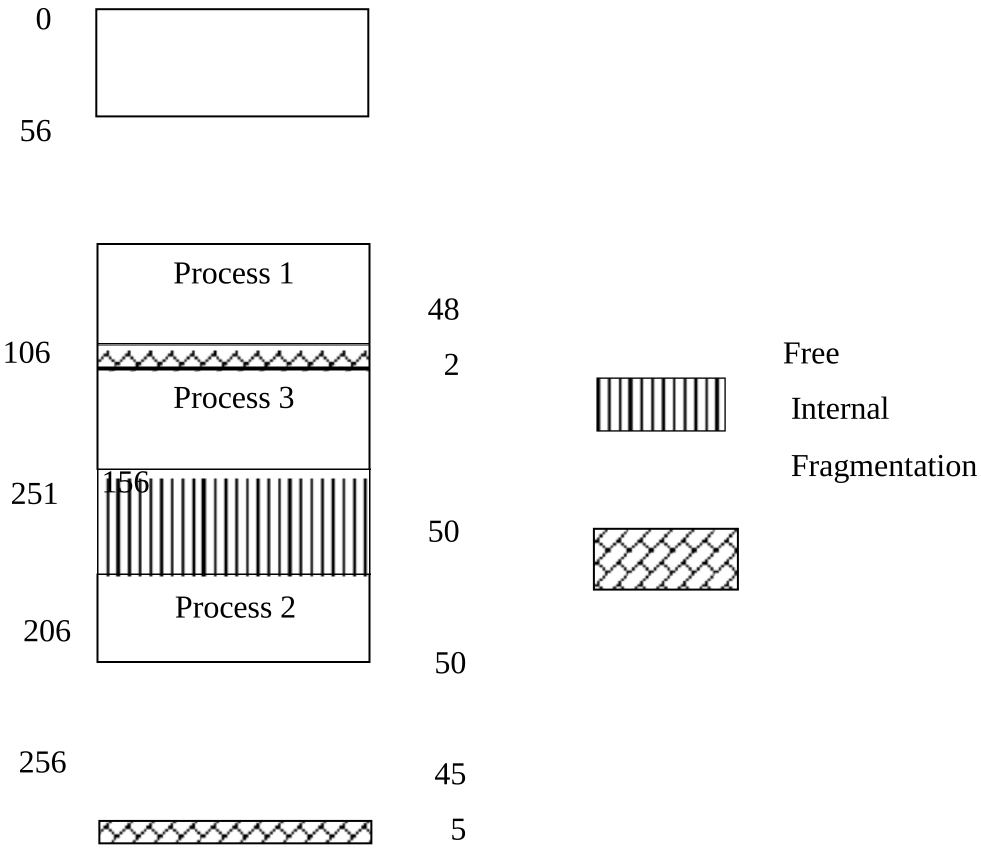
In contiguous memory allocation, one problem occurs i.e. Fragmentation. It can be classified into two categories:

1. Internal Fragmentation
2. External Fragmentation

1. **Internal Fragmentation:** In fixed-size memory management, allocated memory

to a process may be slightly larger than the requested memory by the process. The size difference between occupied memory and partition size, which is not used by another program is known as Internal Fragmentation.

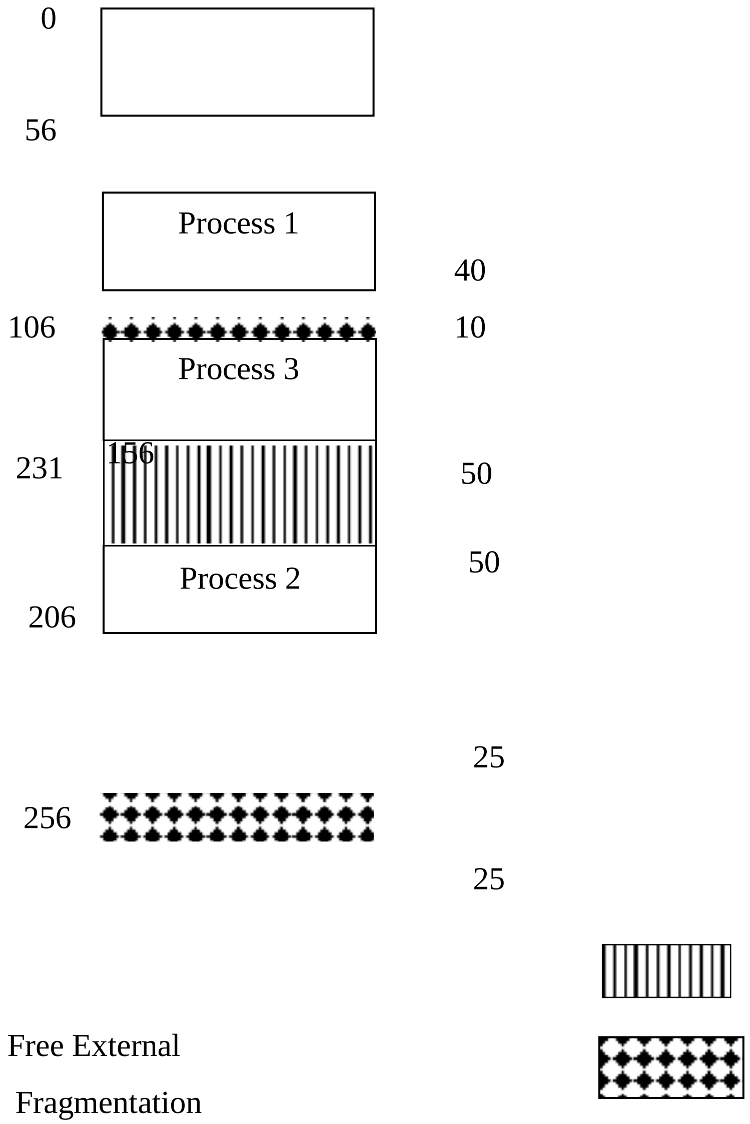
Example of Internal Fragmentation: Let, there are three processes P1, P2, P3 whose size is 48, 45, 50 respectively. The total memory size is 256 and OS occupies 56K and the rest of the memory is divided into fix partition of 50k.



Example of Internal Fragmentation

2. **External Fragmentation:** In variable size memory management, some small chunks are available in memory that cannot be effectively used. Total free memory is larger than the required memory by the requested process but it is not contiguous. It is known as External Fragmentation.

Example of External Fragmentation: Let, there are three processes P1, P2, P3 whose size is 40,25,50 respectively. The total memory size is 256 and OS occupies 56K and the rest of the memory is divided into fix partition of 50k.



Example of External Fragmentation

Non-contiguous Memory Allocation

In non-contiguous memory allocation, the data and instructions of a program may reside in non-contiguous memory locations. It is possible to allocate different memory locations to a process for its execution. In other words, the process can be divided into parts and can be allocated to different memory areas. It offers various advantages over, contiguous memory allocation like it permits sharing of code and data amongst processes. There is no external fragmentation of physical memory. It supports the vertical memory concept.

Contiguous Memory Allocation Methods

The memory is partitioned into different blocks of different sizes for accommodating the programs. The partitioning is of three types:

1. Single partitioning memory management
2. Multiple Fixed Partitioning
3. Multiple Variable Partitioning

Single Partition Memory Management

The main memory is divided into two partitions. One partition is for the resident operating system and, the other for the user processes. In the single partition memory management technique, the user area has a single partition. It is the simplest memory management technique.

Consider an example, where the total main memory is 256 MB. 56 MB is reserved for the operating system and 200 MB is for the user processes.

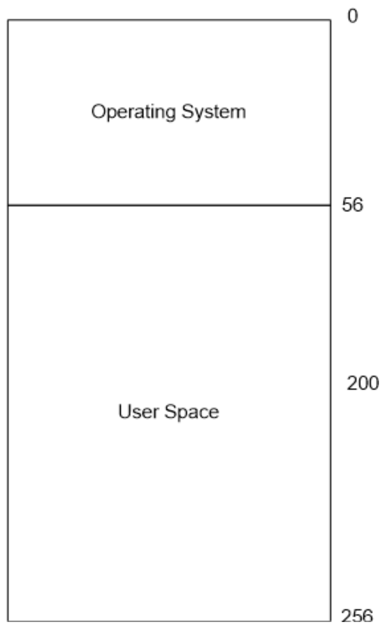


Figure: Single Partitioned Main Memory

Here, the user area has a single partition. At a time, only one program can be executed. The maximum length of the program, which the user can execute, is equal to the size of the partition of the user area. Here, the maximum size of a program can be 200 MB.

Internal fragmentation is much more. Since there is a single partition of memory, we can execute only one program at a time. Consider an example, where the size of a program is 50 MB. In this case, there will be an internal fragmentation of about 150.

Here, external fragmentation is also more. However, if the size of the program increases from the size of the partition, then it is not possible to execute the program.

Advantages

- It is a very simple memory management technique.
- Any program of size less than the size of the user area can be executed easily.

Disadvantages

- Multiprogramming is not possible.

- Internal fragmentation is more.
- It is not possible to execute a program of size greater than the size of the user area.
- The maximum size of a program to be executed is fixed.

Multiple Partitions with Equal Size

In this, whole memory is divided into a fixed number of partitions of different sizes, which may suit the program sizes. Each partition accommodates exactly one process. When a program needs, it is loaded into a partition that one big enough to accommodate the program. Sometimes, some space may be left unoccupied in a partition after loading a program, which is not used by another program that space is wasted and it is known as **Internal Fragmentation**. The degree of multiprogramming is fixed since the number of partitions is fixed is the drawback of this partitioning.

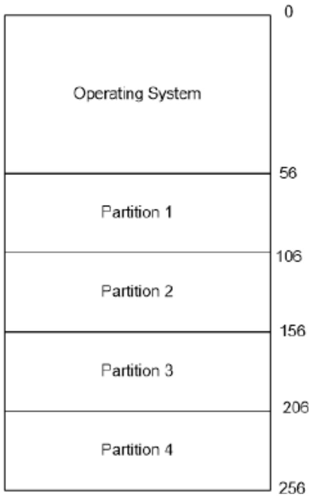


Figure: Multiple Fixed Partitioned Main Memory

In this technique, internal fragmentation is less as compared to the single partition technique. Consider an example, where a program of size 50 MB arrives. In this case, there will be no internal fragmentation. However, if a program of size 40 MB or 30MB arrives, then there will be an internal fragmentation of 10 MB or 20 MB respectively.

In this technique, external fragmentation may be more. Consider an example, where a process of size 55 MB arrives. Even if four of the partitions(200 MB) are free, but it is not possible to execute a program of 55 MB.

Advantages

- It is possible to execute more than one program at a time. In other words, multiprogramming is possible.
- Internal fragmentation is less as compared to a single partition.
- It is simple to maintain.
- The degree of multiprogramming depends upon the number of partitions.

Disadvantages

oIt limits the size of the program, the user can execute.

oExternal fragmentation is high.

oIt is not possible to execute a program having its size greater than the size of the partition.

Multiple Partitions with Variable Size

This scheme is free from the limitation encountered in the case of fixed partitioning. In this, the entire available memory is treated as a single partition. All programs, requesting memory are store in a waiting queue and loaded only when a free partition available which is big enough to occupy that program. When a program is allocated a space exactly equal to its size, the balance unoccupied space is treated as another free partition. When a free partition is too small to accommodate any program, it is called **External Fragmentation.**

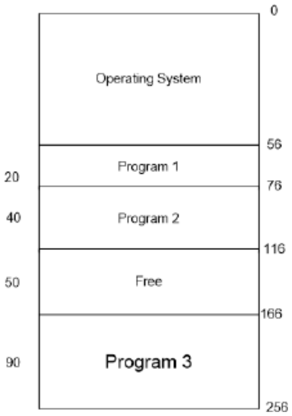


Figure: Multiple Variable Size Partitioned Main Memory

Three strategies can be used to allocate the memory to a process.

1. **First Fit:** According to this strategy, the O/S chooses the very first partition available, whose size is equal to or greater than the size of the process.

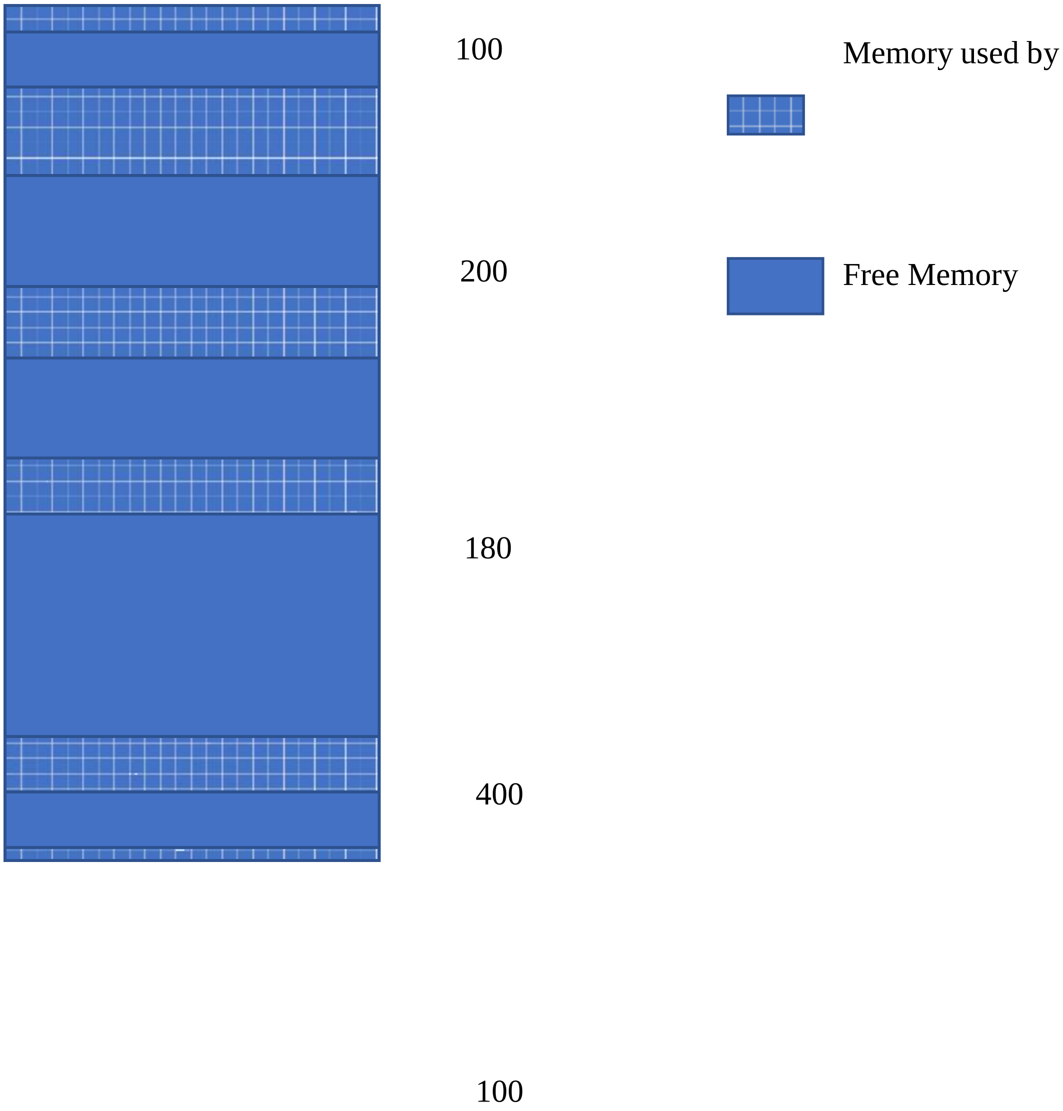
2. **Best Fit:** According to this strategy, the O/S chooses a partition that is smallest and whose size is greater than or equal to the size of the process. It may lead to the formation of small-sized unusable holes of memory.

3. **Worst Fit:** According to this strategy, the O/S chooses the largest partition and allocates it to the process. It may result in bigger-sized unusable holes of memory.

From the point of view of the size of the unusable holes, the first fit and best fit are better than the worst fit. From the point of view of time taken for searching the partition, the first fit is better than the other two i.e. Best fit and Worst fit

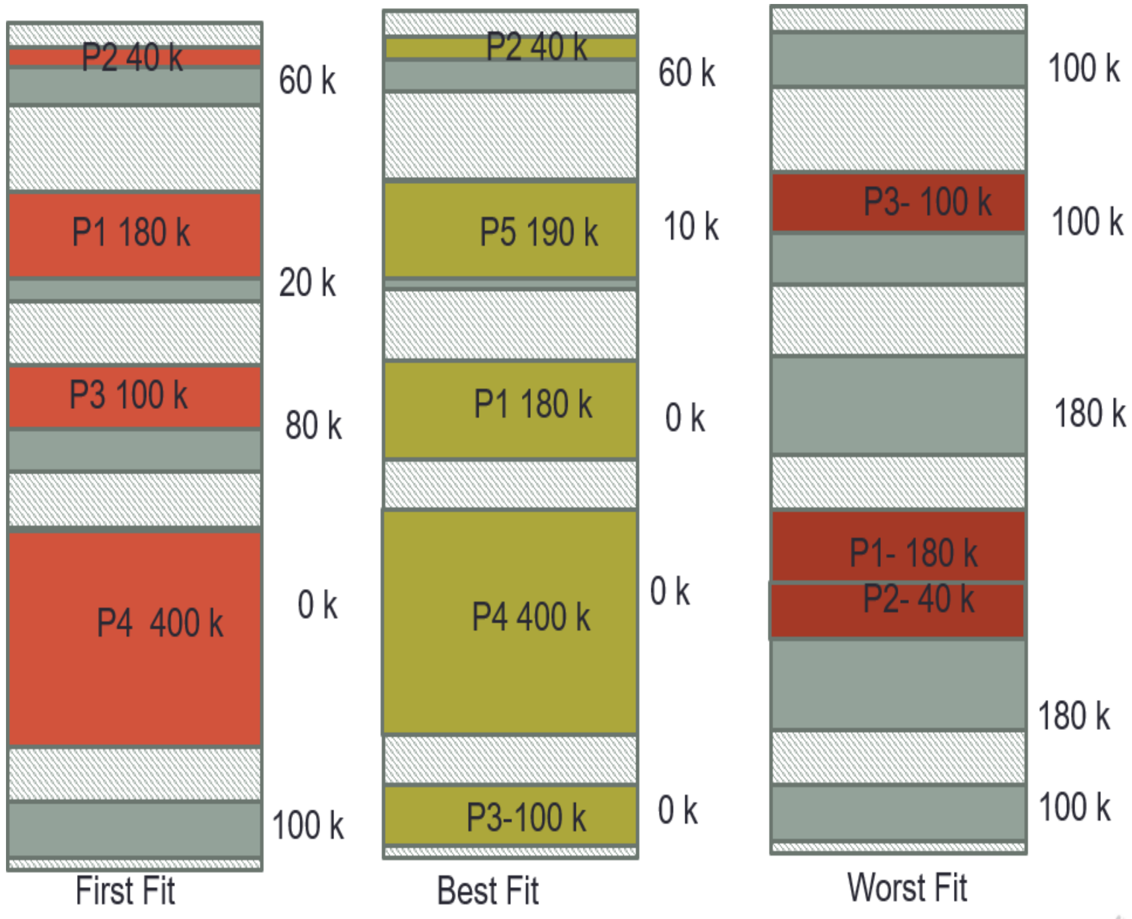
Example of Best Fit

Consider the requests from processes in given order 180K, 40K, 100K,400k, and 190K. Let there be four blocks of memory available of size 100K, 200k, 180K, 400k followed by a block size 100K.



Let there are three process P1, P2, P3, P4, and P5 of sizes 180, 40, 100, 400, and 190 respectively

By using an algorithm of First Fit, Best Fit & Worst Fit is shown by the figure below:



Total free 260 k but process 190k not loaded as memory isnot in contiguous

Total free 70 k all process allocated in memory

Total free 660 k but process P4- 400 k and P5- 190k not loaded as memory is not in contiguous

Compaction

Compaction is a technique to deal with the problem of external fragmentation. Compaction means to move the processes in such a way that scattered pieces of unused (free) memory blocks can be placed together so that they can be used by any other process.

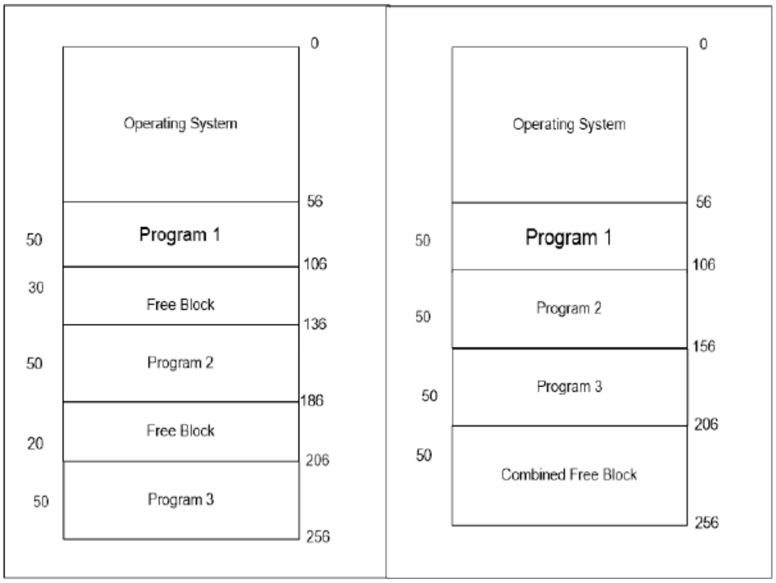


Figure: Before Compaction and After Compaction

Compaction involves the dynamic relocation of a program. This is done by using the relocation register.

However, the compaction algorithm is costly. The problem with the compaction algorithm is that it needs the dynamic relocation of addresses at the execution time of the process. So, the compaction is not possible if the relocation is done statically.

The cost of the compaction increases with the increase in the number of processes to be moved. Moreover, there is a need for a strategy to decide the direction (whether to move upwards or downwards) to move the processes.

Advantages

- Internal fragmentation is very less.
- The degree of multiprogramming can vary.

Disadvantages

- The O/S has to decide about the partition size, almost whenever a program arrives for execution.
- External fragmentation exists.

Memory Protection Contiguous Memory Allocation

When a logical address is generated by the CPU, then it is compared with the content of Limit Register (M). If the logical address is less than equal to M, then it is a valid address else it is an invalid address error, and the process is terminated. If the logical address is valid then the corresponding physical address is computed by adding the content of Relocation Register (B) i.e. base address to the logical address.

Physical Address = Logical Address L + Relocation value R

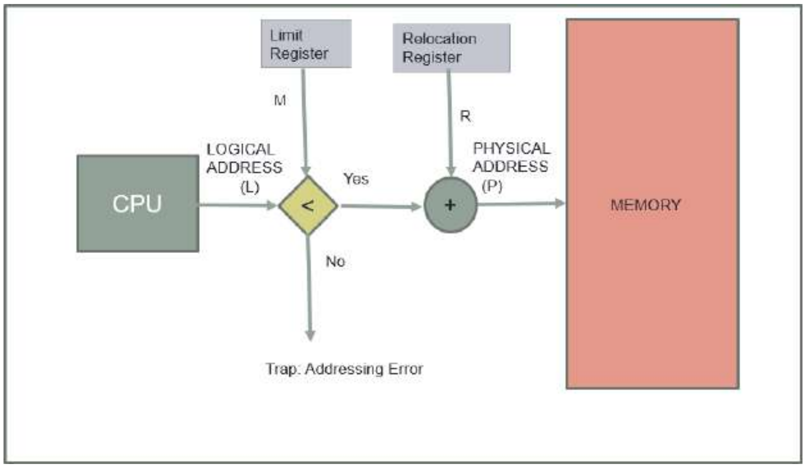


Figure: Memory Protection in Contiguous Memory Allocation

Non-Contiguous Memory Allocation Methods

Non-contiguous memory allocation involves a complex implementation and involves additional costs in terms of memory and processing.

The non-contiguous memory allocation method includes the various strategies like:

1. Paging
2. Segmentation
3. Paging with Segmentation

Paging

Many solutions have been suggested to handle the external fragmentation problem of the memory during the process execution. One scheme permits the logical address space of a process to be noncontiguous, thus allowing a process to be allocating physical memory wherever the latter is available. Memory paging implements this scheme. In the memory paging technique, the logical address space is divided into fixed-sized blocks known as **pages**. The physical address space is also divided into fixed-size blocks known as **frames**. A page is mapped into a frame. Individual pages and frames are recognized by a unique number known as page number and frame number respectively. The size of a page is equal to the size of a frame. A page number forms a part of the logical address and a frame number forms a part of a physical address.

Both page and frame size is usually a power of 2 and depends on the hardware. If a logical address consists of P number of digits, this means there are 2^p addresses in the logical address space. Similarly, if physical addresses consist of q number of digits, this means there are 2^q addresses in the physical address space. **It is always power by 2 because this is to divide logical address into a page number and page offset, easily.**

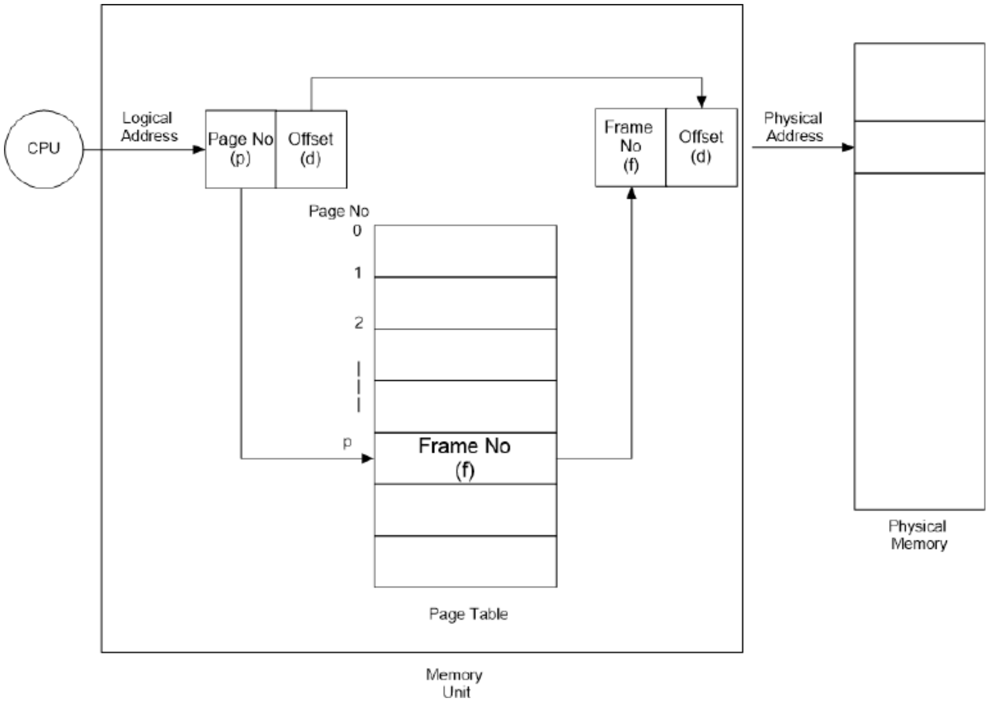


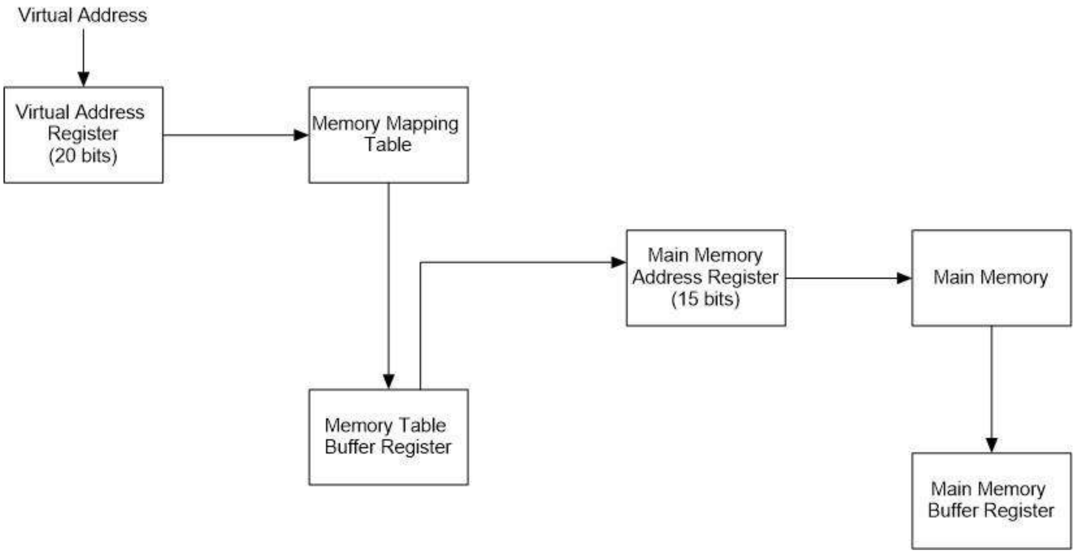
Figure: Paging Hardware

Mapping of Pages to Frame

A page table is maintained for mapping. It divides into two parts: Page Number & Page offset.

p	d
Page No	Offset

A logical address consists of two parts: Page number and Offset. If a logical address is P digits long and page size is 2^n digits long, then the higher or leftmost p-q digits in a logical address denote the page number, and the rest of the logical address denotes offset.


Virtual Address
Virtual Address Register (20 bits)
Memory Mapping Table
Main Memory Address Register (15 bits)
Main Memory
Memory Table Buffer Register
Main Memory Buffer Register

Memory Table for Mapping a Logical Address

For example, the logical address space can hold 2^{13} addresses. This means a logical address is 13 bits long. In other words, the logical address space is 8 KB. There are 8 pages in the logical address space. This means the size of each page is 2^{10} bits=1 KB. The size of the physical address space is 2^{12} bits= 4 KB. This means a physical address is 12 bit long. In the logical address of 13 bits the higher 13 bits (13-10) =3 bits denote the page number and the rest 10 bits denote the offsets. The logical address 011 1010101010 denotes page number 011 and the offset 1010101010.

A physical address consists of two parts: frame number and offset.

f	d
Frame No	Offset

The offset is the same as it is in the corresponding logical address. For example, in the previous example, the offset 1010101010 of the logical address 011 1010101010 is 10 bits long. The physical address corresponding to this logical address can be 01 1010101010, where 01 is the frame number and 1010101010 is the offset.

The operating system maintains a table to convert a logical address into its corresponding physical address. This table is called a **memory page table**. The method of converting a logical address into its corresponding physical address is known as **address translation**.

There are two fields in the page table. The first field stores the corresponding frame number of a page. For example, page 3 is stored in the 13th frame. In the first field of the page table value of the 4th cell will be 13. The second field stores a binary value either 0 or 1 and is known as the **presence bit**. If there exists a frame corresponding to a page, the value of the cell is marked as 1 otherwise the value is set to 0.

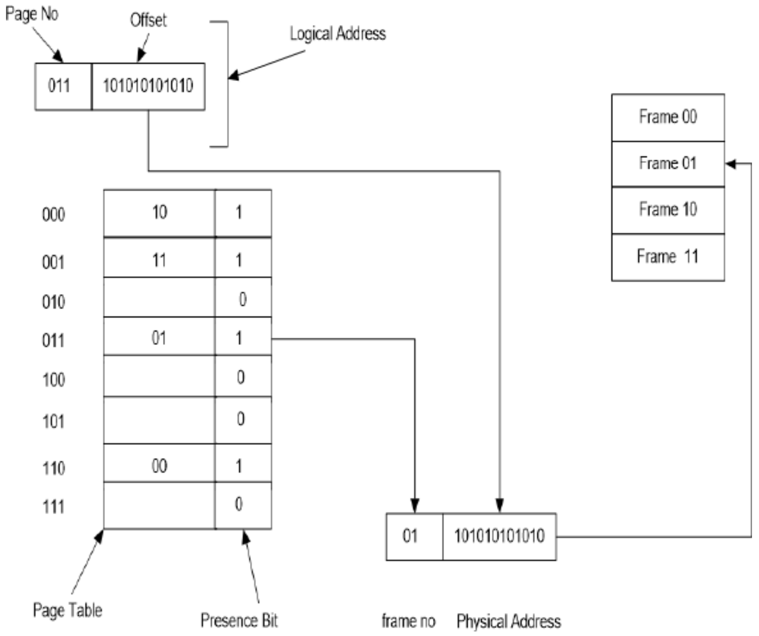
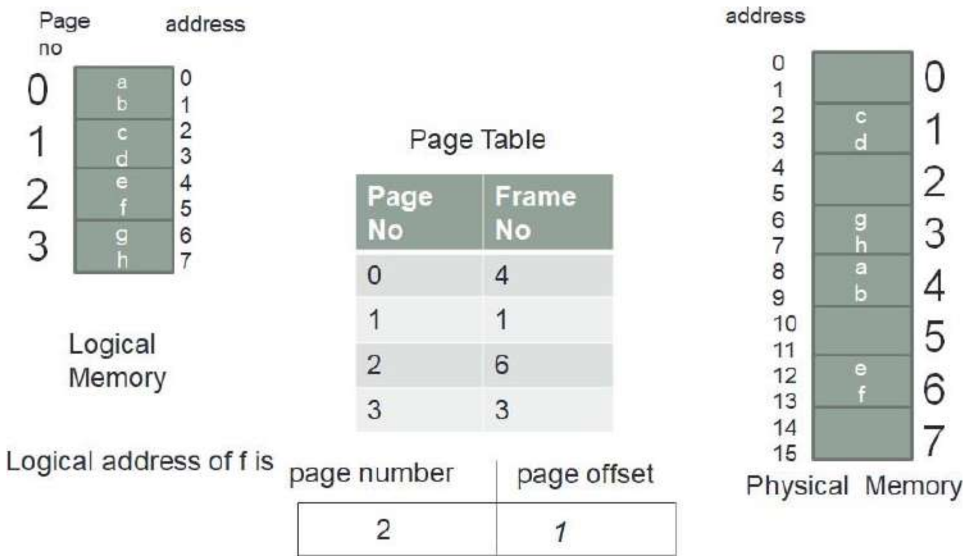


Figure: Mapping

Address Translation in a paging Scheme

The logical address 011 1010101010 is mapped into memory. From the page table, the operating system verifies the presence bit of cell 011, which is 1. This indicates that page 011 has been stored in a frame in the main memory.

Another example Using a page size of 2 bytes and a physical memory of 16 bytes (8 pages), we show how the user‘s view of memory can be mapped into physical memory.



Page No	Frame No
0	4
1	1
2	6
3	3

so physical address calculated (frame no x size)+ offset

The physical address of _f‟ calculated: _f‟ stores on page no 2 and offset is 1. page no 2 is loaded in frame 6.

So address is (6x2)+1 i.e. 13.

In a paging scheme, an operating system maintains a list of free frames. This is also known as **free frame pool, or chain or list**. Any frame that is not allocated to any process is known as a free frame. Whenever a page is released from memory after a process is terminated or aborted the page is added to the free list. When an operating system allocated pages to a process for storing instruction or data, it assigns a free page from the **free frames list**.

Structure of Page Table

Hierarchical Page Tables

With the increasing need for larger applications, there is also a need for larger memory space in a computer. As a result, it becomes difficult to work with long memory addresses. In paged memory system the page table itself becomes very lengthy. The solution is hierarchical paging.

In this, the page table itself is paged. There are two-page tables: outer and inner.

The inner page table is similar to that of a single-page table scenario. The outer page table stores the links for the inner page table. The logical address consists of three parts:

outer page table entry, inner page table entry, and offset.

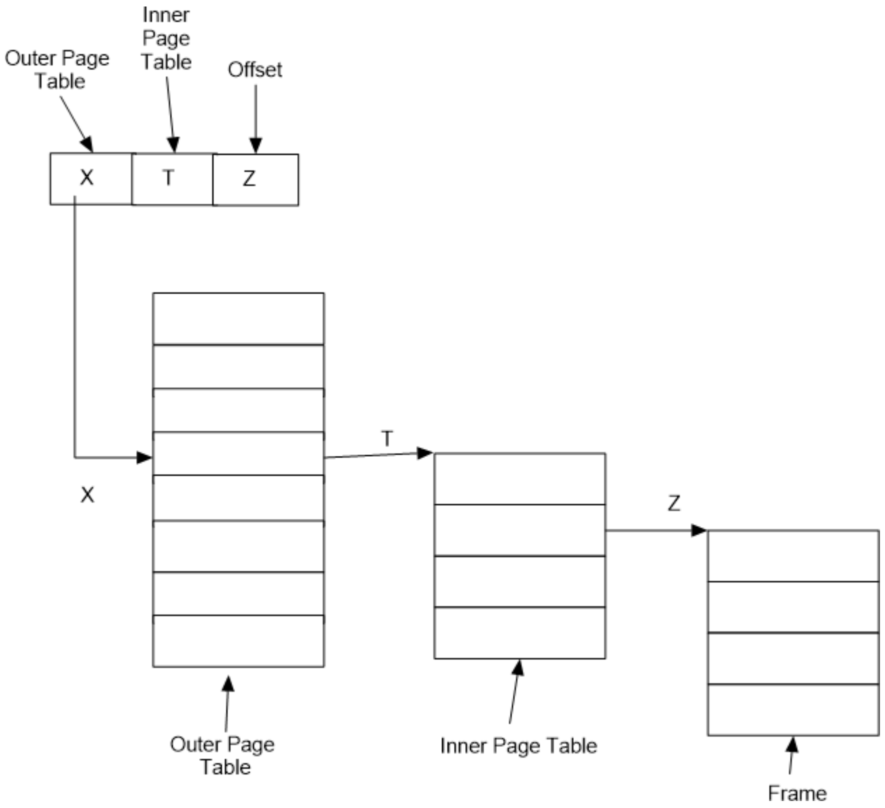


Figure: Hierarchical Page Table

In the process of hierarchical paging, a logical address has been divided into three parts: X, Y, and Z. the symbol X represents the entry in the outer page table, and Y represents the entry in the inner page table, Z represents the offset. The Xth position in the outer page table store the address of the inner page table associated with X. The Y position is searched in that inner page table and Y contains the address of the frame containing the searched page.

Hashed Page Table

This page table is created of length M. Whenever, logical address of generated, a hashing function is applied to the page Number p, to generate an index value i.

i = p % M;

The index value i is used to indexing into the page table. Each entry in the page table is a pointer to a link list. It will provide a mapping between page number p and the corresponding frame number f. It accessed through the index value i will be traversed, till a match forms for page number p and the corresponding frame number f is obtained.

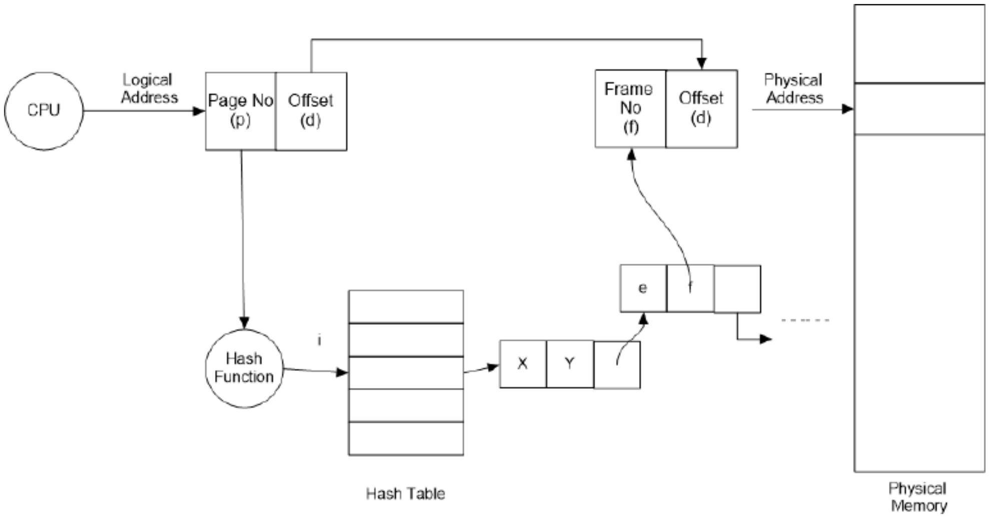


Figure Hashed Page Table

Inverted Page Table (IPT)

An inverted page table is similar to a simple page table along with another entry in the table i.e. process id along with page number. IPT is not process-specific and it does not need switching during content switching. A logical address generated by CPU contains:

Process id (P id),

Page number (p), and

Page offset (d).

Process id & page number is found and corresponding offset (d) which gives the frame number (f), where the desired page is residing. The frame number (f) combined with the offset d, gives the intended physical address.

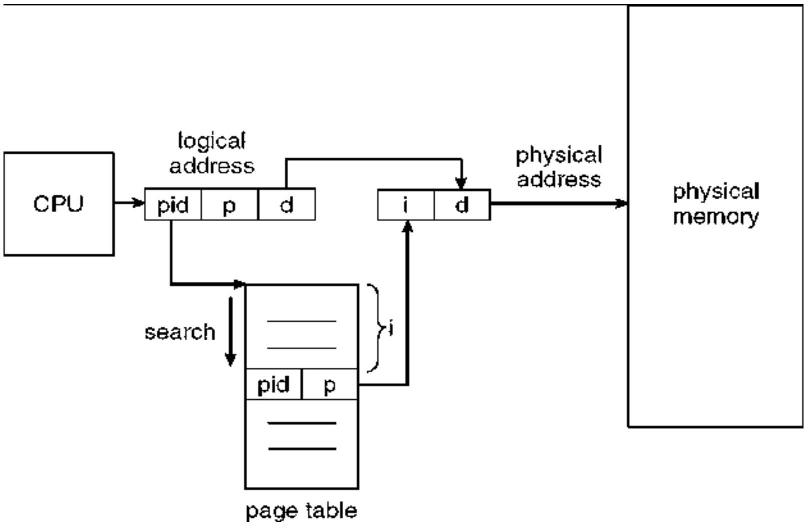


Figure: Inverted Page Table

Translation Look Aside Buffer (TLB)

The page table is implemented in the hardware can enhance the performance substantially. However, it does increase the cost proportionally. Though several schemes of a hardware implementation of the page table exist, in the simplest case, the page table is implemented as a set of dedicated registers. Care must be taken to employ very high-speed logic for these register otherwise the performance will not be up to the mark. The CPU dispatcher reloads these registers, just as it reloads the other registers. Instruction to load is modified the page-table register are, of course, privileged, so that only the operating system can change the memory map. This architecture is used in the DEC PDP-11 computers. It has a 16-bit address while the page size is 8K. As a result, the page consists of 8 entries that are kept in fast registers.

Having a dedicated set of registers for the implementation of the page table has a serious limitation. The register can be used for the purpose only when the page table is reasonably small. Modern computers are capable of having page tables containing as many as 1 million entries. Register implementation in such a system is grossly infeasible. The modern practice is to implement the page into the main memory rather than in the register. The page table is maintained in the main memory and is accessed by a pointer stored in the base register PTBR (Page Table Base Register). One advantage of this practice is that changing the page table requires only changing the register value thus avoiding time-consuming context switching.

This scheme is not without its downside. It suffers from memory access delay. With this approach, a memory location is accessed indirectly. First, the logical memory is indexed into the page table. The page table itself is accessed using the content of the PTBR register which provides the frame number, which is combined with the page offset to produce the actual address.

This approach requires two accesses to memory – one for the page table and one for the index into the page table thereby reducing the CPU time it may become unacceptable. The process can be accelerated using high-speed associative memory or a set of translation look-aside buffer (TLB_S). In associative memory, each register memory each register contains two entries – one for key and another for a value. During a search, the keys are matched with the search value and the corresponding value field is the result if the key was found. The use of associative memory enhances the search speed though at a price for the additional hardware support.

Even associative memory is not unlimited. Therefore, only a few entries of the page table are stored in the associative memory. When the CPU generates a logical address while executing a process the page number of the generated address is searched into the current associative memory for where the frame number is acquired. However, when the page number being searched does not exist in the associative memory the same is searched into the memory and the associative memory is updated with this page number so that it may be found in the next reference top the associative memory.

In this arrangement, a page being searched may not always be found in the associative registers. The extent to which a reference page is found in the associative memory is expressed in terms of hit ratio. Hit ratio is the percentage of times that a page number being searched is found in the memory. For instance, a hit ratio of 60% indicates that out of 100 times a page being searched it is found in the associative memory 60 times. Therefore, one of the performance goals of an operating system is to increase this hit ratio.

The time delay in memory access can be easily estimated in the following manner. Assuming that it takes 10 nanoseconds to search the associative registers, and 50 nanoseconds to access memory, the mapped memory access would take 10 + 50 = 60 ns provided the page number is found in the associative registers.

However, if the search ends in a miss in the associative register after spending 10 ns, then memory access is made for the page table and frame number taking additional 50 ns. Following this, the desired byte is accessed in the memory costing another 50 ns. The total time elapsed comes out to be 50 + 50 + 10=110 ns.

Example: Assuming a hit ratio of 60%, the effective or expected memory-access time can be calculated as shown here under:

Effective access time = (Probability of hit) * Access Time on hit +
(Probability of miss) * Access Time on miss
= 60% of 60 + 40% of 110

= 80 nanosecond

One way of increasing the hit ratio is to use more associative registers. However, as mentioned earlier it can be very costly. Therefore, usually, a trade-off is considered in most cases. One estimate indicates that a hit ratio of 80% to 90% can be achieved with 16 to 12 associative registers.

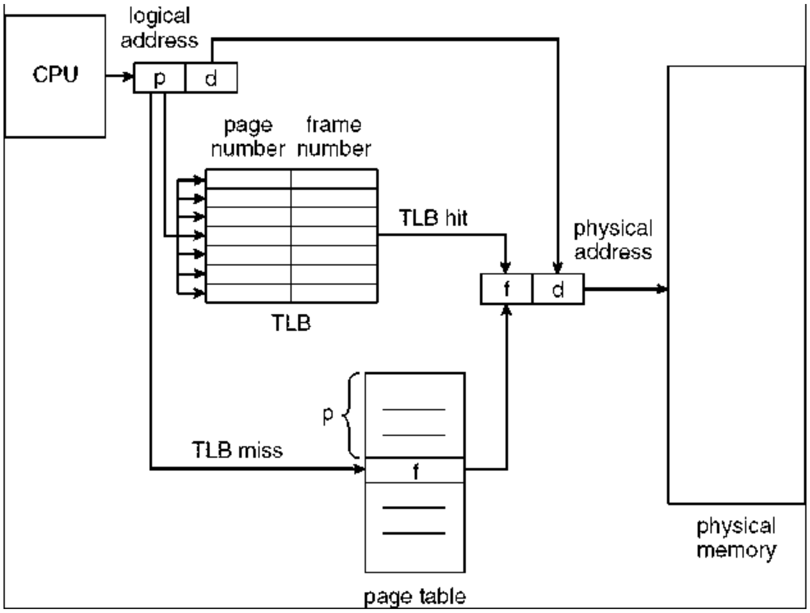


Figure: Paging with TLB

Protection

The issue of memory protection in a paging system is handled by attaching a few bits to each of the frames in the page table. The bit may indicate read-only or read-write or execute-only behaviors of the frame. At the time of accessing the frame number from the page table, the corresponding bits are also examined.

Here if a process attempts to write into a read-only frame a system interrupt is generated by the hardware. This interrupt is caught by the operating system, which in turn takes the memory protection violation action.

Notes

In addition to the protection bits, one more bit attached to each frame. This bit determines whether the frame/page is valid or invalid.

The –valid‖ status of the bit indicates that the associated page in the process‘s logical address space, and is thus a legal page. Otherwise, the page is not in the process‘s logical address space. Illegal addresses are trapped by the operating system using the valid-invalid bit. The operating system controls the accessibility of a frame by setting this bit.

Page A
Page B
Page C
Page D
Page E
Page F

Logical Memory

Page	Page No.	Valid/ Invalid Bit
A		I
B	3	V
C	7	V
D		I
E	1	V
F		I

0	
1	E
2	
3	B
4	
5	
6	
7	C

Main Memory

Page Table

Figure: Paging Protection

Memory Segmentation

Despite all the obvious merits of a memory-paging scheme of memory management, it does not reflect the way a programmer would like to view the memory. In a paging memory system, there is a clear-cut distinction between the users and system views of memory. The user‘s view of memory is not the same as the actual physical memory. The user‘s view is translated onto physical memory by the operating system with the use of special hardware. It is due to this translation that the logical view is different from the physical view of the memory.

A programmer would not like to think that the memory is simply a larger array of bytes wherein at someplace data is stored and at another executable code is stored. The most preferred view of the majority of programmers is that a program is divided into different sections or segments. If you have programmed in COBOL you would appreciate that a COBOL program has different divisions each having one or more optional sections.

Similarly, in a C program, the code is divided between many functions. Your program may have any number of functions, subroutines, data structures, and several modules. Each of these modules or data elements is assigned a unique name by the programmer. Where and how these elements are loaded in the memory is not the concern of the programmer. The programmer does care as to whether the segment containing symbols table is stored in the memory before the main function or after it. Each of these segments must be of variable length. The element of the program should be addressable by the respective names assigned to them.

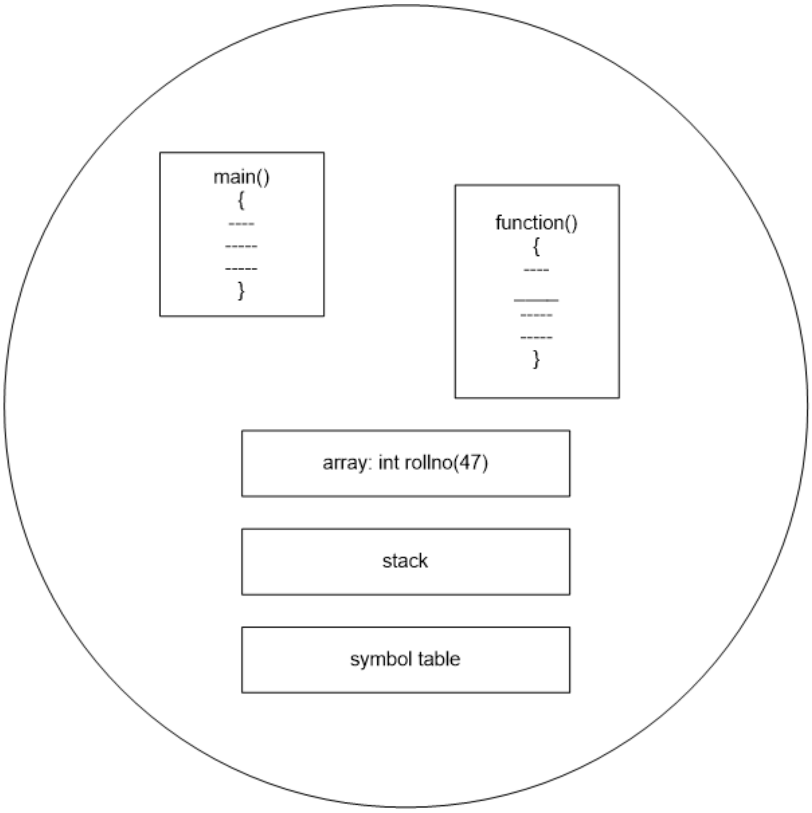


Figure: User's view of a program

An operating system may adopt a memory management scheme to simulate this view of memory.

Segmentation is a technique of non-contiguous memory management technique that employs this scheme. In this scheme, the logical memory space is divided into certain unequal size chunks known as **segments**. According to segmentation, an application is loaded into memory as a collection of modules. In other words, logically related instruction and data items are grouped and loaded into a segment.

For example, an application can consist of five modules: the main program, function A, function B, the stack for storing data when a function is called, and shared data for all modules such as a global variable. Each module is loaded in a distinct segment in the main memory.

Accessing Addresses in Segmentation:

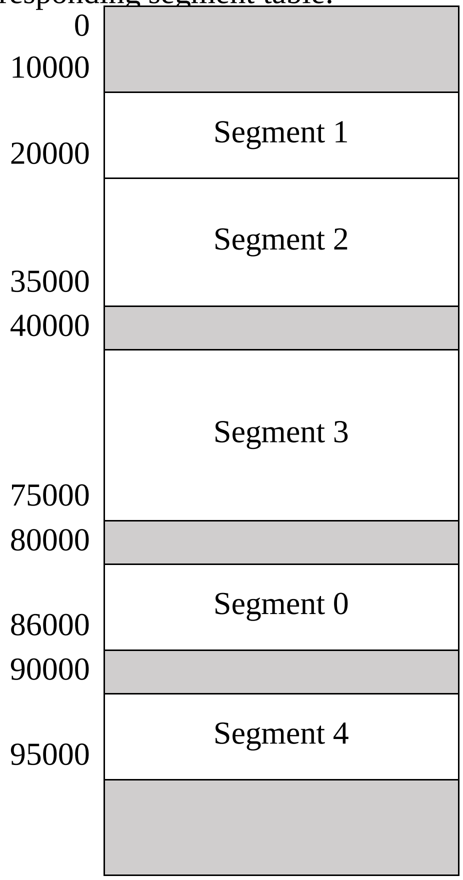
In a segmented memory, a logical address consists of two parts: segment number and offset.

s	d
Segment No	Offset

Figure Address of Segment

An offset is a location within a segment, where any particular data is stored. For example, the logical address 3: 10000 indicates offset 10000 and segment number 3. A segment table is used for the validation of an address. A segment table has two fields: base address of the segment, and size of the segment. The base address of a segment is the location of a particular segment memory and its corresponding segment table:

Segment No	Base Address of the Segment	Size of the Segment
0	80000	6000
1	10000	10000
2	20000	15000
3	40000	35000
4	90000	5000



▨ Free Space

Figure: Segment Table

Segment and Segment Table

The above figure segment table elaborates various memory segments in the main memory and their corresponding entries in the segment table. Segments need not be loaded sequentially or contiguously. For example, segment 2 has been loaded in a lower position in the memory than segment 0.

The starting address and size of a segment are stored in the segment table. For example, the base or starting address of segment 2 is 20000, and is the size of segment 2 is 15000 bytes.

When accessing an address in a segmented memory, the segment table is used for address validation. The logical address 2: 12000 indicates the location 12000 starting from the base of segment 2. Segment 2 starts from location 20000. The size of segment 2 is 15000 bytes. In other words, range of the segment 2 is 20000 to 74FFF. The physical address corresponding to the logical address 2: 12000 is 20000 + 12000=32000. the address 32000 is within the range of segment 2 (i.e. 20000 to 74FFF as 32000<74FFF). Hence, the logical address 2: 12000 is a valid address.

Implementation of Segment Tables

As in the case of the paging system, segmentation can be implemented in a variety of ways. The segment table can reside either in the register or in fast memory. The advantage of register implementation is that both the limits checking and referencing can be done extremely rapidly and simultaneously.

The feasibility of register implementation depends on the number of segments a program has. The larger the number the more register are required making the proposition infeasible. In smaller segmentation register implementation may be recommended. However, modern programs tend to become very large causing a large number of segments. Therefore, in such a system it is more appropriate to maintain the segment table in the memory rather than in the registers. The data structure used for implementation is often a **Segment-Table Base Register or STBR**. This pointer points to the segment table located in the memory. Note that the number of segments used by a program varies widely. Therefore, a Segment-Table Length Register or STLR is used. The address translation for a given logical address (s, d) proceeds as follows:

1. Check that the segment number is legal (i.e. S< STLR)
2. Add the segment number to the STBR(i.e., STBR+S)
3. Read entry from memory at STBR + S
4. Check the offset against the segment length (d< limit)

(a) if true then compute the physical address of the desired bytes as S+d

(b) else raise trap to the OS

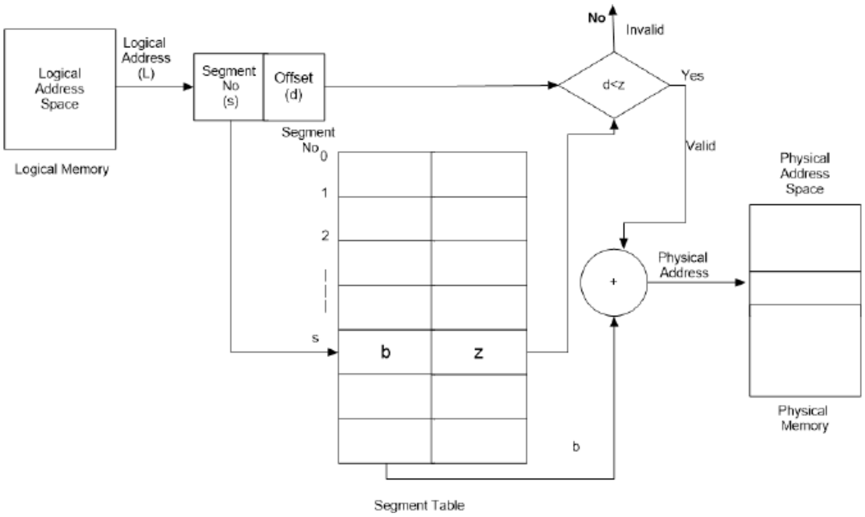


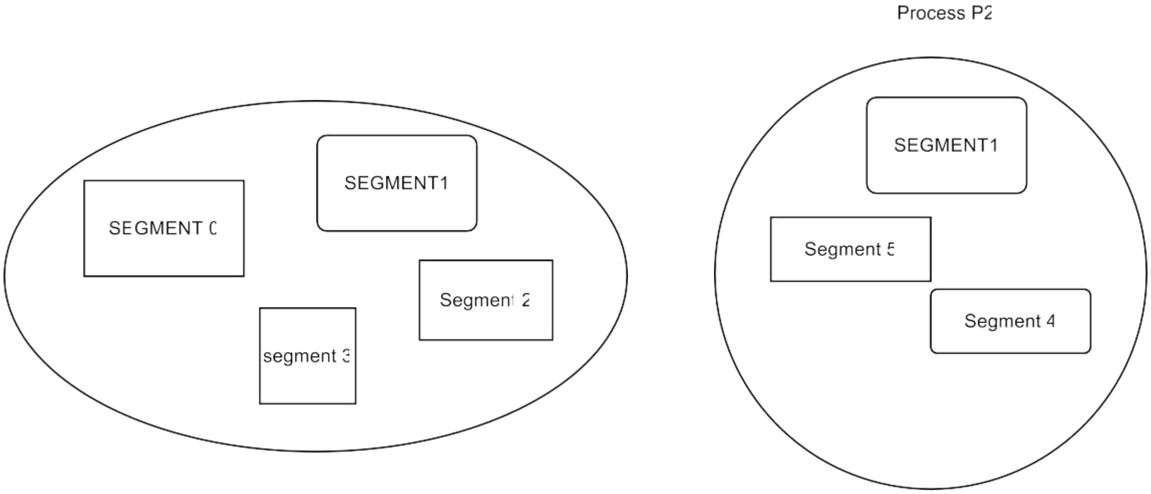
Figure: Segmentation Hardware

Both the schemes- paging and segmentation requires two memory references during address translation. This slows down the OS effectively by a factor of 2. As in the case of paging, a set of an associative registers can be used to speed up the operation. It has been found that a small set of associative registers generally reduces the time required for memory accesses.

Protection and Sharing

Segmentation enforces protection intuitively. Since each segment is a semantically cohesive unit all the corresponding entries are likely used similarly. Therefore, one needs to protect at the segment level using a few protection bits. For instance, since codes do not rewrite themselves the code segment can be assigned a read-only status whereas the data segment can be allowed to perform read and write operations. The memory mapping hardware will check the protection bit associated with each segment table entry to prevent illegal access to memory, such as attempts to write into a read-only segment or to use an execute-only segment as data. For example, by placing an array in its segment, the memory-management hardware will automatically check that array indexes are legal and do not stray outside the array boundaries. This enables the system to take corrective measures early on if a memory protection error does occur.

The main advantage of segmentation is that it can be shared between processes. This is true, particularly with code segments. For each process, a segment table is associated, which the dispatcher uses to define the hardware segment table when this process can share the same code segment as shown in the figure.

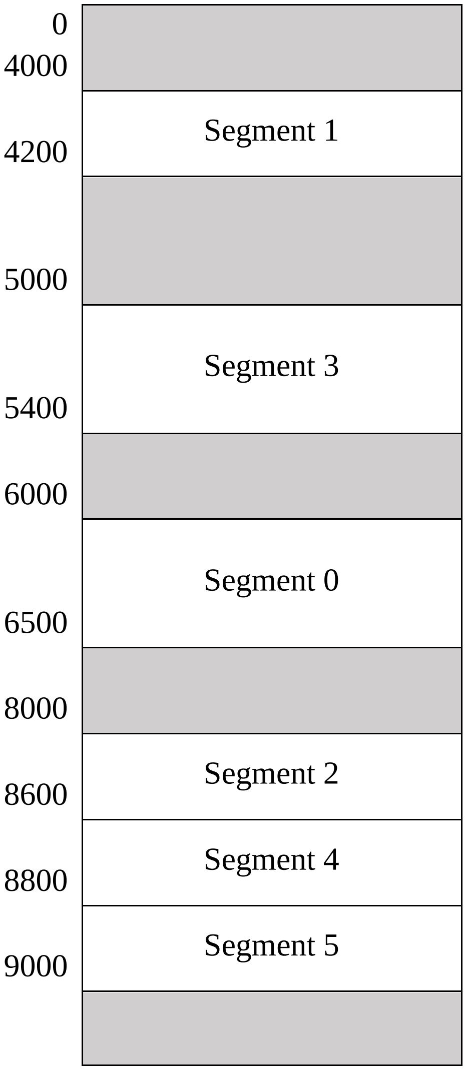


Segment Table P1

Segment No	Base Address of the Segment	Size of the Segment
0	6000	500
1	4000	200
2	8000	600
3	5000	400

Segment Table P2

Segment No	Base Address of the Segment	Size of the Segment
1	4000	200
4	8600	200
5	8800	200



▨

Free Space

Figure: Segmentation Sharing

Here Segment 1 is shared between P1 & P2

Sharing of the segment in a segment memory system

Here in this example, segment (1) is being shared between the two processes p1 and p2. Simply by managing entries in the segment table of processes a segment can be shared among them. The sharing occurs at the segment level. Thus, any information can be shared if it is defined to be segmented. Several segments can be shared in this way.

Segmentation with paging

Segmentation of memory can be implemented with a paging scheme. Unequal size segments can be represented by equal size pages. For example, in a logical address space of 64 KB, there are 32 pages of 2KB each. A program is divided into 3 segments of the size 15 KB, 18 KB, and 10 KB. These three segments require 8, 9, and 5 pages to be loaded into memory. The first segment occupies the memory space of 8 pages of 2 KB each or 16 KB, whereas its requirement is 15 KB memory space. Thus internal fragmentation of 1 KB occurs.

In segmentation with a paging scheme two tables are used: a segment table and a page table. The segment table has two fields: base address and size. The base address of a segment is the address of the first page in the segment. The size of a segment is the number of pages a segment occupies. The page table stores the addresses of the frame where a page is loaded into the main memory. For example, the value of the 15th cell of the page table is 38000. This means page 15 has been loaded into the frame located at the address 38000 in the main memory.

A logical address has three parts: segment number, page number, and offset. The logical address 4:3: 9E indicates the address 9E on page number 3 of segment number 4.

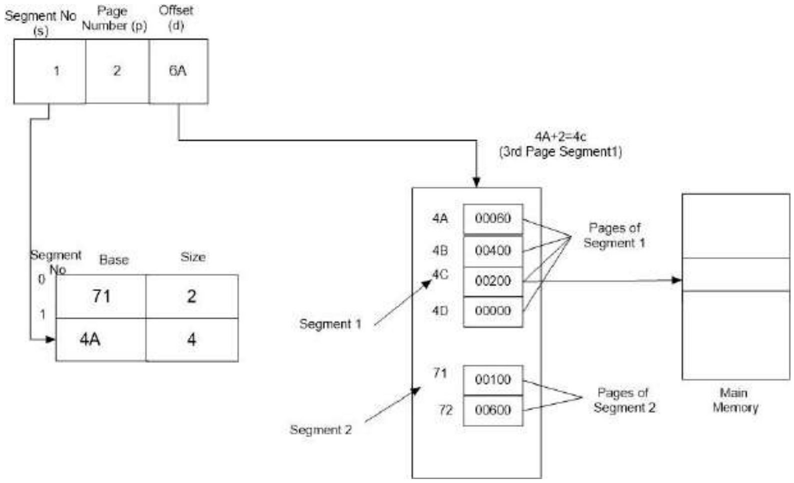


Figure: Segmentation with paging

The main memory is divided into certain equal size frames of 256 bytes (decimal 256 = 100 in hexadecimal) each. The address 6A of page number 2 of segment number 1 is being located. In the segment table, we found that the address of the first page in the page table of segment 1 is 4A. Segment 1 is consists of 4 pages.

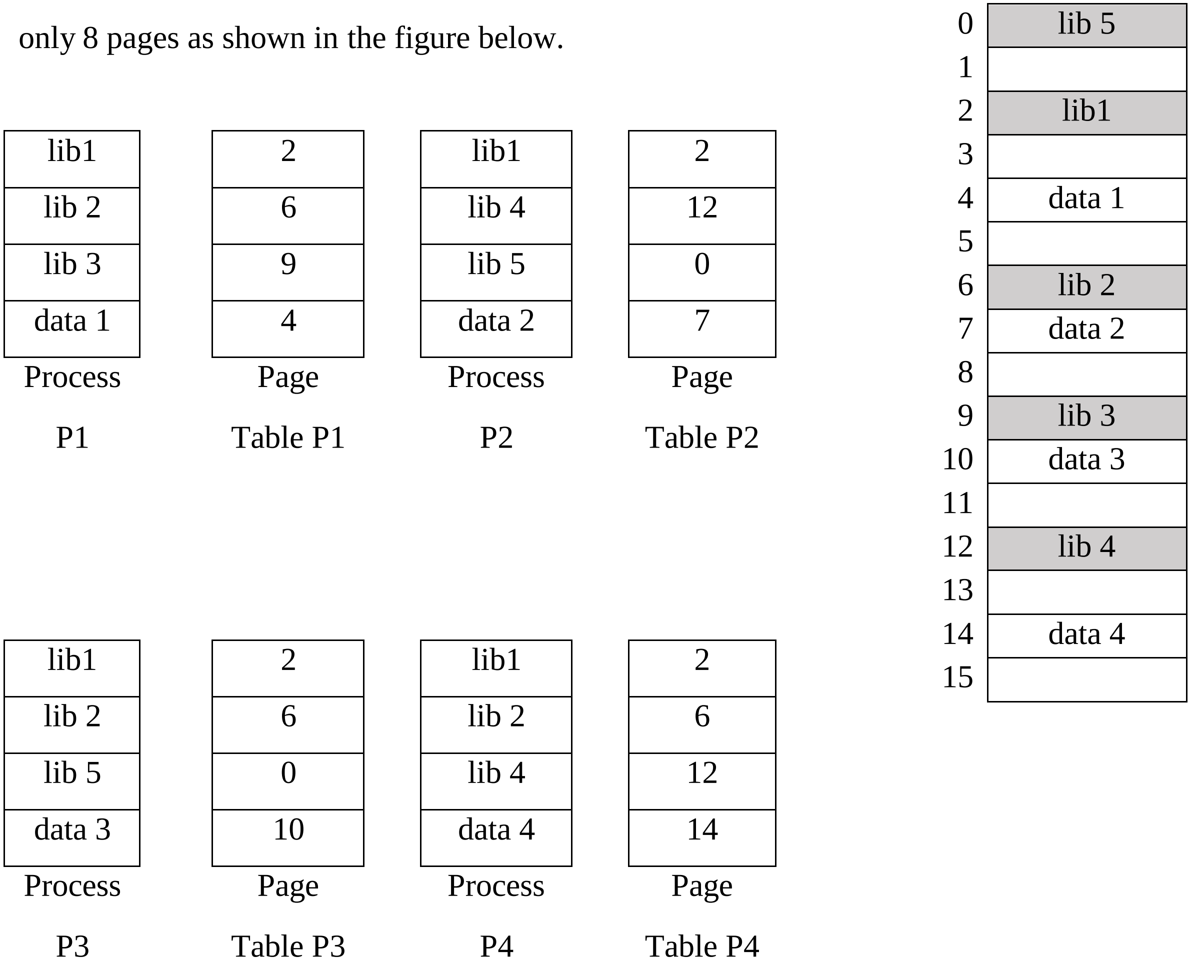
From the page table we see that address of the frame corresponding to page 2 has been stored in the location 4C in the page table (because 4a + 2 = 4C). The cell number 4C of the page table is loaded in the frame located at the address 00200 in the main memory.

Shared Pages

In multiprogramming, more than one program can be loaded. It is common for many users to be executing the same program. If individual copies of these programs were given to each user, much of the main memory would be wasted. Its solution is to share those pages that can be shared. One copy of read-only code shared among processes (i.e., text editors, compilers, window systems). Shared code must appear in the same location in the logical address space of all processes. Each process keeps a separate copy of the code and data. The pages for the private code and data can appear anywhere in the logical address space.

Example

Let, there are four processes P1, P2, P3, and P4. P1 has four pages lib1, lib2, lib3, and data1. P2 has four pages lib1, lib4, lib5, and data2. P3 has four pages lib1, lib2, lib5, and data3. P4 has four pages lib1, lib2, lib4, and data4. So, to load all processes 16 frames are required. In all these processes some pages are shared. So, if we share those pages and loaded all shared pages once by giving the same frame number, then it needs only 8 pages as shown in the figure below.



lib1	2	lib1	2
lib 2	6	lib 4	12
lib 3	9	lib 5	0
data 1	4	data 2	7
Process P1	Page Table P1	Process P2	Page Table P2

lib1	2	lib1	2
lib 2	6	lib 2	6
lib 5	0	lib 4	12
data 3	10	data 4	14
Process P3	Page Table P3	Process P4	Page Table P4

0	lib 5
1	
2	lib1
3	
4	data 1
5	
6	lib 2
7	data 2
8	
9	lib 3
10	data 3
11	
12	lib 4
13	
14	data 4
15	

Shared Pages among Processes

Figure: Shared Pages

Demand Paging

The basic concept of virtual memory is storing instructions and data of a program in the secondary memory, and then they are loaded in the main memory. In other words, the part of a program that is required at any instant is loaded in the main memory while the rest of the program is in the secondary memory.

Virtual Memory

Virtual memory is useful especially in a multitasking operating system, where memory available in the main memory for user application is divided among several user programs and every program has to compete for its memory requirement.

Virtual memory is a technique that permits the execution of processes, with their code only partially loaded into physical memory. Virtual memory is used for the separation of logical address space available to the user and the actual physical memory. CPU-generated addresses are known as **logical addresses** or **virtual addresses**.

Programmers use virtual addresses in applications. The MMU converts the virtual addresses into the corresponding physical memory address.

Programmers are notified that they can fully utilize the logical address space. Since the logical address space, programmers have the illusion that they have a larger memory space at their disposal.

Virtual memory is implemented using a non-contiguous memory allocation technique known as demand paging. This technique is similar to paging except that in-demand paging, pages are swapped in and out of main memory.

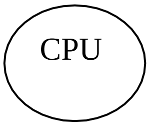
A program is initially stored in the secondary memory. When a page in the program is required, it is swapped into the main memory. This is called demand paging because until a page is not required, it is not loaded.

In demand paging, MMU also maintains an indexed table for address translation, known as a **page table**, similar to that of a simple paging scheme. The page table stores the information about, which page of a program is stored in which frame of the physical memory.

When a page is required, the operating system scans the page table to find out the physical address in the main memory that is in which frame the page is loaded. If the page is found, the operating system continues its processing. In case the page is not present in the main memory, a **page fault** occurs. A page fault is not a fault or an error; rather it indicates a situation that the page is requested by a program that is not currently loaded in the main memory.

When a page fault occurs, MMU swaps in the requested page from the secondary memory into the main memory. In case there is no free space in the main memory, MMU finds free space in the secondary memory, which is also known as the backing store in the context of virtual memory. MMU swaps out an unused page from the main memory and stores the unused page in the free space in the secondary memory for creating space in the main memory. The requested page is loaded into the main memory. Finally, the page table is updated.

Another way is a page table with a valid or invalid bit. The page table includes a valid or invalid bit for each entry. When a page is loaded in memory its frame number is entered and the page validity bit is set to valid. Thus if the bit is set to valid, it indicates that the page is in memory. If the page presence bit is invalid, it indicates that either page does not belong to the logical address space of the process, or it is still not loaded into memory.



Request Page 1 which is not in Page Table, So Page Fault occurs. Frame nowhere the pages stored in the memory

Page No	Frame No	Presence Bit
0	2	1
1		0
2		0
3		0
4	3	1
5	1	1
6	0	1
7		0

Page Table

Frame No	
0	Page 6
1	Page 5
2	Page 0
3	Page 4

Physical Memory

Page 3
Page 2
Page 7
Page 1

Secondary Memory

0	Page 0
1	Page 1
2	Page 2
3	Page 3
4	Page 4
5	Page 5
6	Page 6
7	Page 7

Logical Memory

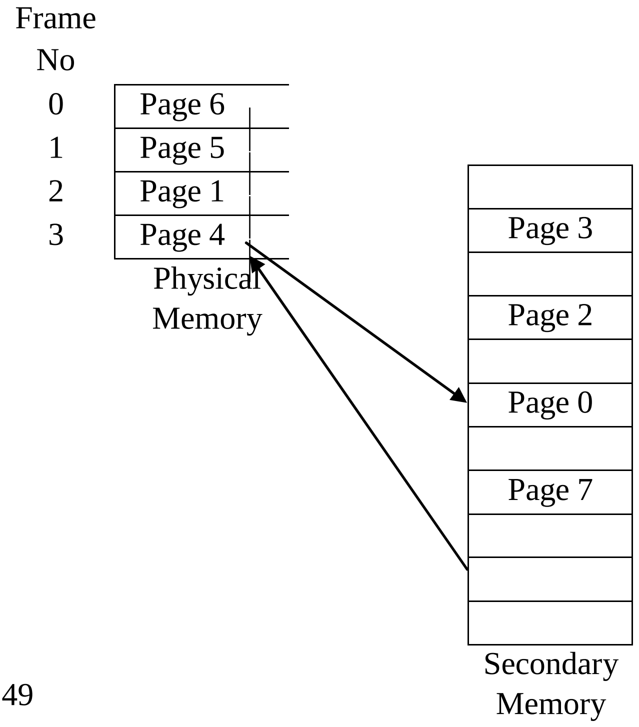
Figure: Before Page Replacement

Let, CPU request page 1. This page is not present in the page table because the presence bit of page1 in the page table is 0 that indicates the cell is empty. Thus page fault occurs.

When a page fault occurs, the operating system searches and loads that page in the memory replacing an existing page.

Page 1 caused a page fault. Page 0 is swapped out of the main memory and stored in the disk to create room for page 1 in the main memory. Then page 1 is swapped in the main memory.

Page No	Frame No	Presence Bit
0		0
1	2	1
2		0
3		0
4	3	1
5	1	1
6	0	1
7		0



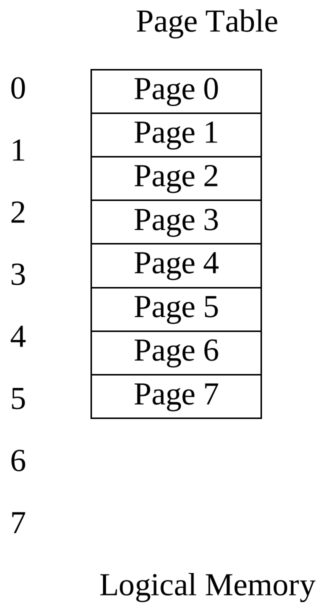


Figure: After Page Replacement

In the above figure, an abstract and basic idea of the functionality of virtual memory has been shown. Actual implementation varies in different operating systems.

Advantages

- It uses memory more efficiently.
- There is no limit on the degree of multiprogramming.

Steps in Handling a Page Fault

Initially, memory is empty, if there is a reference of a page, the first reference to that page will trap to an operating system which is page faults. To handle the page fault there are the following steps:

1. Operating system looks at another table to decide whether the demand is valid or not in memory. If it is invalid demand then it is aborted, else it is not in memory.
2. After that it searches the empty frame if it is not available then the victim finds and swaps out.
3. Demanded page is swapped into that frame.
4. Page tables are reset which assign the frame number to the corresponding page number.

5. Set the valid-invalid bit to valid bit.
6. Restart the instruction that caused the page fault.

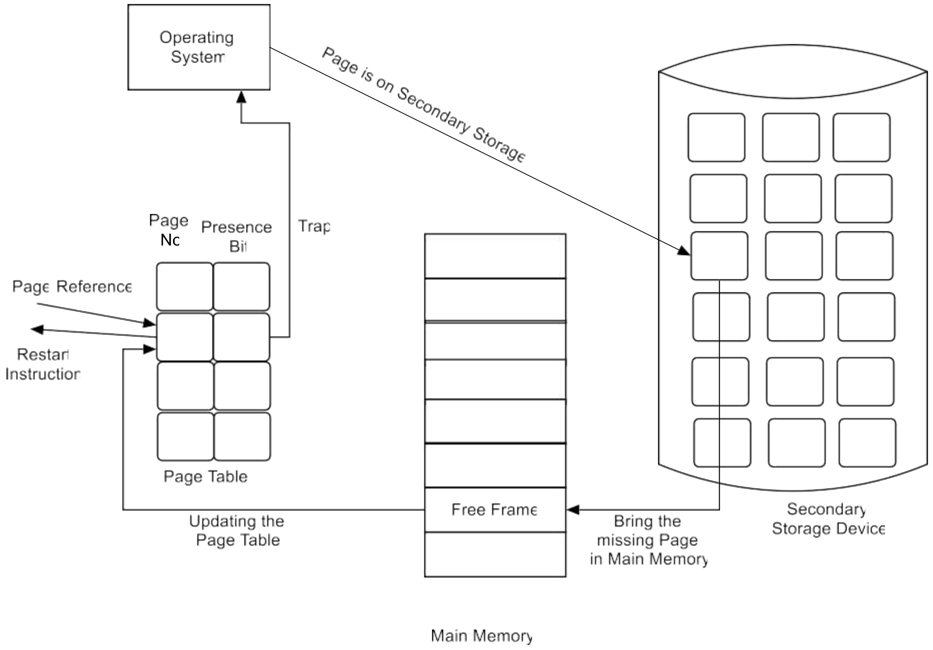


Figure: Steps in Handling Page Fault

Performance of Demand Paging

Demand paging can have a significant effect on the performance of a computer system. Let us calculate effective access time for a demand paged memory. Let P be the probability of a page fault $0 \leq P \leq 1.0$ if P = 0 no page faults and if P = 1 every reference is a fault.

Effective Access time =

$(1 - p) \times ma + pft$

p = page fault

ma = memory access time

pft = p × (page fault overhead + swap page out + swap page in + restart overhead)

EAT is directly proportional to the page fault rate. If the page fault rate is low then EAT is decreased otherwise.

Example: Let memory access time = 50 nanoseconds

Average page fault service time = 4 ms

Then EAT = $(1 - p) \times ma + pft$

$= 1 - p \times 50 + p \times (4 \text{ ms})$

$= 50 - 50p + 4000000p$

$= 50 + 3999950p$ (nanosecond)

Page Replacement

As the number of processes and the number of pages in the main memory for each process increase, at some point in time, all the page frames become occupied. At this time, if a new page is to be brought in, the OS has to overwrite some existing pages in the memory. The page to be chosen is selected by the page replacement policy.

A **page** is a fixed-length block of memory that is used as a unit of transfer between physical memory and external storage like a disk, and a **page fault** is an interrupt (or exception) to the software raised by the hardware when a program accesses a page that is mapped in address space but not loaded in physical memory. The hardware that detects this situation is the memory management unit in a processor. The exception handling software that handles the page fault is generally part of an operating system. The operating system tries to handle the page fault by making the required page accessible at a location in physical memory or kills the program in case it is illegal to access.

The hardware generates a page fault for page accesses where:

- The page corresponding to the requested address is not loaded in memory.
- The page corresponding to the memory address accessed is loaded, but its present status is not updated in hardware.

Example

Let program A have four pages i.e. P1, P2, P3, P4. Currently, P1, P2, P3 are in memory and it requests page 4 i.e. P4. Program B has four pages i.e. P8, P9, P10, P11.

Currently, P8, P10, P11 are in memory. Only six pages can be stored in memory and currently, P1, P2, P3, P8, P10, P11 are in memory so there is no memory space. To load P4, one page must be swapped out. Let it be P2, P2 store in frame 3 which is swapped out, and P4 is loaded in frame 3 which is currently free. The figure shows memory position before swapping page 4 i.e. P4 and the next figure shows the memory position after swapping page 4.

0	Page 1
1	Page 2
2	Page 3
3	Page 4

Logical Address Space of Program A

Page No	Frame No	Valid/ Invalid Bit
Page 1	4	v
Page 2	3	v
Page 3	6	v
Page 4		i

Page Table of Program A

Operating System	
Page 11	1
Page 8	2
Page 2	3
Page 1	4
Page 10	5
Page 3	6

Secondary Memory

0	Page 8
1	Page 9
2	Page 10
3	Page 11

Logical Address Space of Program B

Page No	Frame No	Valid/ Invalid Bit
Page 8	2	v
Page 9		i
Page 10	5	v
Page 11	1	v

Page Table of Program B

Figure showing the page table, logical address, and main memory before swapping

0	Page 1	
1	Page 2	
2	Page 3	
3	Page 4	

Logical Address Space of Program A

Page No	Frame No	Valid/ Invalid Bit
Page 1	4	v
Page 2		i
Page 3	6	v
Page 4	3	v

Page Table of Program A

0	Page 8
1	Page 9
2	Page 10
3	Page 11

Logical Address Space of Program B

Operating System	
Page 11	1
Page 8	2
Page 4	3
Page 1	4
Page 10	5
Page 3	6

Secondary Memory

Page No	Frame No	Valid/ Invalid Bit
Page 8	2	v
Page 9		i
Page 10	5	v
Page 11	1	v

Page Table of Program

Figure showing the page table, logical address, and main memory after swapping

Page Replacement Algorithms

When a page fault occurs, the operating system has to choose a page to remove from memory to brought a demanded page in memory.

A page replacement algorithm is logic or policy regarding how to select a page to be swapped out from main memory to create space for the page, known as the requested page, which has caused a page fault. These are several page replacement algorithms such as :

- First In First Out (FIFO)
- Optimal Page Replacement Algorithm
- Least Recently Used (LRU)
- Clock Algorithm (Second Chance Algorithm)
- Counting Based Algorithms
- Least Frequently Used (LFU)
- Most Frequently Used (MFU)

First In First Out (FIFO) Algorithm

The FIFO algorithm is the simplest of all the page replacement algorithms. It conveys a basic idea that when a page fault occurs, the oldest page in the main memory is to be swapped out of the main memory to create a room or the required page that needs to be executed. It replaces the page that has been in the memory longest.

> **The oldest page in the main memory is one that should be selected for replacement first. (if the number of frames is 3 then page repeated 3 times will replace)**

For example, let there are four pages i.e. 14, 21, 18, 36. Pages number 14, 21, and 18 are present in the memory and page number 21 was loaded first, followed by 14 and then 18. If page number 36 is required to be accessed by the CPU, it results in a page fault because page number 36 is not present in the main memory. The page loaded first, that is page number 21 is swapped out of the main memory and stored in the secondary memory to create room for page number 36. Then page 36 is loaded in the memory. One possible implementation is a FIFO queue of existing pages in memory. The oldest page will be at the FRONT of the queue. Whenever a page fault occurs, the page at the FRONT of the queue is mode victim and the new page is put at the REAR of the queue.

Advantages

It is very simple and easy to implement.

Programmers can easily code this algorithm.

Disadvantages

This algorithm has a severe drawback. If the oldest page is accessed frequently, the performance of this algorithm declines, since whenever the page swapped out will be required, it will result in another page fault, which degrades the performance.

Reference String

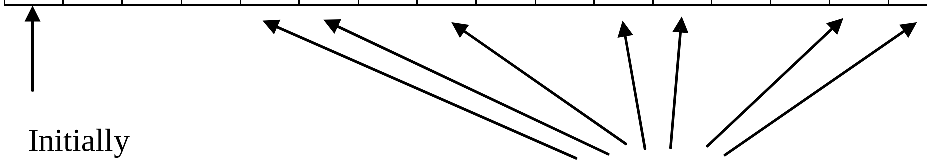
A reference string refers to the sequence of page numbers referenced by a program during its execution.

Assume a reference string:

1, 2, 3, 2,1,4, 2,1, 6, 1, 4, 2,5, 6, 2,1, 6, 3, 2

Let frames 3 are available for allocation page frame. Find number of page faults.

	1	**2**	**3**	**2**	**1**	**4**	**2**	**1**	**6**	**1**	**4**	**2**	**5**	**6**	**2**	**1**	**6**	**3**	**2**
	1	**1**	**1**			**4**		**4**	**4**			**2**	**2**			**2**	**6**	**6**	**6**
		2	**2**			**2**		**1**	**1**			**1**	**5**			**5**	**5**	**3**	**3**
			3			**3**		**3**	**6**			**6**	**6**			**1**	**1**	**1**	**2**



empty Frames No Page Fault

Initially, all 3 slots are empty, so when 1, 2,3 came they are allocated to the empty slots —> **3 Page Faults.**

Next page 2 refer, it is already in memory so —> **0 Page Faults.**

next page 1 refers, it is already in memory so —> **0 Page Faults.**

next page 4 refer, it is not available in memory and no free frame so it replaces the oldest page slot i.e 1. —>**4th Page Fault**

Next page 2 refer, it is already in memory so —> **0 Page Faults.**

Next page 1 refers, it is not available in memory and no free frame so it replaces the oldest page slot i.e 2. —>**5th Page Fault**

next page 6 refer, it is not available in memory and no free frame so it replaces the oldest page slot i.e 3. —>**6th Page Fault**

Next page 1 refers, it is already in memory so —> **0 Page Faults.**

Next page 4 refer, it is already in memory so —> **0 Page Faults.**

next page 2 refer, it is not available in memory and no free frame so it replaces the oldest page slot i.e. 4 —>**7th Page Fault**

next page 5 refer, it is not available in memory and no free frame so it replaces the oldest page slot i.e. 1 —>**8th Page Fault**

Next page 6 refer, it is already in memory so —> **0 Page Faults.**

Next page 2 refer, it is already in memory so —> **0 Page Faults.**

next page 1 refers, it is not available in memory and no free frame so it replaces the oldest page slot i.e. 6 —>**9th Page Fault**

next page 6 refer, it is not available in memory and no free frame so it replaces the oldest page slot i.e. 2 —>**10th Page Fault** next page 3 refer, it is not available in memory and no free frame so it replaces the oldest page slot i.e. 5 —>**11th Page Fault**

next page 2 refer, it is not available in memory and no free frame so it replaces the oldest page slot i.e. 1 —>**12th Page Fault**

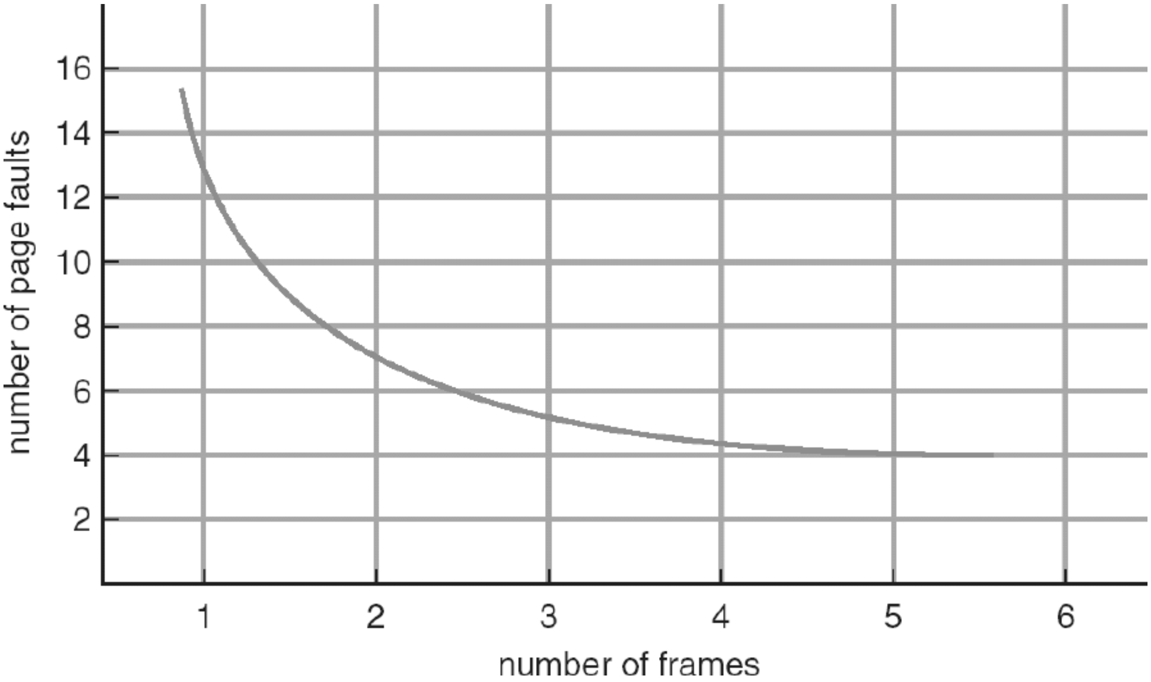
The number of page faults in FIFO is 12

Let frames 4 are available for allocation:

	1	**2**	**3**	**2**	**1**	**4**	**2**	**1**	**6**	**1**	**4**	**2**	**5**	**6**	**2**	**1**	**6**	**3**	**2**
	1	**1**	**1**			**1**			**6**	**6**		**6**	**6**					**3**	
		2	**2**			**2**			**2**	**1**		**1**	**1**					**1**	
			3			**3**			**3**	**3**		**2**	**2**					**2**	
						4			**4**	**4**		**4**	**5**					**5**	

Page fault in FIFO with 4 frames is 9.

By observing these two examples it states that if the number of frames will increase then the number of page fault must be decrease



Graph of Page Faults Versus The Number of Frames

The above graph reveals that if the number of frames increases then the number of page fault decreases but Belady‘s anomaly proves that if the number of frames increases then it may be increased in page fault also while using the First in First Out (FIFO) page replacement algorithm.

Let string is 5, 6, 7, 8, 5, 6, 9, 5, 6, 7, 8, 9, 5, and frameset is 3.

	5	**6**	**7**	**8**	**5**	**6**	**9**	**5**	**6**	**7**	**8**	**9**	**5**
	5	**5**	**5**	**8**	**8**	**8**	**9**			**9**	**9**		**5**
		6	**6**	**6**	**5**	**5**	**5**			**7**	**7**		**7**
			7	**7**	**7**	**6**	**6**			**6**	**8**		**8**

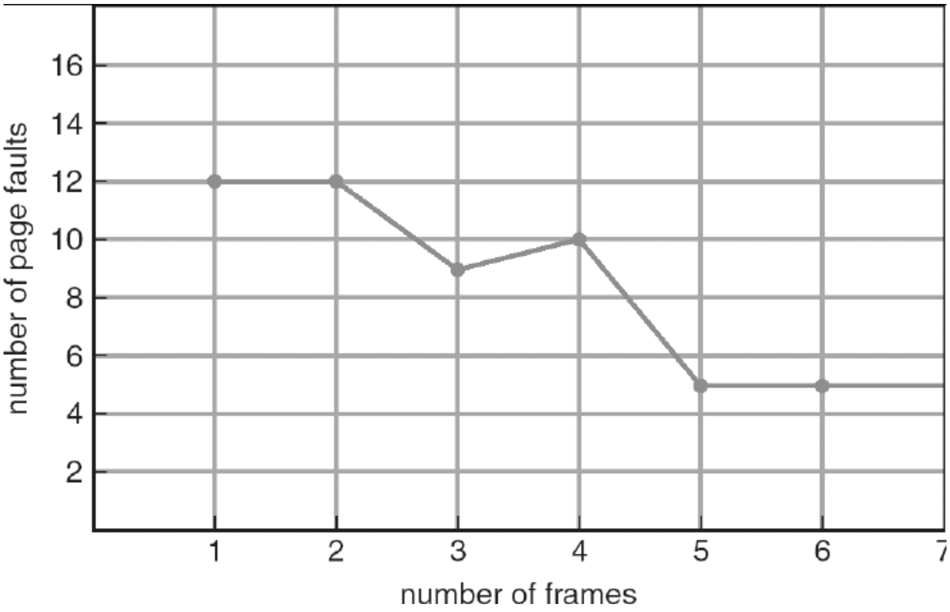
There are 10-page faults

Now with 4 frames set

	5	**6**	**7**	**8**	**5**	**6**	**9**	**5**	**6**	**7**	**8**	**9**	**5**
	5	**5**	**5**	**5**			**9**	**9**	**9**	**9**	**8**	**8**	**8**
		6	**6**	**6**			**6**	**5**	**5**	**5**	**5**	**9**	**9**
			7	**7**			**7**	**7**	**6**	**6**	**6**	**6**	**5**
				8			**8**	**8**	**8**	**7**	**7**	**7**	**7**

there are 11-page faults.

So by the increasing number of frames, it may be the increase of the number of page faults like in the above example where with 3 frames set there are 10-page faults and with 4 framesets there are 11-page faults. This is the Belady‘s Anomaly.



FIFO Illustrating Belady's Anomaly

The Optimal (OPT) Page Replacement Algorithm

Optimal page replacement algorithm is considered the best possible page replacement policy in a virtual memory theoretically but it is difficult to implement. According to the optimal page replacement policy, the page in the main memory, which will not be referred to for the longest time is swapped out from the main memory to create room for the requested page. A page should be replaced, which is to be referenced in the most distant future. Since, it requires knowledge of the future reference string, which is not practical. In the optional page replacement algorithm, the number of page faults is minimum as compared to another algorithm.

Practically, implementation of the optimal page replacement algorithm is done as follows:

Usually, all pages are labeled with the number of instructions that will be executed before this page will be used again in the future. When a page fault occurs the page with the highest number is replaced with the requested page that has caused a page fault.

For example, let pages 14, 21, and 18 are present in the main memory. Page 82 is required to be loaded resulting in a page fault since page 82 is not there in the memory at present. Page 14 is not required till the next 2000 instruction. Page 21 is not required till the next 1500 instructions. Page 18 is not required till the next 1700 instructions. This means page 14 is the one that will not be accessed by the CPU for the longest time. Page 14 is swapped out of the main memory to create room for page 82.

Advantages

It has the lowest rate of occurrence of page faults.

It improves the system performance by reducing overhead for several page faults and swapping pages in and out when a page fault occurs.

Disadvantages

It is very difficult to implement.

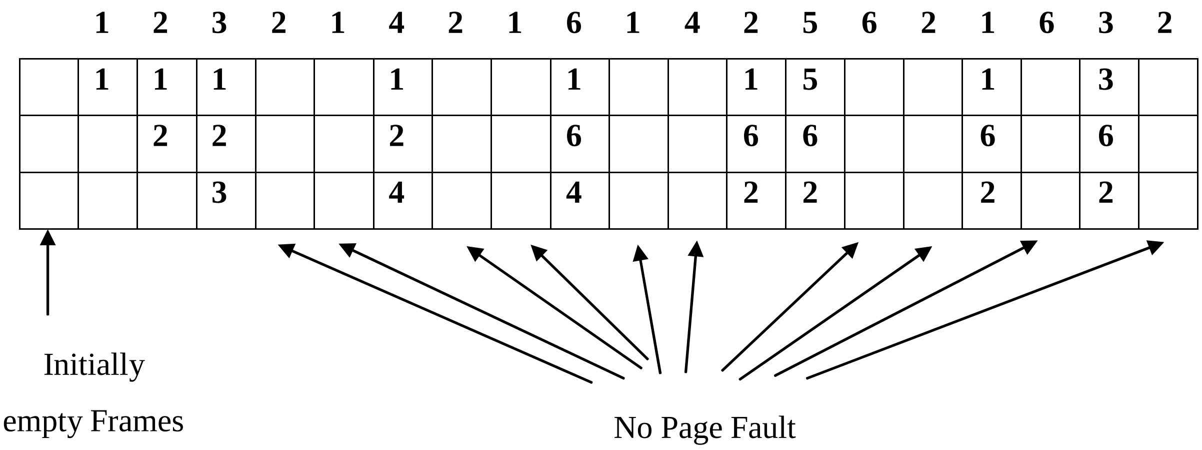
The situation is very similar to that of implementing the SJF algorithm in process management. It becomes very difficult for an operating system to calculate after what interval a page is to be referred to.

Example:

Let a reference string as follow:

1, 2, 3, 2,1,4, 2,1, 6, 1, 4, 2,5, 6, 2,1, 6, 3, 2

Let frames 3 are available for allocation page frame. Find number of page faults.



	1	**2**	**3**	**2**	**1**	**4**	**2**	**1**	**6**	**1**	**4**	**2**	**5**	**6**	**2**	**1**	**6**	**3**	**2**
	1	**1**	**1**			**1**			**1**			**1**	**5**			**1**		**3**	
		2	**2**			**2**			**6**			**6**	**6**			**6**		**6**	
			3			**4**			**4**			**2**	**2**			**2**		**2**	

There is a 9-page fault in the optional page replacement algorithm but in FIFO there are 12-page faults.

Least Recently Used (LRU)

The LRU algorithm uses information about the pages accessed in the recent past to predict the near future. The LRU algorithm is when a page fault occurs, the page that has not been referred to for the longest time is swapped out of the main memory to create space for the requested page that has caused the page fault. It replaces the page which has been used least recently.

Implementation of LRU can be done in various ways. One of the common methods to apply the LRU in a scheme for virtual management is using an array. The array stores the information about the page present in the main memory. The front end of the array stores the page accessed recently. The rear end of the array stores the page that has not been accessed for the longest time.

Whenever a page, that is present in the main memory, is accessed, the information about the page in the array is shifted to the front end of the array. If a page fault occurs the page indicated by the rear end of the array is swapped out of the main memory and the requested page is swapped in the main memory. Information about the page swapped in is stored in the front end of the array.

For example, the array stores four-page numbers {12, 56, 27, 61}. Page 12 is at the front end of the array. This indicates that page 12 has been accessed recently. Page 61 is at the rear end of the array. This indicates that the page that has not been referenced for the longest time is page 61. Page 27 is accessed. No page fault occurs because page 27 is present as the main memory. Information about page 27 is shifted at the front end of the array. The array becomes { 27, 12, 56, 61}. Page 43 is required to be accessed. A page fault occurs since page 43 is not in the main memory. Page 61 is at the rear end of the array, which indicates that it has not been accessed for the longest time. Page 61 is swapped out of the main memory and information about page 61 is removed from the array. Page 43 is swapped in the main memory and information about page 43 is inserted at the front end of the array. The array becomes { 43, 27, 12, 56}.

Advantages

It is very feasible to implement.

This is not as simple as the FIFO algorithm but not as complicated to implement as the optimal page replacement algorithm.

Disadvantage

The LRU algorithm requires additional data structure and hardware support for its implementation.

Let a reference string as follow:

1, 2, 3, 2,1,4, 2,1, 6, 1, 4, 2,5, 6, 2,1, 6, 3, 2

Let frames 3 are available for allocation page frame. Find number of page faults.

	1	**2**	**3**	**2**	**1**	**4**	**2**	**1**	**6**	**1**	**4**	**2**	**5**	**6**	**2**	**1**	**6**	**3**	**2**
	1	**1**	**1**			**1**			**1**		**1**	**1**	**5**	**5**		**1**		**1**	**1**
		2	**2**			**2**			**2**		**4**	**4**	**4**	**6**		**6**		**6**	**6**
			3			**4**			**6**		**6**	**2**	**2**	**2**		**2**		**3**	**2**

Initially
empty Frames

No Page Fault

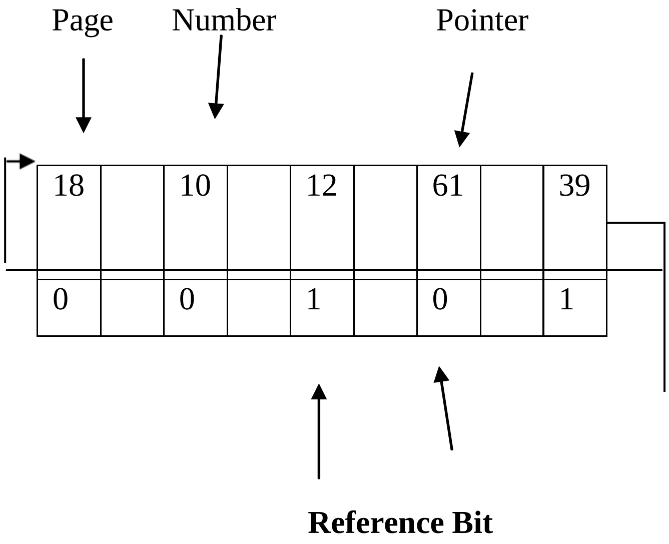
Page fault in LRU is 12. In this case, it is higher than optimal because a page that swap out is needed after one page so, it increases the page fault but in normal case page fault in LRU is less than FIFO but greater than the optional page replacement algorithm.

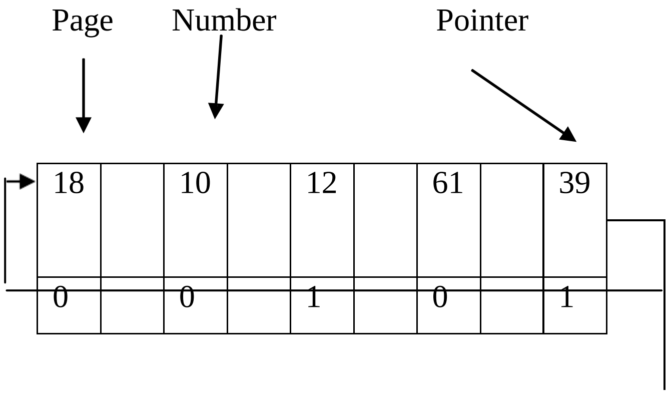
Clock algorithm

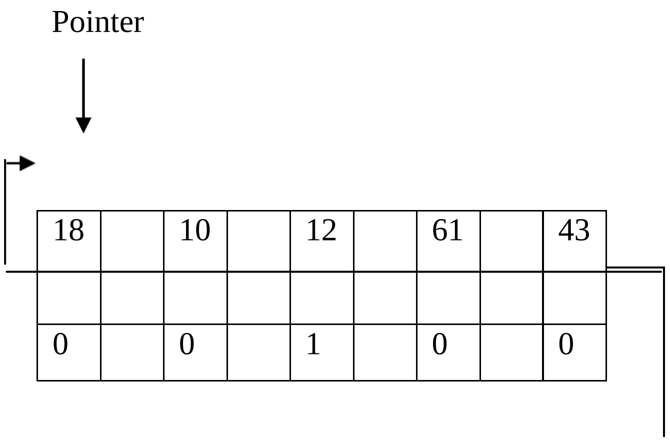
The clock algorithm also known as the second chance algorithm is a variation of the FIFO algorithm. In this, the page present for the longest time in the memory is given a second chance to remain loaded in the main memory. When that page is encountered for the second time, is swapped out to create room for the page that has caused a page fault. It is also referred to as **Not Recently Used (NRU)**. It replaces a resident page, which has not been accessed in the near past.

Working

This is implemented using the concept of a circular queue. Each cell in the circular queue contains two values: a page number and its corresponding **reference bit**. The value of the reference bit can be either 0 or 1. If the value of a reference bit of a page is 1 it means that the page was encountered as the oldest page and followed a second chance. If the value of the reference bit is 0 this indicates that this page has not been encountered as the oldest page yet. When a page is found the oldest page present in the memory for the first time, its reference bit is set from 0 to 1. The next time when that page is found, it is swapped out of the main memory for creating free space in the main memory.







Page 43 swapped in from the secondary memory replacing page 39

A circular chain or pull is organized to implement the clock algorithm for selecting a page to be swapped out in a virtual memory scheme. The pointer is directing presently at page number 61. a page fault occurs. Page 61 has not been given a chance to remain loaded in the memory since its references bit is 0. The reference bit of page 61 is made 1 and the pointer is moved to the next page in the pull that is page 39.

The reference bit on page 39 is 1. This indicates that previously page 39 has been given a chance to be remain loaded in the memory. Thus this page is swapped out of the memory to create space for the new to swapped on page 43.

Counting Based Algorithms

Some page replacement algorithms apply the logic based on how many times a page has been accessed. Usually, these types of algorithms are implemented using an array that stores information about a page number and the number of times it has been accessed. The type of counting algorithms is:

Least Frequently Used (LFU) page replacement algorithm

Most Frequently Used (MFU) page replacement algorithm

When a page fault occurs, the operating system verifies how many times it has been accessed and proceeds according to the logic provided by the counting algorithm adopted for page replacement in the virtual memory by the operating system.

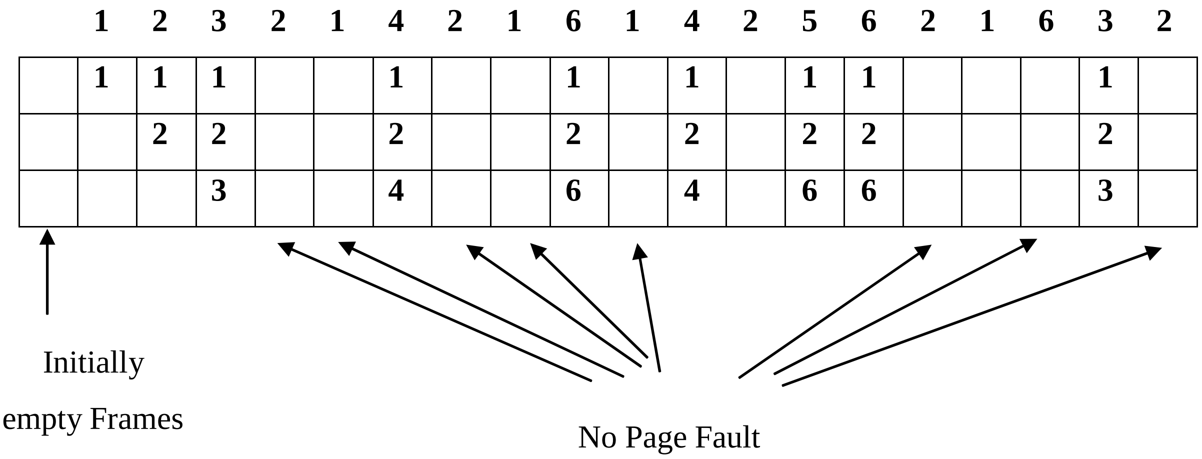
Least Frequently Used (LFU)

In the LFU algorithm, the page that has been accessed for the lowest/fewest number of times from the time when the page is loaded in the memory is replaced when a page fault occurs. The logic behind applying this algorithm is that some pages are accessed more frequently than others. Counting how many times a page has been accessed is used as an estimate of the probability of a page being referenced. Whenever a replacement is necessary, a page with the least count is replaced. The main drawback of this algorithm is that some pages may have a high usage initially and may build a high count but they have low usage subsequently, would remain in memory due to high count.

Let a reference string as follow:

1, 2, 3, 2,1,4, 2,1, 6, 1, 4, 2,5, 6, 2,1, 6, 3, 2

Let frames 3 are available for allocation page frame. Find number of page faults.



	1	**2**	**3**	**2**	**1**	**4**	**2**	**1**	**6**	**1**	**4**	**2**	**5**	**6**	**2**	**1**	**6**	**3**	**2**
	1	**1**	**1**			**1**			**1**		**1**		**1**	**1**				**1**	
		2	**2**			**2**			**2**		**2**		**2**	**2**				**2**	
			3			**4**			**6**		**4**		**6**	**6**				**3**	

There are 9-page faults is LFU

Note: If the counter of all the pages is the same then we adopt FIFO.

Most Frequently Used (MFU)

This algorithm replaces the page with the largest usage count. It is based on the assumption that the pages, with a smaller count, have been brought in recently and would need to be resident.

1 2 3 2 1 4 2 1 6 1 4 2 5 6 2 1 6 3 2

	1	**1**	**1**			**4**		**4**	**4**	**4**		**4**	**4**		**2**	**1**		**3**	**3**
		2	**2**			**2**		**1**	**6**	**6**		**6**	**6**		**6**	**6**		**6**	**2**
			3			**3**		**3**	**3**	**1**		**2**	**5**		**5**	**5**		**5**	**5**

Initially
empty Frames

No Page Fault

There are 13-page fault is MFU

Note: If the counter of all the pages is the same then we adopt FIFO.

Thrashing

In certain situations, a system spends more time processing page fault by swapping in and out pages than executing an instruction of processes. In other words, sometimes handling the pages faults becomes a huge overhead. This situation is called **thrashing**. Thrashing degrades the performance of a system. When multiprogramming increase more which degrades the performance is known as thrashing.

In a situation, where too many pages are active and required frequently, a page is required that is not present in the memory resulting in a page fault. One of the pages loaded in the main memory, which is also accessed frequently, is swapped out to create room for the required page. The required page is swapped in; the page swapped out is required for execution causing another page fault. Thus, a series of pages fault occurs and swapping becomes a large overhead.

In case a large number of processes are running in a multitasking operating system simultaneously, the memory becomes over-committed and thrashing occurs. In another word, the degree of multiprogramming is directly related to thrashing. The term, degree of multiprogramming, indicates many processes that are being simultaneous. This figure shows how system performance declines because of thrashing:

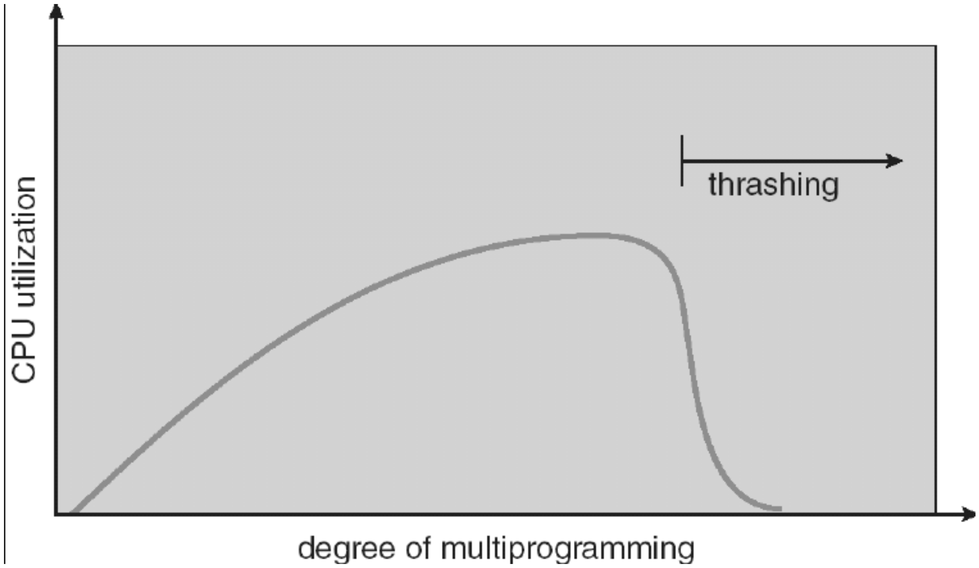


Figure: Effect of Thrashing on System Performance

This figure shows that system performance in terms of throughput reaches an optimal level to an extent if the degree of multiprogramming is increased. If the degree of multiprogramming is further extended thrashing occurs and system performance degrades. The peek of the graph represents the optimal performance of the system.

In certain cases, thrashing can be avoided. In most cases, an operating system attempt to recover from thrashing by suspending the execution of current processes and preventing the execution of new processes to start.

Selecting an appropriate page replacement policy to operate virtual memory plays an important role in the paging of thrashing. A page replacement strategy based on the local mode prevents the occurrence of thrashing to an extent. In local mode, while running a process, When a page fault occurs, swapping out of a page, which belongs to another process, is not allowed. A page belonging to the same running process is selected and swapped out to create room for the required page. In such cases, the thrashing of a particular process does not affect other processes. If other processes have occupied enough frames in the main memory, they can continue execution properly.

A page replacement policy based on the global model can lead the system to thrash. In global mode, when a page fault occurs, any page, regardless of which process it belongs to, is swapped out to create room for the required process that needs immediate execution. In global mode, a process may occupy a huge part of the main memory while other processes are competing for room in the main memory and hence resulting in page fault frequently.

Points to Remember

- The logical address is generated by the CPU.
- The logical address is also known as the virtual address.
- Logical and Physical addresses are different for execution time address binding.
- The base register is also called the relocation register.
- The user program deals with the logical address.
- Swapping is a technique to temporarily removing the inactive program from the memory of a system.
- Memory allocation methods are: first fit, Best fit, Worst fit.
- Paging is a memory management method.
- Logical memory divided it into similar size called pages.
- Physical memory divided it into similar size called frames.
- Paging eliminates fragmentation.
- In segmentation, the program is divided into variable size segments.
- Paging and segmentation can be shared.
- Virtual memory allows the execution of partially loaded processes.
- Virtual memory is the separation of user logical memory from physical memory.
- Demand paging is similar to a paging system with swapping.
- With the use of virtual memory, CPU utilization can be increased.
- In demand paging, pages are swapped in and out of the main memory.
- In demand paging, MMU also maintains an indexed table for address translation,

known as a page table.

- This table is similar to that of a simple paging scheme.
- When the page is not present in the main memory a page fault occurs.
- Demand paging uses memory more efficiently.
- In demand paging, there is no limit on the degree of multiprogramming.
- Effective Access time calculated as

  (1 – p) × ma + pft

  where p = page fault

  ma = memory access time

  pft = p × (page fault overhead + swap page out + swap page in + restart overhead)

- EAT is directly proportional to the page fault rate.
- Demand paging can have a significant effect on the performance of a computer system.
- Page replacement algorithms are:
  - First In First Out (FIFO)
  - Optimal Page Replacement Algorithm
  - Least Recently Used (LRU)
  - Clock Algorithm (Second Chance Algorithm)
  - Counting Based Algorithms

  Least Frequently Used (LFU)

  Most Frequently Used (MFU)

- FIFO replaces the page at FRONT of Queue.
- Optimal it replaces the page that will not be used for the longest time.
- LRU replaces the page that has not been used for the longest time.
- LFU replaces the page, which has the smallest count.
- MFU replaces the page, which has the highest count.
- Belady anomaly state that page fault rate may increase as the number of allocated frames increases,
- FIFO replacement algorithm suffers from Belady anomaly.
- The degree of multiprogramming increases more which degrade the performance of the system is known as thrashing.

- Thrashing degrades the performance of a system.

PRACTICE EXERCISES

1. What is memory management?
2. How Operating system helps in memory management?
3. What are the different types of memory management techniques?
4. Explain the following terms:

   (i) Internal fragmentation

   (ii) External fragmentation.

   (iii) Compaction
5. Explain the strategies of the first fit, best fit, and worst fit in the multiple partition memory management techniques?
6. What is the basic difference between Single partition and Multiple partition memory management techniques?
7. What is Paging?
8. Why is page size always power by 2?
9. Explain the technique of Segmentation in memory management?
10. Why paging is used?
11. Describe the following:

    — Swapping

    — Structure of Page Table

    — First Fit

    — Best Fit

    — Worst Fit

    — Address Binding

    — Production and Sharing

12 Write a note on virtual memory.

13. Explain the demand paging.
14. What do you mean by page fault?

15. Write steps handling the page fault.

16. Explain the performance of demand paging.

17. Why we need the page replacement algorithm.

18. Describe the working of various page replacement algorithms.

19. What do you mean by reference string?

20. What is the Belady anomaly?

21. Write a short note on Thrashing.

22. Consider the following page reference string:

    1, 2, 3, 4, 7, 5, 1, 2, 1, 7, 5, 3, 2, 4, 2, 3, 1, 6, 4, 2, 1, 2, 3, 6

    How many page faults will occur for the following page replacement algorithms?

    Assume a set of four page-frames (initially all empty).

    FIFO

    LRU

    Optimal

M.Sc. (Computer Science)
OPERATING SYSTEM

UNIT IV : FILE SYSTEM

STRUCTURE

OBJECTIVE

- Understanding Concept of File, different methods to access files.
- Using File system structure and their implementation,
- Concept of Directory and its allocation methods

FILE CONCEPT

A file is a group of similar records which is stored in memory. The file is treated as one unit by users and applications. It may be mentioned by name. The filename should be sole which means in the same location file's name should be unique. It may be created, deleted, appended, truncated. There should be file manager which provides a protection mechanism to allow machine user to administrator how processes executing on behalf of different users can access the information in a file.

File System

The file system in the operating system is assigned with the work of storing, controlling, and managing data that is stored on disks or secondary storage in the form of files. File system management is responsible for maintaining consistency in data when multiple users access files concurrently. It also provides measures for file protection at times when the system crashes.

Introduction

Files are stored permanently on secondary storage devices, such as hard disks. A file system is a part of the OS that is responsible for controlling secondary storage space. It hides device-specific complexities and provides a uniform logical view of users. A function of file system includes:

- It allows users to provide a facility to give the name of the file as user-defined names, to create, append, truncate, and delete files.
- It maps user-defined names with low-level spotters, names in a machine-understandable format so that the machine finds a file uniquely.
- It provides a uniform logical view of data to users rather than a physical view i.e. internal structure of files in which they are stored on the disk, by giving user-friendly interface.
- It controls transferring of data blocks between secondary storage and main memory and also between different files.

- It offers semantics or the file-sharing rules and regulations between numerous processes and users.
- It also allocates and manages space for files on secondary storage devices, such as disks or magnetic tapes. Space management is an important part of the file system.
- It protects the files from system failures and applies measures for recovery and backup.
- It provides security measures for confidential data such as electronic funds or criminal records.
- It also provides encryption and decryption facilities to users. Encryption is a mechanism of converting data into some code form that is unreadable to everyone except the recipient. Decryption is the reverse process of encryption.

A file system is implemented as a layer of OS and is placed between the kernel and the memory manager. It consists of utility programs that run as constrained applications that are used to control access to files. Users may access files with the help of the file system only.

File Attributes and Naming

The file is the named collection i.e. each unit is having a name, that is unique and the end-user identifies a file by its name, usually considered as a linear array i.e. sequentially arranged, of date and records. Almost all input/output operations are performed through files. All inputs are done via files and all outputs are recorded in files. Data or information stored in files is of many types. A file has a certain defined structure according to its types such as executable programs, numeric and textual data, pictures, images, and sound recordings. Each file is saved in a directory which is acts as a container for that file. Access to file is done through these directories. A user assigns a name with each file through which it can refer. Moreover, the file system requires more information about a file that is attached with the file in the form of attributes to maintain consistency and reliability of data and control multiple accesses.

File Attributes

File attributes are required by any file system to manage or maintain a file. It may differ from one operating system to another operating system. File attributes are the information about a file that is associated with every file. Some attributes are unique for each file in disks such as file locations, name and creation, date, and time. Few attributes are accessible for users, such as access privileges, name, or size of a file, whereas some of them are specifically assigned to a file for file system usage.

File attributes vary from one OS to another, but few of them are needed by every OS. The major types of attributes are:-

(1) User-Defined attributes

(2) System-Defined Attributes

User-Defined attributes

- **File name:** An identifier chosen by the user to address the file. Usually a string of alphanumeric characters, some OS allows the use of special characters, such as #, *, or $, in file names, and must be unique in its file directory.

File type: Type of information stored i.e. binary file, text file, picture file, or program file.

Owner: Name of the creator of files that controls and provides access privileges, such as read-only or read-write, to other users.

Permitted privileges: It contains information that determines read privileges, write privileges and execute privileges.

System-Defined Attributes

- **Low-level identifier:** Machine understandable names, usually in binary digits that are used by hardware to identify a file by mapping it with a user-defined name. This attribute also consists of numbers.
- **Location:** Address of sector or area where the file is stored on the disk. Usually, this information is used as a pointer; a value used by programs to find the location of a certain sector or file, to the location and consists of numbers.
- **File size:** Current file size in bytes, or words, or in blocks.
- **Creation date:** This contains the date when the file was created.
- **Allocated size:** Total allocated size of a file by the file system.
- **Volume:** Indicates which type of device is used for storing files.
- **Date of last modification:** Contains date and time of last update, insertion, and deletion in a file.

All the attributes of a file are stored in a directory where the file resides. Their storage takes more than 1 Kb space per file. Some attributes are stored in the header record associates with each file.

Naming

The most significant attribute of a file is its name that is given by the user who created it. Users should associate a name with every file to uniquely identify it and access files through these names. The file's name should be unique in its directory. In a shared system, it is recommended that the user must assign a unique name to a file.

Files are accessed by giving a complete path or address i.e., you have to specify the names of directory and subdirectories with the filename, all when combined makes a complete address for a file. It is possible, that a directory is also stored in another directory that is contained in the root directory, a directory at the top level. Therefore, a file in a system can be located by a giving-by-giving complete path from the root directory to the files, specifying all the intermediates directories, to the file system. All the directories, other than the root directory, are the subdirectories of the root directory. The root directory is denoted by '\'.

An example of the pathname for a file:

C: \ Rohit \ rsts \ os.doc

The pathname of a file is also called the complete address for a file. Slash is used to restricting names in a sequence and also indicate subdirectories of a directory. Two or more files are allowed to have similar names provided they have different pathnames i.e. their parent directory should be different.

The process that calls a file is associated with it a current directory also called the working directory. In this case, a complete pathname is not required rather; a user can access files giving pathname starting from this working directory.

File Operations

Basic operations on files are:

Create a File: For creating a file, address space in the file system is required. After creating a file, the entry of that file must be entered in the directory. Then directory entry contains the file name and location of that file which is stored in the file system.

Writing a File: A system call is used for writing into a file. It requires two parameters one file name and the second information which is to be written.

Reading a File: A system call is called to read a file. It requires two parameters: one the file name and its memory address.

Delete a File: System will search the directory with the file to be deleted. It releases all file space that can be reused by another file.

Truncating File: The user may need to remove the contents of the file but want to keep its attributes. It recreates a file.

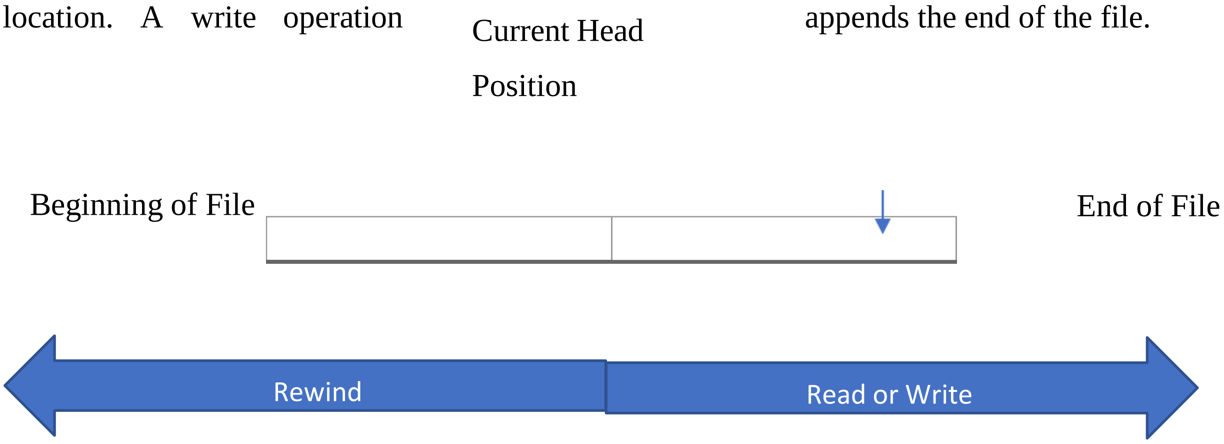
ACCESS METHODS

In the file, information can be accessed in various ways. Different types of file access methods are:

1. Sequential access
2. Direct access
3. Indexed Access

1. Sequential Access

It is a very simple method among the other methods. In the file, information is serially accessed which means one record after other records. For example: in a tape recorder, the tape plays in sequential mode. A read process reads the file's next portion and automatically advances a file pointer, that keeps tracks of Input/Output location. A write operation appends the end of the file.



2. Direct Access

It allows random access to any block. This model is based on a disk model of a file. It allows programs to read and write records in any order.

3. Indexed Access

The records in a logical sequence according to a key contained in each record. The system maintains an index containing the physical address of certain records. By using the key attributes, the records can be indexed directly.

Sequential and direct access method both are not supported by all operating system. Some operating system uses sequential access method, some operating system uses direct access method and some operating system used both access methods.

DIRECTORY AND DISK STRUCTURE

Directories are considered as symbolic tables of files that are store all the related information about the file it holds, with the content. This information includes file attributes, location, type, and access privileges. Directories are also known as containers of files.

The directory is itself a file that is owned by the OS. Millions of files present in the system need to be managed. Directories provide means to organize files in a structure. Each entry in a directory contains information about a file. Similar to files, operations such as insertion, deletion, and searching can be performed on it. Operations performed on these entries are:

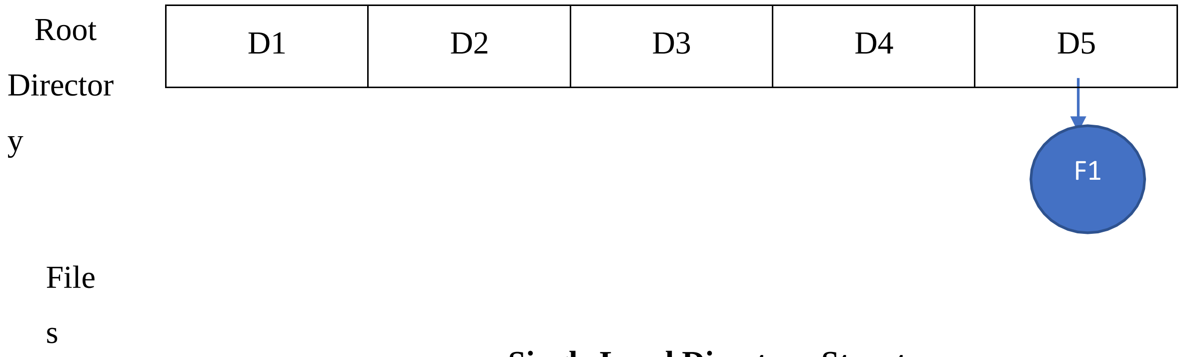
- **Searching a file:** Whenever a file is referenced, the directory must search for the related entry.
- **Create a file:** An entry of every newly created file needs to be added to the directory.
- **Delete a file:** Whenever a file is deleted, the related entry should be removed from the directory.
- **List directory:** A list of files in a directory should be shown whenever a user requests it.
- **Rename a file:** The name should be changeable when the use of the file changed or its location changes.
- **Update directory:** Whenever a file attributes changes, its corresponding entry

needs to be updated.

Based on these entries and their operations, the structure for directories can be organized in different ways. The three most common structures for organizing directory are, single-level, two-level, and hierarchical structures.

Single-Level Structure

It is the simplest form of directory structure having only one level of directories. The entire files are contained in the same directory. It appears like the list of files or sequential files having file names serving as the key. The logical structure of a single directory is given in the figure below:



Single Level Directory Structure

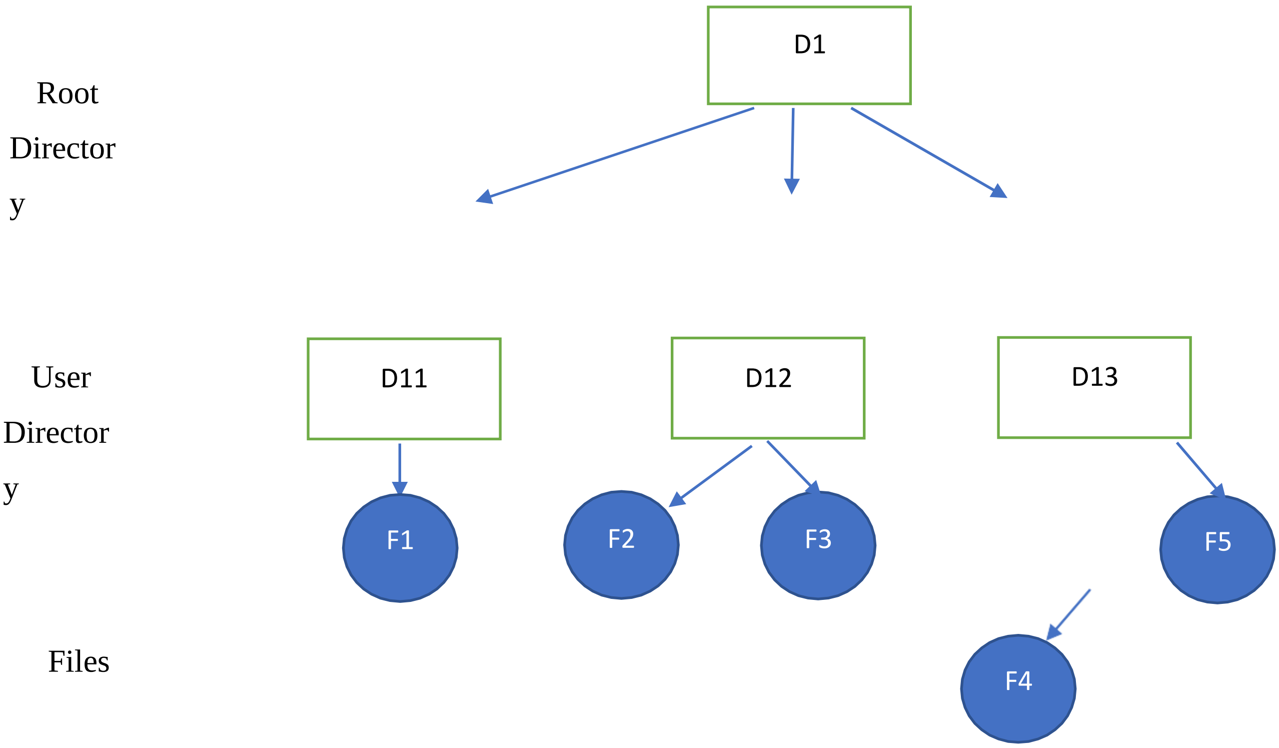
The directory structure is implemented in an earlier single-user system. It becomes outdated and inadequate in the multiple-user system. Even for a single user, it is difficult to keep track of the files if the number of files increases. Moreover, files are of different types, such as graphic files, text files, and executable files, and if the user wants to arrange these files in an organized manner such as group files by type, this structure becomes inconvenient.

Files in the single-level directory should have unique names because they are contained in one single directory. In a shared system, unique naming becomes a serious problem. These drawbacks lead us to design another structure of directories named a two-level structure.

Two-Level Structure

As the name suggests, this structure is divided into two levels of directories i.e. a master directory and user, and all these directories are contained and indexed in the master directory. The user directory represents a simple list of files of that user.

The two-level structure looks like an inverted tree of height 2. The root of this tree is the master directory having user directories as its branches. Files are the leaves of these branches. The logical structure of two-level directories is shown in the figure below:



Two Level Directory Structure

A user name and file name are the pathnames for a file. This structure solves the problem of unique names up to a certain extent i.e. user can assign duplicate names to files provided files are present in different directories. Names need to be unique in the user’s

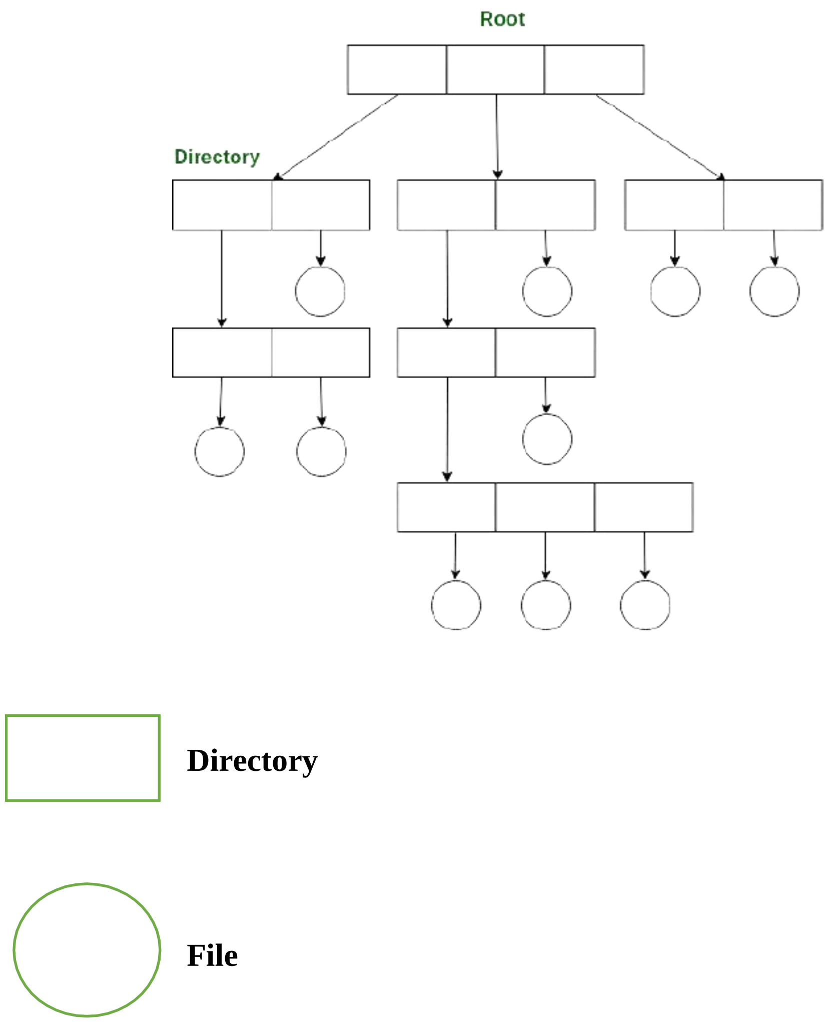
directory because two file names cannot be the same in one single directory. User searches a file in his directory only, which allows different users to have files with the same names.

To create or delete a file, OS searches only the user’s directory that initiates the command. OS uses a special system program or system calls, to create or delete a user directory. This system program creates or deletes a user directory entry from the master directory.

This structure provides no help in grouping files of different types. In a shared system, one user wants to access another user’s file because files are shared in a shared system network, which again creates the problem of uniqueness in file names. The user has to give a complete pathname to name a file in another user's directory.

Hierarchical Structure or Tree-Structured Directories

This is the most powerful and flexible structure and is implemented in almost every OS. The two-level structure is extended into a more advanced hierarchical structure of arbitrary levels. It uses the same concept of a two-level structure of master directory having user-directories as subdirectories. In addition, each user directory, in turn, has subdirectories and files as its branches and leaves. A typical hierarchical structure of directories and files is shown in the figure below.



Tree Level Directory Structure

Users can create their subdirectories to organize files of different types, such as separate subdirectories for graphic files, or separate subdirectories for text files. Special system calls are used to create or delete directories. Internal formats, i.e. the internal structure in which the details of the directory are stored, of each directory ha

s an entry that stores special bits representing a subdirectory or a file i.e. 0 bit represents a file and 1 bit represents a directory.

The user always works on the files in the current directory. The current directory holds all the files, which a user currently requires. OS searches the current directory for reference to a file. In a hierarchical structure, the user can access a file, which is not in the current directory, giving pathname. Users can change the current directory also through system calls.

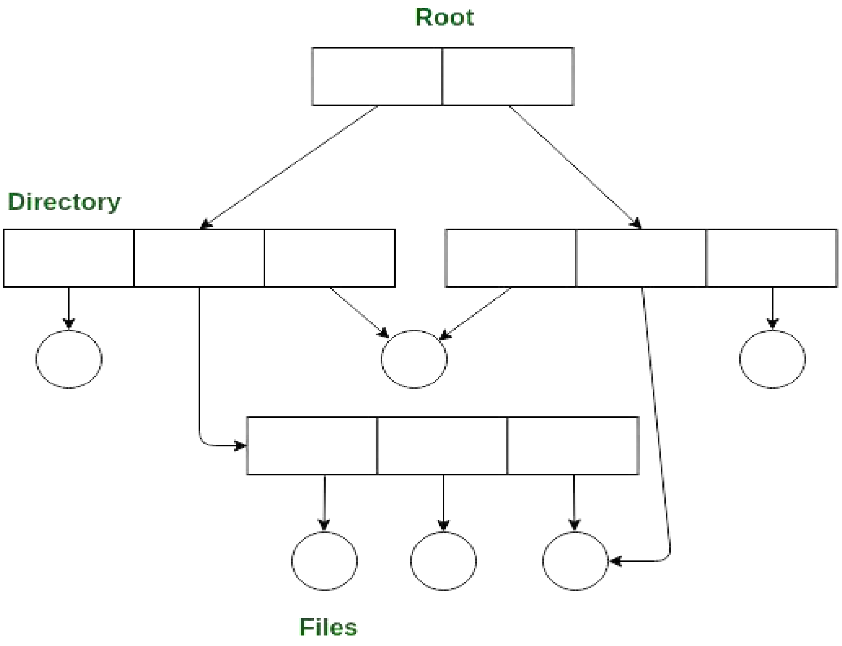
In a hierarchical structure, a file can be referenced in two ways, *absolute pathname,* and *relative pathname.*

Absolute pathname starts from the root and ends at the required file following a path of directories and subdirectories.

Relative pathname starts from the current directory to the file. Users can access another user file by giving its pathname. In a hierarchical structure, the pathname to a file can be longer than a two-level directory. This increases the search time for a file that resides in other user directories.

Acyclic Graph Directories

It always directories to shared subdirectories and files. Some files or directories may be in two different directories. Shared files and directories can be implemented by using links. Link is implemented as an absolute path or relative pathname. It is more flexible than a simple tree structure but sometimes it is more complain.



The acyclic graph directory structure

Characteristics

1. Files and sub-directories can be shared.
2. It is more flexible than three structures.
3. Deletion of a shared file is difficult.
4. Consistency should be taken care of.

General Graph Directories

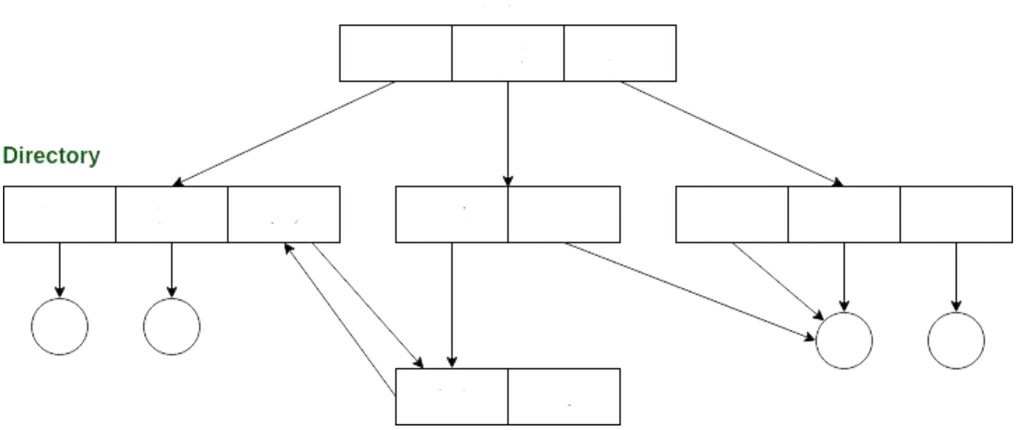
In this structure, cycles are also permissible within a directory structure are from more than one parent directory, multiple directories can be made. The main disadvantage of the general graph directory structure is to compute the total size or memory required which has been taken by the files and directories.

Characteristics

It permits more cycles.

It is more flexible as compared to the other directories structure.
It is more expensive than others directory structures.

It requires garbage collection.



A General Graph Directory structure

FILE-SYSTEM STRUCTURE

The file system resembles a complete directory structure that includes a root directory following subdirectories and files under it. The file system is required to enhance the function of retrieval and storage of files. We can mount more than one file system that creates a directory structure showing an image of a single file system. Commands used for file system operations are:

Chfs: Change the file system characteristics

Crfs: Add a new file system

Rmfs: Remove a file system

Mount: Make files system available for user access.

In the real world, different Operating systems use different file systems. Each file system has a different directory structure and different ways to represent files.

1. CD ROM

CD ROM is an acronym for Compact Disc-Read Only Memory. Data is stored on one side of it that is coated with the highly reflected material. Data is stored in the form of series of microscopic pits, i.e. data is stored in binary digits of 0 and 1 and a pit represents a binary 1, which is read by a low-powered laser in a drive unit. To read data, the laser rotates the disk at various speeds at a Constant Linear Velocity (CLV).

CD ROM is used for storing audio in digital format and is of size 6547.4 MB. CD ROM can store data equivalent to 500 floppy disks. Data on CD ROM can be accessed much faster than any other tape. Accessing speed of CD ROM is 75-150 sectors per second. To read a CD ROM, we need a CD ROM reader and an appropriate software driver that controls the functionality of CD ROM.

Data is stored in the form of sectors with error-correction codes i.e. codes that are used to debug any error if occurred in the stored data. Data on CD ROM can be stored and retrieved, as files for that we need a file system to manage the stored data. The file system allows you to access the contents of CD ROM using the interface of a normal file system i.e. the file system stored on the disk.

2. FAT16

To understand the FAT 16 file system, a discussion on File Allocation Table (FAT) is necessary. FAT is a table that has entries in the form of pointers. These pointers point to the files that are present in a partition. Partition is a logical division of the table. Each entry in this table consists of a block number, a 0 valued block number represents as unused block any new entry is stored on it. Partition of the disk depends on the file system used. The two most common file systems are FAT 16 and FAT 32.

FAT 16 is the old MS-DOS file system created by Microsoft in 1977 that used 16 bits table. FAT 16 support filenames of 8.3 standard i.e. 8 characters for the name and 3 characters for file extension.

The value stored in the FAT table is 16 bits. A 16-bit file allocation table means that the size of the address stored in the entry of FAT is limited to 16 bits. This implies that you can't have storage allocation units i.e. the entries in the FAT 16 table are restricted to 65536 units. This is the main drawback of FAT 16.

Another drawback of FAT 16 is that its partition is of fixed size i.e. of 32k. That wastes a lot of disk space in storing files of small size such as pictures, HTML documents, etc.

3. FAT32

It is an advancement of FAT 16 that supports 32 bits addresses. Entries in FAT 32 tables can hold a value up to 32 bits. It divides the hard disk into smaller partitions of a small disk area than FAT 16 so that disk space is used more efficiently with increased storage capacity.

FAT 32 allows you to create a partition up to 2048 GB size. The number of clusters in FAT 32 ranges around 4 billion i.e. 4,294,967,296 inexact. FAT 32 supports long file names of a maximum of up to 255 characters. File names are not case sensitive i.e. no matter whether the names are in capital letters or small letters; it saves the case of filenames once created. Minimum disk space needed for FAT 32 is 8 GB. Os that supports FAT 32 are: OSR2, Windows 98, Windows 2000, and Linux.

4. Unix

Unix is a multi-user and multitasking OS that allows multiple users to access multiple files concurrently. It is a data structure that resides on disks. Unix file system has a tree-like structure having a root directory (/) at the top node. This hierarchical structure appears to users as a single file system having all types of files in one tree, the concept of drives such as A, C: etc is not found here. Each node, a vertex of the UNIX file system tree, represents a separate file system.

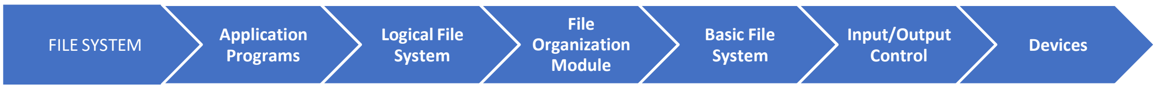
Thus, the separate device is mounted in one single file system. A file in UNIX is a collection of randomly addressable bytes, not sequentially; each byte has a different address, not in the continuity of a previously-stored byte. Files are allocated on the block randomly on the disk. The index method is used to keep track of each file. A filename can extent up to 255 characters excluding forward-slash (/) & Null and are case sensitive. File in Unix may contain holes i.e. no data blocks. The directory is also a file except the user can’t write on it. Unix associates its file with three access permissions: read, write, and executable. Users can access files giving absolute and relative pathname.

UNIX file system is made up of four parts:

- **Boot block:** Contains boot program and load kernel into memory when a user boots the system or switch on the system.
- **Superblock:** Contains all information about the attributes of a file.
- **I-list:** Contain a list of I-nodes of the file system. I-node is a data structure that holds information about an individual file, such as file size, access privileges, or file name. Each file is related to I-node and is identified by I-nodes number.
- **Data block:** Contains actual file contents.

FILE-SYSTEM IMPLEMENTATION

File system structured in the layers which are: Application Programs, Logical File System, File Organization Module, Basic File System, Input/Output Control, Devices



- **Input/Output Control level –**

  Device drivers perform as an interface between devices and Operating Systems, these drivers help to transfer data from disk to main memory and vice versa. It picks block numbers as input and as output, it returns low-level hardware-specific instruction.

- **Basic file system –**

  It refers to normal commands to the particular device driver to do operations: read and write on particular physical blocks on the disk. A block is part of memory in the buffer that can contain the contents of the disk block and cache stores frequently used file system metadata. It also handles the memory buffers and caches.

- **File organization Module –**

  It contains information about the files such as their location, and their logical blocks, and physical blocks. Physical blocks were entirely different so they do not match with logical numbers of the logical block which are numbered from 0 to N. It contains some free space that is tracked by unallocated blocks.
- **Logical file system –**

  It manages metadata information about a file i.e includes all details about a file except the actual contents of the file. It also keeps with the help of File Control Blocks. File Control Block (FCB) contains all the information about a file such as the owner of the file, size of the file, permissions granted, file location, and its contents.

Advantages :

1. Replication of code is lessened.
2. Each file system can have its independent logical file system.

Disadvantages :

If someone accesses many files simultaneously then its outcomes will reduce the performance of the system.

We can **implement** a file system by using data structures which are:

1. **On-disk Structures –**

   Generally, it contains the following information:

   a. total number of disk blocks available

   b. free disk blocks available

   c. location of them.

Below given are different on-disk structures :

1. **Boot Control Block –**

   It is generally the 1st block of volume and it has information that is needed to boot any operating system. In UNIX, this block is known as a boot block and in NTFS it is known as a partition boot sector.

2. **Volume Control Block –**

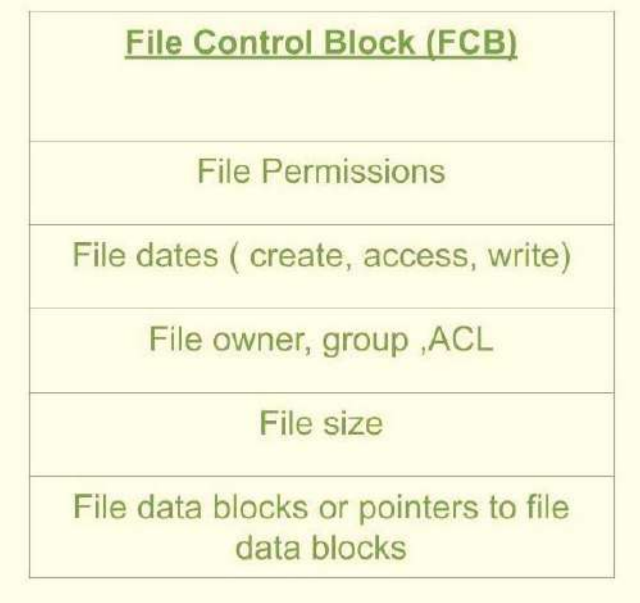
   It contains information about a particular partition for example:- count of free blocks, size of blocks, and its pointers also. In UNIX it is known as superblock and in NTFS it is stored in the master file table.

3. **Directory Structure –**

   They kept file names and linked inode numbers. In UNIX, it contains file names and associated file names and in NTFS, it is stored in the master file table.

4. **Per-File FCB –**

   It holds details about files and it has an exclusive identifier number to allow association with the directory entry. In NTFS it is kept in the master file table.



File Control Block (FCB)
File Permissions
File dates (create, access, write)
File owner, group ,ACL
File size
File data blocks or pointers to file data blocks

Structure of File Control Block (FCB)

In-Memory Structure :

They are maintained in the main memory and these are helpful for file system management for caching. Several in-memory structures given below:

1. **Mount Table:** It is a table in which information about every mounted volume is stored.
2. **Directory-Structure cache –** It contains the directory information about which directory is freshly accessed.
3. **System-wide open-file table** –It holds the copy of the File Control Block (FCB) of every open file.
4. **Per-process open-file table –** It holds information on which process opened the file and it also maps with appropriate system-wide open-file.

DIRECTORY IMPLEMENTATION

A directory can be implemented in two ways:

1. Linear List
2. Hash Table

Linear List

The linear list is the simplest method. In this linear list of filenames used. For searching purposes, a linear search technique is used to find a particular entry. It is very simple but time-consuming to execute. Directory information is used often. Users also notice the slow implementation of access to it.

Hash Table

With the hash table, it decreases the directory search time. Insertion and deletion are very simple. Hash table takes the value which is compiled from the file name and then it returns a pointer to a file name in the form of a linear list. It always uses a fixed size.

ALLOCATION METHODS

The space allocation strategy is often closely related to the efficiency of file accessing and of logical to physical mapping of disk addresses. Three major methods of allocating disk space are widely used which are:

Contiguous Allocation Method

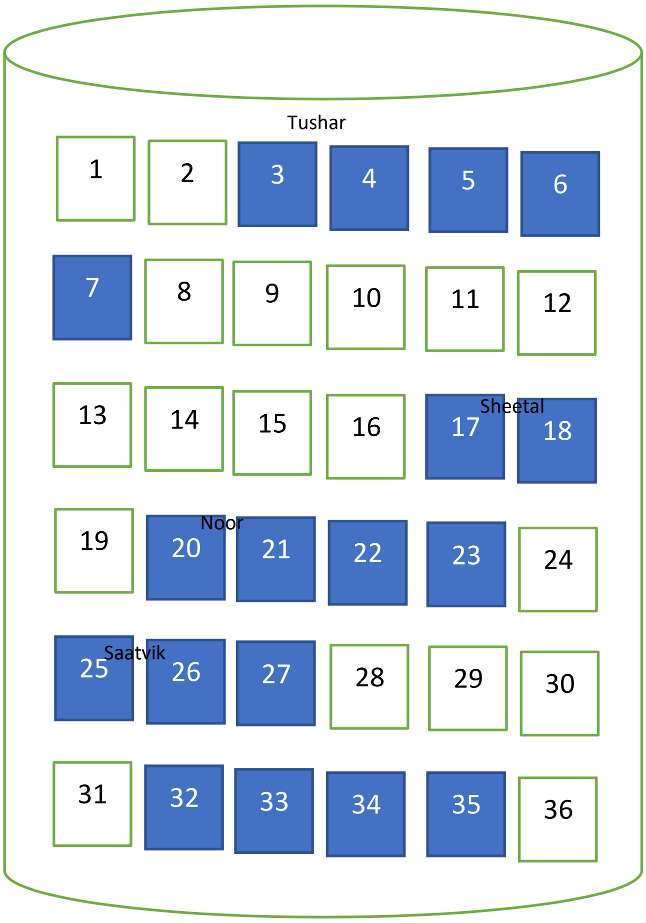
The contiguous allocation method always requires every file to occupy a set of adjoining addresses or blocks on the disk. Disk addresses describe in a linear order on the disk. The important thing to be noted is that, in this ordering, the block of memory represents as B, accessing block where is B+1 after block B normally requires no head movement. When head movement is needed, it is only one track. Thus, the number of disks seeks required for accessing contiguous allocated files is minimal.

Contiguous allocation of a file is defined by the disk address or blocks and the length of the first block. Suppose, the file has a size of N blocks, and its starting location is location B, then it must occupy consecutive blocks namely B, B+1, B+2,….., B+n-1. The directory entry must contain for each file indicates the address of the starting block of the file and the total length of the file in terms of blocks. This is the simplest method of allocation. Performance is good because the entire file can be read from the disk in a single operation.

The difficulty with a contiguous allocation is finding space for a new file. If the file to be created is n blocks long, then the OS must search for n free contiguous blocks. The most common approaches used to select a free hole among the set of available holes are First-fit, best-fit, and worst fit. Both first-it and best-fit are a better option as compare to worst fit in terms of both criteria which is time and storage utilization. Neither first-fit nor best fit is best in terms of storage utilization, but first-fit is generally faster.

These algorithms also have a disadvantage which is external fragmentation. As files are allocated and also deleted, the free disk space is broken into tiny chunks. External fragmentation exists when enough total disk space exists to satisfy a request, but this space not contiguous; storage is fragmented into a larger number of small holes.

Another problem with a contiguous allocation is determining how much disk space is needed for a file. At the file creation time, how much space it requires must be known and allocated. One question arises here, how does the creator know about the size of the file which is to be created? In some cases, this determination may be fairly simple, but generally, the estimation of the size of an output file is a little difficult. A directory containing the file information is maintained. Starting block address and a total number of blocks in a file are read from this directory to read/write a file.



Directory		
File	Start	Length
Tushar	3	5
Saatvik	25	3
Noor	20	4
Sheetal	17	2
Rohit	32	4

■ USED

□ FREE

Contiguous File Allocation

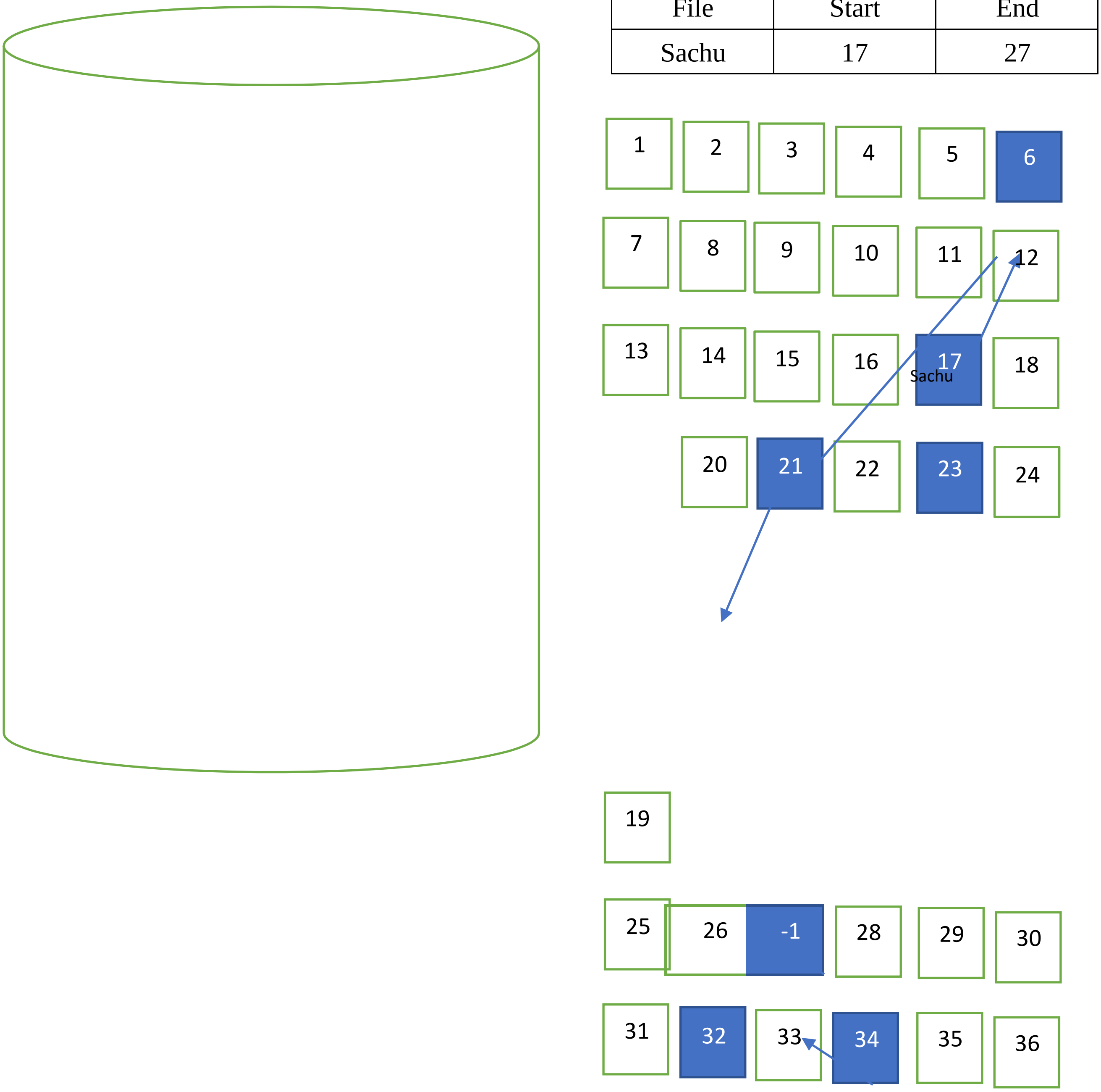
Characteristics

1. It supports flexible size portions.
2. It requires Pre-allocation.
3. It needs only a single entry of a file.
4. Allocation frequency is only once.

Linked List Allocation method

The problems in contiguous allocation can be traced directly to the requirement that the spaces be allocated contiguously and that the files that need these spaces are of different sizes. These requirements can be avoided by using linked allocation.

Directory		
File	Start	End
Sachu	17	27



Link List Allocation

In linked allocation, every file is fragmented into Blocks and makes a linked list of these disk blocks. The directory stores a pointer to the first Block of the file. For example, a file of

7 blocks starts at block 17, they might be next block 6, then block 21, block 32, block 23, block 34, and finally last block 27. Each block holds a pointer to the next block and the last block contains a NULL pointer which shows that end of the link list. The value -1 may be used for NULL to differentiate it from block 0.

In the linked allocation method, every directory entry contains a pointer to the file's first disk block. Initially, the pointer is initialized with a null value which shows that it is an empty file. A write to a file removes the first free block from the free block list and writes to that block. Then the address of this new block is associated with the last block of the file and then linked to this block at the end of the file. To read a file, the pointers are used by a followed pointer from block to block. The problem of the **Contiguous File** allocation method i.e., external fragmentation is resolved in the linked allocation. Any free block can be utilized to fulfill a request. Notice also that there is no need to declare the size of a file when that file is created. A file will still grow as possible as there are free blocks.

The linked allocation has some disadvantages. The foremost problem is that it is inefficient to support direct access; it is effective only for sequential-access files. To locate the i^{th} block of a file, it should begin at the start of that file and follow the pointers till the i^{th} block is reached. Note that every access to a pointer needs a disk read.

Another severe problem is reliability. A bug in OS or disk hardware failure might result in the pointer being lost and damaged. The effect of which could be picking up a wrong pointer and linking it to a free block or into another file.

Characteristics

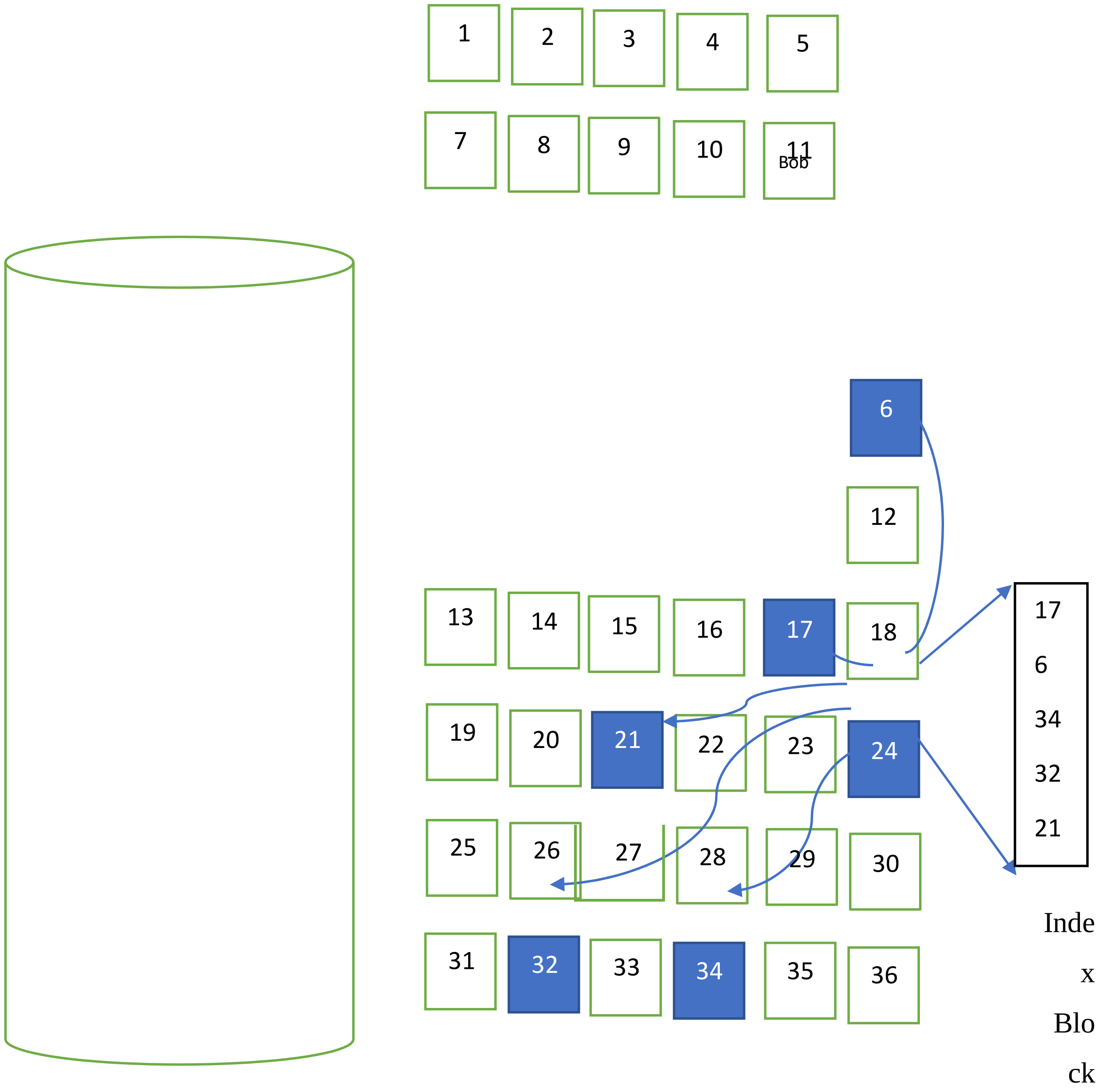
1. For a File there is one entry in the File allocation table size.
2. It's supports fixed size portions.
3. In this allocation Pre-allocation is feasible.
4. Allocation occurrence is Low to High.

Indexed Allocation Method

The problem of both contiguous and linked allocation is resolved in the indexed allocation method. In this allocation, one block is used as an Index block in which all the pointers bring together into one location.

Of course, the index block will occupy some space and thus could be considered as an overhead of the method. In indexed allocation, every file has its induvial index block, which is nothing an array of disk sectors of addresses. The jth sector of the file is pointed by the jth entry in the index block. The directory keeps the address of the index block of a file only. To read the jth sector of the file, from the index block, the pointer in the jth entry is read to locate the desires sector. Indexed allocation supports direct access, it resolves the external fragmentation. Any free block anyplace on the disk might satisfy a request for extra space.

Directory	
File	Index Block
Bob	24



Indexed Allocation Method

Characteristics

1. It supports both sequential and direct access.

2. There is no external fragmentation.

3. It does suffer wasted space.

4. Manage the Pointer is overhead.

5. It is faster than the other two approaches.

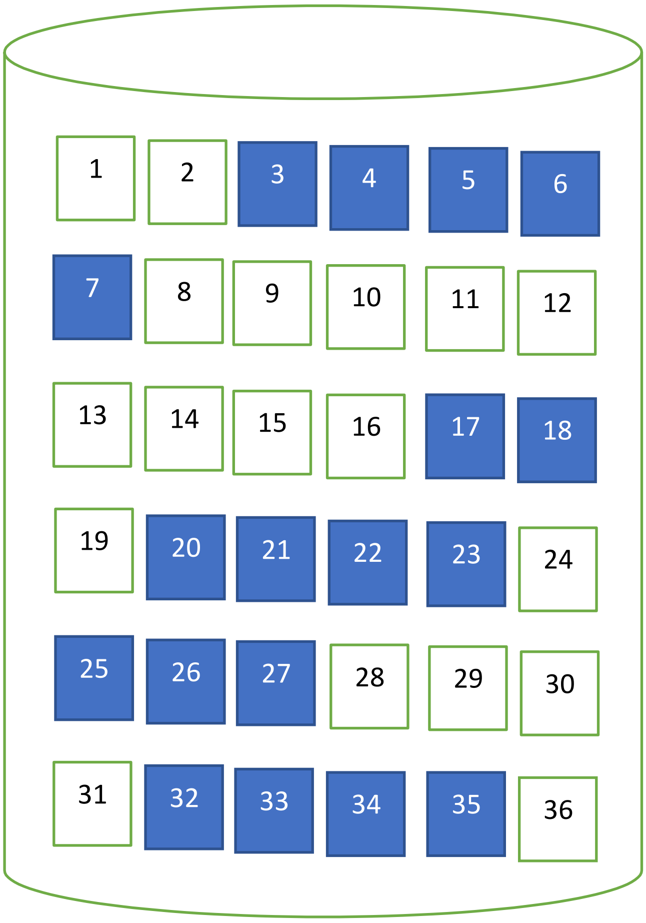
Free-Space Management

Since there is only a restricted amount of disk space, it is necessary to utilize the space from deleted files for new files. To keep track of free disk space, a free-space list is maintained by the system. This list keeps records of all disk blocks which are free. Whenever a file is created, the free space list has to be searched for the required amount of space and allocate space to a newly created file. After that from the free-space list that space is removed. Whenever a file is deleted then the free space list is updated by deleted file space added to the free-space list.

Bitmaps or Bit – Vector

In the bitmap method a table, called bit map is maintained for keeping track of the information about which part of memory is allocated to which process and which portion of memory is free. The bitmap is dynamic. This means the table is updated each time when a new process is allocated memory space, an existing process is swapped out of the main memory, a process is swapped in the main memory, or a process is completed and released from memory.

In the bitmap method, the memory is divided into certain equal size units or blocks. While allocating memory space to a program an OS can only allocate a complete unit of memory space and not a part of the unit. There is a cell in the bitmap for each unit of memory. If a unit is allocated to any process the value of the cell corresponding to the unit is



set to 1 otherwise if the memory unit is free, the value is set to 0.

Figure Bit Vector

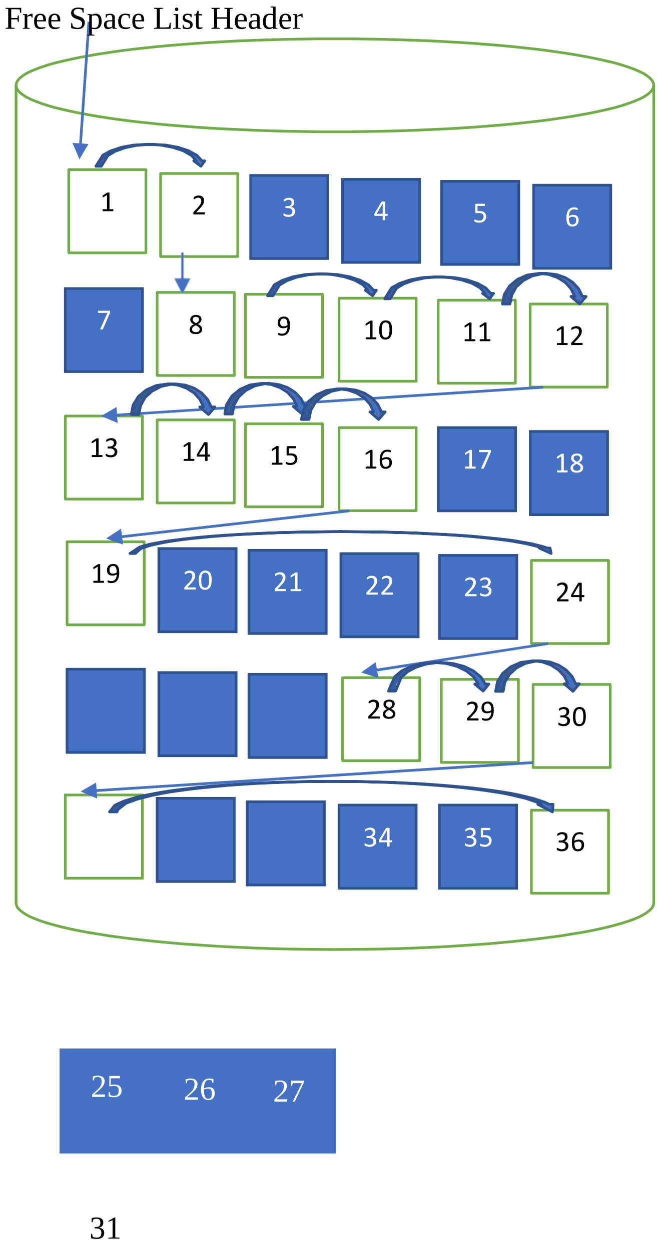
1, 2, 8, 9, 10, 11, 12,13,14,15,16,19,24,28,29,30,31,36 are flocks free so bit map of this example is 0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,1,1,0,1,1,1,1,0,1,1,1,0,0,0,0,1,1,1,1,0

There is a cell in the bit map corresponding to every memory unit. The value of the 3 cell in the bitmap is 1 because its corresponding memory unit is in use by the process. Value of the first and second cell is 0, because it's free.

An **advantage** of the bitmap method is that it is very simple to implement. On the other hand, the **disadvantage of this method is** it is very slow. OS needs much time to search for a hole, when allocating memory space to a new process or an existing process swapped in.

Linked List

In this method, a pool or chain of nodes is created to keep track of memory that is free and maintain a linked list i.e. the first block contains the address of the next free block and so on. The address of the very first free block is stored in a special node called as START node. The last block of the linked list contains NULL, which implies that there is no other free block.



Link Free Space List on Disk

For accessing the information in the linked list, it is loaded into physical memory. Whenever any allocation is to be done, blocks are removed from the head of the list and allocated. It occupies a large space. Traversing is also time-consuming. The following list shows a free block list.

Grouping

A modification of the free-list approach, in the first free block, keeps the addresses of n free blocks. There are only the first n-1 blocks that are free. The last one is the disk address of another block containing addresses of another n free block. The importance of this implementation is that a block contains the addresses of a larger number of free blocks so it can be found quickly if we need the number of free blocks.

Example: Block 1 Contains the address of Block 2. Block 2 contains the address of the first block of the next group block i.e., 8, Block 8 stores the address of 9 to 16 and the last block is 16 so it contains the address of the first block of the next group 19. As 19 is a single free block so, it contains the next group address i.e. 28. Block 28 keeps the address of 29 and 30. Block 30 contains the address of 31 and again Block 31 is the only single block in Grouping so it contains the address of 36. 36 is the last block os it stores nothing

Counting

Another method is to take benefit of the fact that, generally, many contiguous blocks may be allocated or freed contiguously, mainly when we used the contiguous allocation method. Thus, rather than keeping the entire list of free disk addresses, we noted the address of the first free block along with it the number n of free contiguous blocks which follow the first block. Each entry in the free-space list then contains the disk address along with the count. Although each entry needs more space as compared to a simple disk address, the overall list will be shorter, as long as the count is generally greater than 1.

First Free Block	Count
1	2
8	9
19	1
28	3
31	1
36	1

Points to Remember

The file system must identify and locate the selected file before operation on a file.

The file control block contains information about the file.

A disk can be divided into multiple partitions.

The file can be accessed by

— Sequential Access

— Direct Access

— Indexed Access

Three major methods of allocating disk space are widely used which are
Contiguous Allocation Method
Linked Allocation Method

— Indexed Allocation Method

The directory can be implemented in two ways:

— Linear List

— Hash Table

Free space management techniques are:

— Bit Vector

— Linked List

— Grouping

— Counting

PRACTICE EXERCISES

1. What is a File system?
2. Write a note on file attributes and naming?
3. What is File Organization?
4. Discuss the various methods of file allocation?
5. What are the disadvantages of the contiguous file allocation method?
6. What is the difference between linked allocation and indexed allocation methods?
7. What is a directory?
8. What are the different levels of the directory structure?

9. Discuss the various file systems?

10. Discuss the concept of free space management?

11. What are the various methods of Free Space Management?

M.Sc. (Computer Science)
OPERATING SYSTEM

UNIT V1: INTRODUCTION TO LINUX

STRUCTURE

OBJECTIVES

- To Understand Linux's shell, Kernel, and file system of Linux.
- Introduction to different types of directories: Parent, Subdirectory, Home directory; rules to name a directory, Important directories in Linux File System

INTRODUCTION TO LINUX

Linux which is pronounced as Lin-nucks is an operating system similar to UNIX. Some people also called it as descended form of UNIX. Linux is first released in 1991 by Linus Torvalds at the University of Helsinki. Since then it has gain huge popularity among the programmers. The Linux is an open-source operating system. It means it comes with the source code, so that one can change and customize the operating system according to the requirements. By the term open source it is meant that

- The independence to run the program in operating system for any function.
- The freedom to study any program that how it works and to alter it to make it work according to your need.
- The liberty to redistribute copies of the program to facilitate your neighbours.

- The free will to share copies of your modified versions to others for use.

Initially, Linux was created for personal computers and gradually it was used in other machines like servers, mainframe computers, supercomputers, etc. Nowadays, Linux is also used in embedded systems like routers, automation controls, televisions, digital video recorders, video game consoles, smart watches, etc. The biggest success of Linux is Android (operating system) it is based on the Linux kernel that is running on smart phones and tablets. Due to android Linux has the largest installed base of all general-purpose operating systems. Linux is generally packaged in a Linux distribution.

LINUX DISTRIBUTIONS

Linux has a numeral of dissimilar versions to ensemble any kind of user. Roughly all the editions of the Linux can be downloaded free of charge. These editions are labelled allocations (or, in the small form, –distros‖). There are additional 600 distributions of Linux**.** Some of the general distributions of the Linux are:

- LINUX MINT
- MY LINUX
- MANJARO
- DEBIAN
- UBUNTU
- DEEPIN
- ANTERGOS
- ARCH LINUX
- GENTOO LINUX

- KALI LINUX
- SOLUS
- FEDORA
- ELEMENTARY OS
- OPENSUSE

Every distribution has a dissimilar take on the desktop. A small amount of the Server distributions of the Linux comprises:

- Red Hat Enterprise Linux
- Ubuntu Server
- Centos
- SUSE Enterprise Linux

A few of the above server distributions are at no cost (such as Ubuntu Server and CentOS) and a little have an associated cost (such as Red Hat Enterprise Linux and SUSE Enterprise Linux). Those with an associated price also comprise support.

CHARACTERISTICS OF LINUX OPERATING SYSTEM

Following are a few of the significant characteristics of Linux Operating System.

- **Portable** – Portability means software be able to work on dissimilar types of hardware in similar way. Linux kernel and application programs sustain their setting up on any kind of hardware stage.
- **Open Source** – Linux source code is liberally accessible and it is community based growth project. Multiple teams carry out task in collaboration to augment the potential of Linux operating system and it is constantly evolving.
- **Multi-User** – Linux is a multiuser scheme means numerous users can access network

resources like memory/ ram/ application programs at similar time.

- **Multiprogramming** – Linux is a multiprogramming arrangement means numerous functions can run at similar time.
- **Hierarchical File System** – Linux offers a standard file arrangement in which organization files/ user files are ordered.
- **Shell** – Linux provides a particular interpreter program that can be employed to carry out commands of the operating system. It can be employed to do a variety of kinds of processes, call application programs etc.
- **Security** – Linux offers user security by means of authentication characteristics like password protection/ controlled admittance to particular files/ encryption of information.

Linux is fast, free and easy to use, power laptops and servers around the world. Linux has numerous more characteristics to amaze its user such as:

- **Live CD/USB:** Approximately all Linux distributions have Live CD/USB characteristic by which user can run/try the OS even with no mounting it on the scheme.
- **Graphical user interface (X Window System):** People think that Linux is a command line OS, wherever its precise also but not essentially, Linux have packages that can be mounted to construct the whole OS graphics based as Windows.
- **Support's** most national **or customized keyboards:** Linux is engaged worldwide and consequently obtainable in many languages, and supports on the entire of their custom national keyboards.
- **Application Support:** Linux has its own software ordnance through which users can download and set up thousands of desires just by concerning a command in Linux Terminal or Shell. Linux can3 besides run Windows demand if essential.

HISTORY OF LINUX OPERATING SYSTEM

The History of Linux commenced in 1991 with the beginning of a personal project by a Finland student Linus Torvalds to create a new open operating scheme kernel. From then, the resulting Linux kernel has been perceptible by fixed expansion during history.

- In the year 1991, Linux was started by a Finland student Linus Torvalds.
- Hewlett Packard UNIX(HP-UX) 8.0 was enlightened.
- In the year 1992, Hewlett Packard 9.0 was liberated.
- In the year 1993, NetBSD 0.8 and FreeBSD 1.0 liberated.
- In the year 1994, Red Hat Linux was initiated. Caldera was originated by Bryan Sparks and Ransom Love and NetBSD1.0 liberated.
- In the year 1995, FreeBSD 2.0 and HP UX 10.0 were liberated.
- In the year 1996, K Desktop Environment was initiated by Matthias Ettrich.
- In the year 1997, HP-UX 11.0 was liberated.
- In the year 1998, the fifth invention of SGI Unix i.e IRIX 6.5, Sun Solaris 7 operating system, and Free BSD 3.0 was liberated.
- In the year 2000, the conformity of Caldera Systems with the SCO server software division and the professional services division was declared.
- In the year 2001, Linus Torvalds liberated the Linux 2.4 edition source code.
- In the year 2001, Microsoft ordered a trademark suit besides Lindows.com
- In the year 2004, Lindows name was altered to Linspire.
- In the year 2004, the first liberate of Ubuntu was discharged.
- In the year 2005, The project, open SUSE begin a complimentary allocation from Novell's community.
- In the year 2006, Oracle liberated its own allocation of Red Hat.
- Dell initiated allocated laptops with Ubuntu pre-installed in it, in the year 2007.
- The Linux kernel 3.0 version was liberated in the year 2011.
- Google Linux-based Android declared 75% of the smart phone market share, in form of the number of phones dispatched, In the year 2013
- Ubuntu asserted 22,000,000 users in the year 2014.

ARCHITECTURE OF LINUX

Linux architecture has the subsequent components:

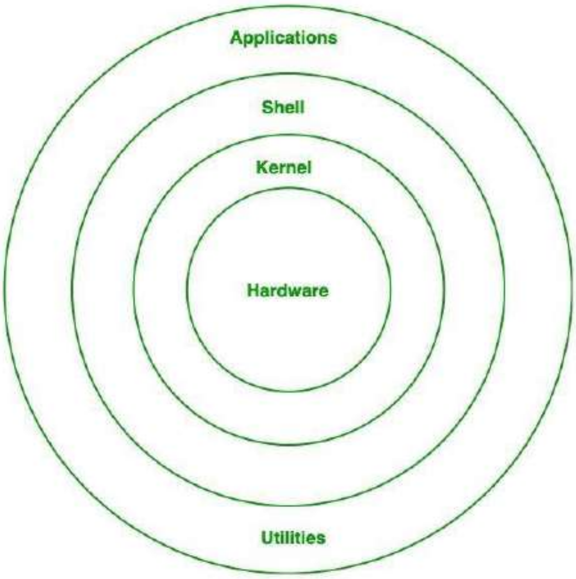


Figure 6.1: Architecture of Linux Operating System

I. **Kernel:** Kernel is the heart of the Linux supported operating system. The kernel is the mainly significant part or mind of every operating system as it handles the conversation among a machine's hardware and its software. In easy words, the kernel acts as a spirit of the network and handles the memory, peripheral apparatus, and CPU. The kernel places at the ―lowest‖ point of the OS. This creates the method seem as if it is the only procedure running on the device. The kernel too well is accountable for avoiding and justifying clashes among diverse processes. Particular kernel types are as

- Monolithic Kernel,
- Hybrid Kernels,
- Micro Kernels and
- Kernels from Exo.

A few of the key design ethics executed by Linus kernel are:

- In Linux, the whole thing is a file philosophy
- Multi-users ability

- Multitasking capability
- Portability among GPU architecture
- Modularity
- Security
- Configurability

II. **Shell:** It is an interface to the kernel which covers the difficulty of the kernel's tasks from the client. It takes instructions from the user and carry out the kernel's functions. Linux include two kinds of command keys: text mode comparable to those originated in mainly UNIX systems (example the bourne shell, the bourne over shell, the c shell, the turbo C shell and the korn shell) and graphical user interface (GUI) such as the KDE (K Desktop Environment) and GNOME (GNU Network Object Model Environment). **Terminal** is the purpose that brings it all collectively, in the sense that it offers a visual illustration of the shell for the client to penetrate commands. In other words, in a GUI (graphical user interface), where requests and other descriptions are visually symbolized by images that the client can manoeuvre by ticking on them with a cursor, a terminal request release a window where the client can type in instructions for the shell to infer into binary communication for the kernel.

III. **System Library:** It is the exceptional kind of tasks that are used to execute the functionality of the operating system. These pieces carry out the background facility of the Linux operating system similar to scheduling, printing, sound, etc that each start-up during boot of the operating scheme, or after you register into your system.

IV. **Hardware Layer:** This layer comprises of all the peripheral apparatus like RAM/ HDD/ CPU etc. Some of the substantial hardware that is incorporated in any tool comprises of mouse, keyboard, the graphics chipset, display, and, but you do have one, your system interface card

V. **System Utility:** It offers the functionalities of an operating system to the client. A Linux-based scheme will typically appear with such a set of established Unix-like utility services; these were all normally basic tasks used in day-to-day operating system procedures, and also precise devices and requests. This is frequently software that was available and liberated under the open-source license by the GNU Project, consequently the software can be explicitly downloaded, restructured, and reallocated to all.

Advantages of Linux

The main benefit of Linux, is it is an open-source operating system. This means the source code is simply obtainable for everybody and you are authorized to give, alter and allocate the code to anybody without any consent.

- In terms of security, Linux is further secure than any other operating scheme. It does not signify that Linux is 100 percent safe it has a little malware for it but is fewer susceptible than any other operating scheme. So, it does not necessitate any anti-virus software.
- The software renewed in Linux are simple and recurrent**.**
- A variety of Linux distributions are accessible so that you be able to utilize them according to your desires or according to your flavour.
- Linux is liberally available to employ on the internet.
- It has huge community support.
- It offers high constancy. It seldom slows down or freezes and there is no necessitating rebooting it after a small time.
- It preserves the privacy of the client.
- The presentation of the Linux scheme is much elevated than other operating scheme. It permits a large quantity of people to employee at the similar time and it grips them resourcefully.
- It is network pleasant.
- The flexibility of Linux is elevated. There is no necessitate to establish a complete Linux suit; you are authorized to establish only required apparatus.
- Linux is companionable with a huge number of file set-ups.
- It is quick and simple to fit from the web. It can also mount on any hardware even on your previous computer structure.
- It carry out all tasks correctly even if it has partial liberty on the hard disk.

Disadvantages of Linux

- It is not extremely user-friendly. So, it might be puzzling for beginners.
- It has tiny marginal hardware drivers as contrast to windows.

Widows Vs Linux

Few Differences between Linux and Windows

S.NO	Linux	Windows
1.	Linux is a open source operating scheme.	While windows are the not the open source operating scheme.
2.	Linux is at no cost.	While it is expensive.
S.NO	**Linux**	**Windows**
3.	It's file name case-sensitive.	While it's file name is case-insensitive.
4.	In linux, monolithic kernel is employed.	While in this, micro kernel is employed.
5.	Linux is more competent in comparison of windows.	While windows are fewer competent.
6.	There is forward slash is employed for separating the directories.	While there is back slash is employed for isolating the directories.
7.	Linux offers more protection than windows.	While it offers less safety than linux.
8.	Linux is extensively employed in hacking point based systems.	While windows does not offers much competence in hacking.

UNIX VS LINUX

Now, we will observe what is the distinction between linux and unix:

Key Differences	Linux	Unix

Key Differences	**Linux**	**Unix**
Cost	Linux is liberally dispersed, downloaded during magazines, Books, website, etc. There are waged versions also accessible for Linux.	Diverse flavors of Unix have dissimilar pricing depending upon the kind of vendor.
Development	Linux is Open Source, and thousands of programmers work together online and supply to its expansion.	Unix systems have dissimilar versions. These editions are primarily created by AT&T as well as other business vendors.
User	Everybody. From home customer to developers and computer aficionado alike.	The UNIX can be employed in internet servers, workstations, and PCs.
Text made interface	BASH is the Linux evasion shell. It presents support for numerous command interpreters.	Initially prepared to job in Bourne Shell. Though, it is now well-suited with numerous others software.
GUI	Linux offers two GUIs,viz., KDE and Gnome. Although there are numerous substitutes such as Mate, LXDE, Xfce, etc.	General Desktop atmosphere and also has Gnome.
Viruses	Linux has had concerning 60-100 virus listed today which are at present not scattering.	There are among 80 to 120 viruses informed till date in Unix.
Threat detection	Threat finding and way out is extremely quick since Linux is chiefly community driven. So, if any Linux clients post any type of threat, a team of competent developers commence working to overcome this risk.	Unix client necessitate longer wait instance, to acquire the appropriate bug fixing patch.

Architectures	Originally urbanized for Intel's x86 hardware processors. It is obtainable for over twenty dissimilar kinds of CPU which also comprises an ARM.	It is obtainable on PA-RISC and Itanium machines.
Usage	Linux OS can be fitted on a variety of kinds of plans like mobile, tablet computers.	The UNIX operating scheme is employed for internet servers, workstations & PCs.
Best feature	Kernel renewed with no reboot	Feta ZFS - next age group file system DTrace - dynamic Kernel Tracing
Versions	Dissimilar editions of Linux are OpenSuse ,Redhat, Ubuntu, etc.	Dissimilar editions of Unix are BSD ,HP-UX, AIS, etc.
Supported file type	The File arrangement maintained by file kind like devpts, xfs, nfs, cramfsm ext 1 to 4, ufs, NTFS.	The File arrangement sustained by file kinds are vxfs, zfs, hfx, GPS, xfs.
Portability	Linux is moveable and is booted as of a USB Stick	Unix is not moveable
Key Differences	**Linux**	**Unix**
Source Code	The source is obtainable to the common public	The source code is not accessible to everyone.

TYPES OF FILES

In Linux and UNIX, the whole thing is a file. Directories are records; files are files, and tools like keyboard, Printer, mouse, etc. are files.

Let's look into the File types in more detail.

General Files

General Files also called as Ordinary files. They can include image, video, program or simply text. They can be in ASCII or a Binary format. These are the most commonly used files by Linux Users.

Directory Files

These files are a warehouse for other file types. You can have a directory file within a directory (sub-directory).You can take them as 'Folders' found in Windows operating system.

Device Files

In MS Windows, devices like Printers, CD-ROM, and hard drives are represented as drive letters like G: or H:. In Linux, these are represented as files. For example, if the first SATA hard drive had three primary partitions, they would be named and numbered as /dev/s*da1*, /dev/sda2 and /dev/sda3.

Note: All device files reside in the directory /dev/

All the above file types (including devices) have permissions, which allow a user to read, edit or execute (run) them. This is a powerful Linux/Unix feature. Access restrictions can be applied for different kinds of users, by changing permissions.

Linux Directory Structure

In **Linux**, everything is a file. A directory is nothing but a special file that contains details about all the files and subdirectories housed within it. The Linux contains following directories for different purpose:

i. **Root Directory (/)**

The **root directory** also represented as ― / ― is the topmost level of the system drive. Every other directory in your Linux system is located under the root directory. You may imagine the / directory similar to the C:\ directory or drive on your Windows system. This is not at all true but you may understand it as imaginary vision because Linux doesn’t have drive letters. It is commonly used to represent a filesystem. A filesystem is the hierarchy of directories that is used to organize directories and files on a computer.

ii. **Essential User binaries (/bin)**

This folder contains the essential user programs which are also called as binaries whichare present when the system is mounted in single-user mode. It also contain the files for common commands likes ls, cat, cd, cp etc. The '/**bin**' **directory** also contains executable files

iii. **Static Boot Files (/boot)**

It contains the files which are required to boot the system such Linux kernel files, GRUB configuration files.

iv. **Historical mount point for CD-ROMs (/cdrom)**

/mnt/cdrom and /cdrom are the mount point directories for the CD-ROM drive. The /cdrom directory isn't part of the Filesystem Hierarchy Standard (FHS), but you'll still find it on Ubuntu and other operating systems.

v. **Device Files (/dev)**

In Linux, the /dev directory contains device and other special files. To see a list of these items, run the following command in a terminal session.

```
ls –l /dev
```

These include terminal devices, usb, or any device attached to the system.

vi. **Configuration Files (/etc)**

The /etc/ directory contains system-wide configuration files — user-specific configuration files are located in each user's home directory.

vii. **Home Folder (/home)**

The home directory for each user takes the form /home/username (where username is the name of the user account). For example, if your user name is john, the system will have a home folder located at /home/john

viii. **Essential Shared Libraries (/lib)**

The lib folder is a library files directory which contains libraries needed by the essential binaries to be used by the system. These are supportive files which are used by an application or a process for their proper execution. The commands in /bin or /sbin dynamic library files are located just in this directory.

ix. **Recovered Files (/lost+found)**

Whenever a there is crashed in the file system, the system boots again and check is performed. Any file which got corrupted by crash will be found in lost+found folder.Every disk partition has a lost+found directory.

x. **Removable Media (/media)**

This Directory is used for mounting files systems on removable media like Zip drives, floppy drives, CD-ROM drives.

xi. **Temporary Mount Points (/mnt)**

This directory is used for temporarily mounted filesystem. For example, if you're mounting a FAT partition for some operations, it might be mounted at /mnt/fat.

xii. **Optional Packages (/opt)**

It contains subdirectories for optional software packages which are written as a result of copy/install operations.

xiii. **Kernel and Process Files (/proc)**

It is a special directory for virtual filesystem. Instead of standard files it contains special files that represent system and process information.

xiv. **Root Home Directory (/root)**

This directory is distinct from / directory which is used for the system root directory. It is the home directory for root user. Instead of using the /home/root, it uses the location at /root.

xv. **Application State Files (/run)**

This folder contains data which describe the system since it has been booted. It gives applications a standard place to store temporary/transient files containing data related to sockets and process Ids.

xvi. **System Administration Binaries (/sbin)**

It contains files related to administrative commands. This directory includes essential binaries that might be needed to run by the root user for system administrator of the system.

xvii. **SELinux Virtual File System (/selinux)**

Ubuntu doesn't use SELinux. The /selinux directory contains special files used by SELinux for security (example Fedora and Red Hat). It is similar to /proc folder.

xviii. **Service Data (/srv)**

This holds the data for system services like Apache HTTP, FTP, etc.

xix. **Temporary Files (/ tmp)**

When the system or user executes the applications, the temporary files are stored in this folder. These temporary files are usually deleted automatically when the system gets started again.

xx. **User Binaries and Read Only Data (/usr)**

This directory is meant to contain the applications and files which belong to users, dissimilar to applications and files used by the system.

xxi. Variable data Files (/var)

This folder holds numerous system files which may be related to log, mail directories, print spool, etc. that may change over time in numbers and size.

Parent, Subdirectory, Home Directory

Root Directory (/) holds the top position as the main directory. All the directories mentioned in above section from sr. No. 2 to sr. No. 21 are the subdirectory of the root directory. /Home and /root are the home directories for the normal user and root user respectively.

Naming Rules for Directory and Files

1. The file names in the Linux are case sensitive. For example a filenamesarea.txt, Area.txt and AREA.txt all are considered as different files.

2. The filename can be created either by using uppercase letters, lowercase letters, numeric numbers, –_‖ (underscore) and –.‖ (dot) symbols.

3. In Linux, other special characters such as blank space can also be used, but the rules for using other symbols are hard and it is heal their not to use them.

4. As mentioned in above point filenames may contain any character but one cannot use slash sign–/‖ (root directory), because it is reserved character and can only be used as the separator between files and directories while mentioning the pathname. One cannot use the null character.

5. .(dot) sign should not be used in the filename. Some time dot sign increases the readability of filenames but this may add confusion in understanding the extension of files. Dot sign should be used to identify the extension of the file. For example:

   - .tar.gz = zipped or Compressed files

   - .sh = used to identify the shell file

6. Older versions of the Unix operating system uses filenames upto 14 characters long. But in today's era latest version of Linux and UNIX can use filename length upto 255 characters (255 bytes).

7. A filename should be unique in a particular directory. For example, in directory /home/john, one cannot give two file name as try.txt and try.txt. However, different directory can have same filenames inside them. For example, you can create try.txt in

/tmp directory and try.txt inside /home/john directory.

PRACTICE QUESTIONS

Q1.What are the basic components of Linux?

Q2.Write down the name of some Linux variants.

Q3.Which popular office suite is available free for both Microsoft and Linux?

Q4.Suppose your company is recently switched from Microsoft to Linux and you have some MS Word document to save and work in Linux, what will you do?

Q5.What is the difference between Linux and Unix?

Q6.What is Linux Kernel? Is it legal to edit Linux Kernel?

Q7.What is the difference between BASH and DOS?

Q8.What are the process states in Linux?

Q9.What is the advantage of open source?

M.Sc. (Computer Science)
OPERATING SYSTEM

UNIT VII : LINUX COMMANDS

STRUCTURE

Objective
Linux Commands
File Permissions in Linux
Change in Ownership and the Associated Group
Advice before using Commands
Summary
Practice Exercise

OBJECTIVE

- Understanding Linux commands

OVERVIEW OF LINUX COMMANDS

The command line is one of the most powerful features of Linux. There are a countless number of commands in Linux. In this chapter, we will introduce few of most commonly and regularly used Linux commands for easy learning.

cal / ncal command

The **cal** command is used to print the calendar for a specified month and/or year on the monitor i.e. standard output device.

$cal: This will print the calendar for the current

$cal 2020: This command will print the twelve month calendar for the year 2020.

$cal 7 2020: This command will print the calendar July 2020.

mkdir command

This command is used to make the subdirectory under the current directory

$mkdir student: This command will create the directory student under the current directory

cd command

The cd command is used to change the current directory to the specified directory

$cd student: This command will change the current directory to student directory.

mv command

This command is used move the files from one folder to another folder.

$mv /home/john/try.txt /home/merry/. This command will move the try.txt file from /home/john directory to home/merry folder

$mv /home/merry/*.* /home/john/. This command will copy all the files in /home/merry folder to /home/john folder.

cp command

This command is used to copy the files from one folder to another folder.

$cp /home/john/try.txt /home/merry/. This command will copy the try.txt file from /home/john directory to home/merry folder

$cp /home/merry/*.cpp /home/john/. This command will copy all the files with extension .cpp from /home/merry folder to /home/john folder.

date command

The date command displays the current day, date, time, and year.

$ date: It will display current system date like "**Mon May 24 17:14:57 IST 2021**"

echo command

The echo command prints whatever you will write except quote. This command is used to insert the text on a file as shown below.

$echo "Welcome to The Linux Class": This command will print the line Welcome to The Linux Class.

rm command:

One can use “rm” command to delete any file or directory. To delete directories recursively we can use rm command along with -r , -R

$rm –r fname: This command will remove the folder with name fname.

$rm rea*.* : This command will remove all files starting with name prefix rea in current folder

rmdir command:

one can use rmdir to delete a directory. This command can only be used to delete an empty directory.

$rmdir dir_name: This command will remove the folder with name dir_name.

cat command:

cat is used to display the contents of the file whose name is mntioned along with the command. Example to see the text inside the file area.txt we may write the command as follows.

$cat area.txt: It will display the contents inside the filename area.txt

It is also used to merge the contents of the two files. The output is printed on the standard output device.

$cat filename1 filename2: It will display the contents of both the files together after merging.

pwd command

pwd command is used to display the name of present working directory. With the help of pwd command we are able to know about the directory in which we are working with. It gives us the absolute path, which means the path that starts from the root.

$pwd : It will the complete path of current directory. Example if you are working under user john under home directory then we will get output as /home/john.

who command

who command used to find out last system boot time, the system‟s current run level, List of the users who logged in the system and more. The different variations of who command are as follows:

Description	**Example**
The who command displays the following information for each user currently logged in to the system if no option is provided	$who
To display host name and user associated with standard input such as keyboard	$who -m –H
To show all active processes which are spawned by INIT process	$who -p –H
To show status of the users message as +, – or ?	$who -T –H
To show list of users logged in to system	$who –u
To show time of the system when it booted last time	$who -b –H
To show details of all dead processes	$who -d –H
To show system login process details	$who -l -H

To count number of users logged on to system	$who -q –H
To display current run level of the system	$who -r

To display all details of current logged in user	$who –a
To display system"s username	$whoami
To display list of users and their activities	$ w
To display user identification information	$ id

pwd command

pwd command is used to display the name of present working directory. With the help of pwd command we are able to know about the directory in which we are working with. It gives us the absolute path, which means the path that starts from the root.

$pwd : It will the complete path of current directory. Example if you are working under user john under home directory then we will get output as /home/john.

ls command

This is called list command. It is used to display the files stored in a particular directory.

ls: To view the brief, multi-column list of the files in the current directory.

ls –a: To also see files starting with symbol "dot" (configuration files that begin with a period, such as .login).

ls –la: To see the permissions, owners and size of the files along with their names.

ls -la | less: If the listing is enough long that it is not fit into one screen view then reading becomes difficult. This command combines ls with the less utility for the same.

bc Command

This command is used for arbitrary precision CLI calculator which can be used like this: **$ echo 30.07 + 16.00 | bc**

more Command

This command enables us to view the large text data one screen display at a time.

gzip Command

gzip command is used to compress a file, replaces it with a file with .gz extension. The example of usage of gzip command is as shown below:

```
$ gzip passwds.txt
$ cat filename1 filename2 | gzip > try.gz
```

tar Command

This powerful utility command is a used for archiving files in Linux as follows.

```
$ tar -czf  demo.tar.gz
```

FILE PERMISSIONS IN LINUX

Linux is a **multi-user operating system** which means it can be accessed by many users simultaneously. The multi-user concern of Linux imposes security concerns as an unsolicited or **malign user** can **corrupt, change or remove crucial data**. Linux categorizes authorization into 2 levels.

1. Ownership
2. Permission

Ownership in Linux

Every file and directory in Linux operating system can be assigned 3 different types of owner, given below.

User

The person who creates the file is by default the owner of the file. Such user is the owner of the file. Therefore, a user is also sometimes called an owner.

Group

A group or user- group consists of multiple users. All the users which belong to a same group will have the same access permissions to the file. For example there is a project where the number of people/users needs access to a file. You could add all users to a single group and assign the permissions to the group instead of assigning the permissions to each user individually. No one other than the group member can read or modify the files.

Other

In this category, any general user who can access to a file is included in other category. This person has neither produced the file, nor do he / she belong to a particular usergroup who could own the file. Practically, it means everybody else. Therefore, this category is also called as the set permissions for the world.

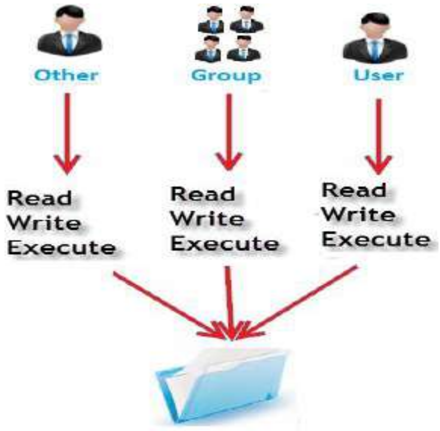
It is matter of the question that how Linux distinguishes between the three category of the users. One user 'A' cannot affect a file that contains the information of some other user 'B'. It means that you do not want another colleague in your organization to access your data. This is where the role of setting permissions comes in, and this also defines user behavior.

Let us understand the role of setting permissions in the Linux OS.

Permissions

Every file and directory in your Linux operating system has three types of permissions defined for all the 3 types of owners.

- **Read:** The read permission gives the authority to read as well as open the file. This permission on a directory gives you the capability to view its contents.
- **Write:** This permission gives a person the authority to modify the contents inside the file. You can rename, remove and add files present in the directory. For example if you don‟t have write permission on a folder but have the write permission on one of its file then you will be able to change the contents of the file but you will not be able to remove, move or rename the file placed inside the directory.
- **Execute:** Similar to Windows where files with ".exe" extension are executable. But in Linux OS, one cannot execute any file without having permission on it. If the execute permission is not granted on file, then you will not be able to execute the file. However you can see/modify the contents inside the file.



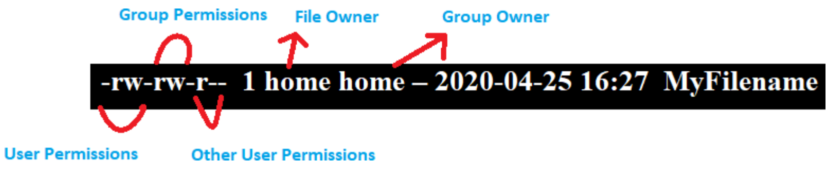
Categorization of Permissions that can be assigned to different types of Users

Let us see view this with the help of commands:

Issue command ls - l on terminal gives you the filenames with file types and access permissions.

```
ls  –  l
home$  ls –l
```

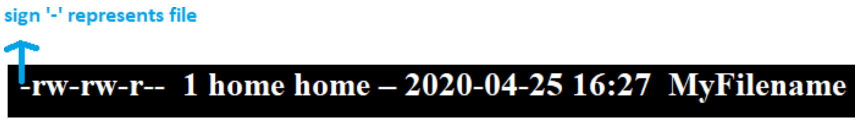
-rw-rw-r-- 1 home home – 2020-04-25 16:27 MyFilename



```
-rw-rw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

The string **'-rw-rw-r--'** in the output describes about the permissions given to the owner, user group and the world.

Here, the first '-' represents that this line indicates information about file

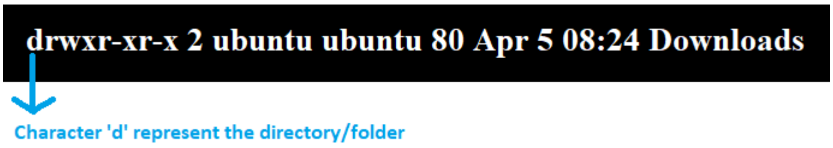


```
-rw-rw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

If it could have been a directory then **d** would have been shown as below:

drwxr-xr-x 2 ubuntu ubuntu 80 Apr 5 08:24 Downloads

In the above output, the first character „d‟ represent the directory.



```
drwxr-xr-x 2 ubuntu ubuntu 80 Apr 5 08:24 Downloads
```

Character 'd' represent the directory/folder

The characters shown in the above example are easy to remember.

- = have no permission

x = the permission to execute

w = the permission to write

r = the permission to read

Let us explore the meaning of the output as follows:

The first part in the output line is **'rw-'**. This indicates that the owner 'home' can have two permissions i.e. read and write the file but the owner is not having the permission to execute that is why sign „-„ has been used.

However, many distributions of the Linux like Ubuntu, CentOs and Fedora etc. can add users to the same group name which is the name of the user. Therefore a username 'john' may be added to a group having name as 'john'.

The second part of the output is **'rw-'** which indicates that group having named as „**home‟** have the permission to read and write the file but don‟t have the permission to edit the file**.**

Third part is used to represent the permissions for the other person from the outside world. It means there could be any other user. The string **'r--'** indicates that other users have the permission to only read the file but these persons may not be able to write and execute the file.

Using „chmod" command to change the directory/file permissions

Permissions help you to restrict the others in performing operations on files/directories in form of read, write and execute. This can be accomplished by changing permissions on files/directories.

We can use the '**chmod'** command which stands for 'change mode'. Using the command, we can set permissions (read, write, execute) on a file/directory for the owner, group and the world. The Syntax of command is as follows:

```
chmod permissions filename
```

There are 2 types of modes for using chmod command -

1. **Absolute mode**
2. **Symbolic mode**

Absolute Mode / Numeric Mode

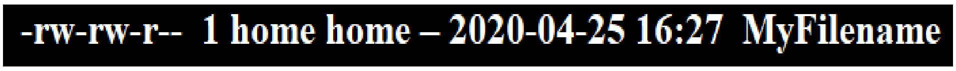
In this technique/mode, the permissions on files are represented as three-digit octal number instead as characters. The following table represents the different numbers for each type of permissions.

Number	Permission Type	Symbol
0	No Permission	---
1	Execute	--x
2	Write	-w-
3	Execute + Write	-wx
4	Read	r--
5	Read + Execute	r-x

6	Read +Write	rw-
7	Read + Write +Execute	Rwx

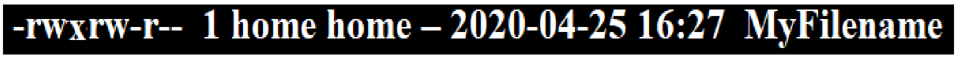
Let us look at the examples of using chmod command.

Command **"ls –l MyFilename"** shows the current permissions on the MyFilename



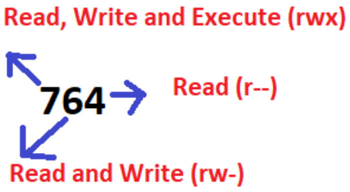
```
-rw-rw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

If we issue the command **"chmod 764 MyFilename"** then we will see the following results



```
-rwxrw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

In the above-given terminal window, we have changed the permissions of the file 'sample to '764'.



'764' absolute coding can be interoperated as follows:

- Owner can read, write and execute
- User group can read and write
- World can only read

This can be seen in the output as '-rwxrw-r-

In this way; you can change the permissions on file by assigning an absolute number.

Symbolic Mode

In the Absolute mode, you change permissions for all 3 owners. In the symbolic mode, you can modify permissions of a specific owner. It makes use of mathematical symbols to modify the file permissions.

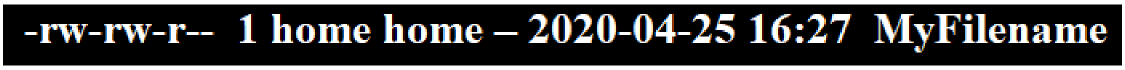
Operator	Action of applying the operator
+	Adds the particular permission to the file / directory
-	Removes the particular permission from file / directory
=	Sets the new permission by overriding the previous permissions.

The various types of owners of file and directories are given as follows:

User Denotations	
U	user/owner
G	Group
O	Other
A	All

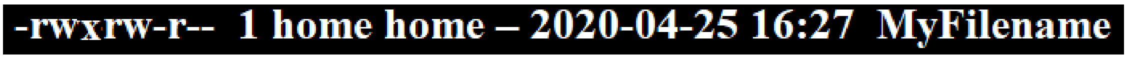
In symbolic mode we shall be setting permissions by making use of characters like rwx unlike numbers in absolute mode eg 764. Let us apply example for symbolic mode using the previous file permissions on MyFilename:

Command **“ls –l MyFilename”** shows the current permissions on the MyFilename



```
-rw-rw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

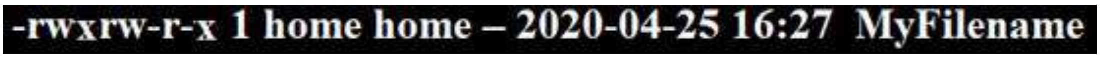
After applying the command **"Chmod u = rwx MyFilename"**, we shall get the following result:



```
-rwxrw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

In above command "u" stands for user. It means new permissions will be set for the user. The command will assign the user permissions of read, write and execute to the MyFilename.

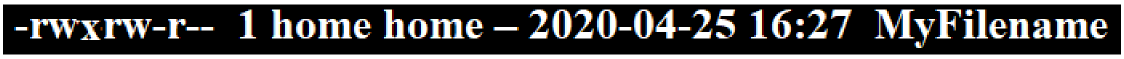
Let us apply another command as **"chmod o+x MyFilename"**. Then the permissions will be changed as follows.



```
-rwxrw-r-x 1 home home – 2020-04-25 16:27  MyFilename
```

String **"o+x"** in above command will add the execute permission for the other user.

Let us again apply command as **"chmod o-x MyFilename"**. Then the execute permission which was given just now will be removed. The output of **"ls –l MyFilename"** command will look as follows after applying the command.



```
-rwxrw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

CHANGE IN OWNERSHIP AND THE ASSOCIATED GROUP

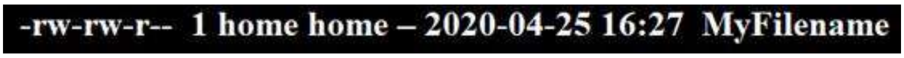
To change the ownership of particular file/directory, we may use the **"chown"** command as follows:

chown new_user_name

If there is need to change the group along with the user for a file or directory, then we may use the following command

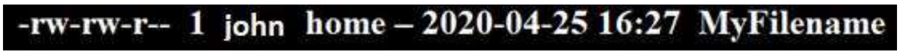
chown new_user_name:new_group_name filename

Let us look at the output of **"ls – l MyFilename"** command



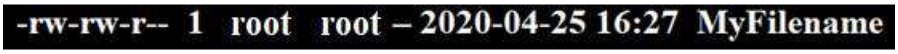
```
-rw-rw-r--  1 home home – 2020-04-25 16:27  MyFilename
```

Now apply the “**chown john MyFilename”** command and then apply **“ls – l MyFilename”** command. It you are not logged in as root user then you may use **“sudo chown john MyFilename”** command to act as super user. The system may ask you the password for the super user in such case.



```
-rw-rw-r-- 1 john home – 2020-04-25 16:27 MyFilename
```

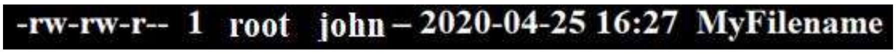
Let us change the user and group both to root by applying **“sudo chown root:root MyFilename”** command. The output of the file may look as follows after applying the command.



```
-rw-rw-r-- 1 root root – 2020-04-25 16:27 MyFilename
```

If you want to change only the group of the user then may also apply **“chgrp”** command. Command “**chgrp”** stands for change group.

Let us apply the command **“sudo chgrp john MyFilename”** to change the group from root to john. Then we will get output as follows:



```
-rw-rw-r-- 1 root john – 2020-04-25 16:27 MyFilename
```

ADVICE BEFORE USING COMMANDS

- The file stored in the **/etc/group** contains the name of all the groups defined in the system at any moment of time.
- We may use the command **"groups"** on terminal to find the name of all the groups where we are a members.
- We may use the command **“newgrp”** to work as a member a group other than your default group
- No file/directory can own to two groups simultaneously.
- Linux do not have nested groups.
- x- eXecuting a directory means Being allowed to "enter" a dir and gain possible access to sub-directories.

SUMMARY

- Linux being a multi-user system uses permissions and ownership for security.
- There are three user types on a Linux system viz. User, Group and Other
- Linux divides the file permissions into read, write and execute denoted by r,w, and x
- The permissions on a file can be changed by 'chmod' command which can be further divided into Absolute and Symbolic mode
- The 'chown' command can change the ownership of a file/directory. Use the following commands: chown user file or chown user:group file
- The 'chgrp' command can change the group ownership **chrgrp group filename**
- What does x - executing a directory mean? A: Being allowed to "enter" a dir and gain possible access to sub-dirs.

PRACTICE QUESTIONS

Q1.What are the basic commands for user management?

Q2.Which command is used to uncompressed grip files?

Q3.What are the modes used in VI editor?

Q4.What are the file permissions in Linux?

Q5.Which are the Linux Directory Commands?

Q6.What are inode and process id?

Q7.Explain Process Management System Calls in Linux

Q8.Explain the redirection operator.

Q9.How to copy a file in Linux?

Q10.How to terminate a running process in Linux?

Q11.How to create a new file or modify an existing file in vi?

Q12.Explain Regular Expressions and Grep

M.Sc. (Computer Science)
OPERATING SYSTEM

UNIT VIII: SHELL SCRIPTING

STRUCTURE

Objectives
Introduction to Shell
Shell Scripting
Shell Prompt
Variable Names
Positional Parameter / Command Line Arguments
Practice Questions

OBJECTIVE

To learn the usefulness of shell scripting

INTRODUCTION TO SHELL

- The shell is a user program or it is an environment provided for user interaction.
- It is a command language interpreter that executes commands read from the standard input device such as keyboard or from a file.
- The shell gets started when you log in or open a console (terminal).
- Quick and dirty way to execute utilities.
- The shell is not part of system kernel, but uses the system kernel to execute programs, create files etc.

Different shells are available in Linux which are as under:

- **BASH (Bourne-Again SHell)** - Most common shell in Linux. It's Open Source.
- **CSH (C SHell) -** The C shell's syntax and usage are very similar to the C programming language.
- **KSH (Korn SHell) -** Created by David Korn at AT & T Bell Labs. The Korn Shell also was the base for the POSIX Shell standard specifications.
- **TCSH** - It is an enhanced but completely compatible version of the Berkeley UNIX C shell (CSH).

SHELL SCRIPTING

A shell script is just a normal Linux file which contains Linux and shell commands. The simplest shell scripts simply group together commonly used sequences of commands.

More complex scripts use the shell's programming syntax to perform more advanced tasks. Shell scripts are not compiled but interpreted. This means that each time they are run a shell is executed to read the file and run the commands it contains. Languages which are compiled, such as C, will produce a compiled code which will run faster than an equivalent shell script.

Shell scripts usually begin with a #! and a shell name.
For example: #!/bin/sh
If they do not, the user's current shell will be used

SHELL PROMPT

There are various ways to get shell access:

- **Terminal** - Linux desktop provide a GUI based login system. Once logged in you can gain access to a shell by running X Terminal (XTerm), Gnome Terminal (GTerm), or KDE Terminal (KTerm) application.
- **Connect via secure shell (SSH)** - You will get a shell prompt as soon as you log in into remote server or workstation.
- **Use the console** - A few Linux system also provides a text-based login system. Generally you get a shell prompt as soon as you log in to the system.

You may type the following into the Terminal to find out the shells available in your system.

cat /etc/shells

Why write shell scripts?

– To avoid repetition: If you do a sequence of steps with standard Linux commands over and over, why not do it all with just one command?

– To automate difficult tasks: Many commands have subtle and difficult options that you don’t want to figure out or remember every time.

– Creating your own power tools/utilities.

– Automating command input or entry.

– Customizing administrative tasks.

– Creating simple applications.

– Since scripts are well tested, the chances of errors are reduced while configuring services or system administration tasks such as adding new users.

Advantages

- Easy to use.
- Quick start, and interactive debugging.
- Time Saving.
- Sys Admin task automation.
- Shell scripts can execute without any additional effort on nearly any modern UNIX / Linux / BSD / Mac OS X operating system as they are written an interpreted language.

Disadvantages

- Compatibility problems between different platforms.
- Slow execution speed.
- A new process launched for almost every shell command executed.

VARIABLE NAMES

Variables names may comprise upper and lower case alphabetic characters, digits and underscores. A user defined variable name cannot start with a digit. As with most things in Unix variable names are case sensitive, e.g. FOO and foo are two distinct variables.

In Linux, there are two types of variable

1) System variables - Created and maintained by Linux itself. This type of variable defined in CAPITAL LETTERS.

2) User defined variables (UDV) - Created and maintained by user. This type of variable defined normally by using lower case LETTERS.

Some System variables

You can see system variables by giving command like $ set, Some of the important System variables are

System Variable	**Meaning**
BASH=/bin/bash	Our shell name
BASH_VERSION=1.14.7(1)	Our shell version name
COLUMNS=80	No. of columns for our screen
HOME=/home/vivek	Our home directory
LINES=25	Number of columns for our screen
LOGNAME=students	Our logging name
OSTYPE=Linux	Our o/s type : -)
PATH=/usr/bin:/sbin:/bin:/usr/sbin	Our path settings
PS1=[\u@\h \W]\$	Our prompt settings
PWD=/home/students/Common	Our current working directory
SHELL=/bin/bash	Our shell name
USERNAME=vivek	User name who is currently login to this PC

Special Shell Variables

Some variables have special significance to the shell. For example, the PATH variable is used by the shell to contain the list of directories to search for commands. Other variables have special meaning to other Linux utilities, for example the TERM variable contains the current terminal type.

```
bash$ variable1=23
```

Use echo command to display variable value.

```
bash$ echo variable1
        variable1
bash$ echo $variable1
        23
```

To display the program search path, type:

```
bash$ echo "$PATH"
```

To display your prompt setting, type:

```
bash$ echo "$PS1"
```

All variable names must be prefixed with $ symbol, and the entire construct should be enclosed in quotes. Try the following example to display the value of a variable without using $ prefix:

```
bash$ echo "HOME"
```

To display the value of a variable with echo $HOME:

```
bash$ echo "$HOME"
```

You must use $ followed by variable name to print a variable's contents.

The variable name may also be enclosed in braces:

```
bash$ echo "${HOME}"
```

This is useful when the variable name is followed by a character that could be part of a variable name:

```
bash$ echo "${HOME} work"
```

To define User Defined Variable (UDV) use following syntax

Syntax: *variablename=value*

NOTE: Here 'value' is assigned to given 'variablename' and Value must be on right side = sign For e.g.

```
$ no=10 # this is ok
$ 10=no # Error, NOT Ok, Value must be on right side of = sign.
```

To define variable called 'vech' having value Bus

```
$ vech=Bus
```

To define variable called n having value 10

```
$ n=10
```

How to Define variable x with value 10 and print it on screen

```
$ x=10
$ echo $x
```

Linux Shell Script Tutorial

How to Define variable xn with value Rani and print it on screen

```
$ xn=Rani
$ echo $xn
```

How to print sum of two numbers, let's say 6 and 3

```
$ echo 6 + 3
```

This will print 6 + 3, not the sum 9, To do sum or math operations in shell use expr, syntax is as

follows Syntax: *expr op1 operator op2*

Where, op1 and op2 are any Integer Number (Number without decimal point) and operator can be

+ Addition

- Subtraction

/ Division

% Modular, to find remainder For e.g. 20 / 3 = 6 , to find remainder 20 % 3 = 2, (Remember its
integer calculation)

* Multiplication

$ expr 6 + 3

Now It will print sum as 9 , But

$ expr 6+3

will not work because space is required between number and operator (See Shell Arithmetic)
Q.4.How to define two variable x=20, y=5 and then to print division of x and y (i.e. x/y)

$x=20

$ y=5

$ expr x / y

Q.5.Modify above and store division of x and y to variable called z

$ x=20

$ y=5

$ z=`expr x / y`

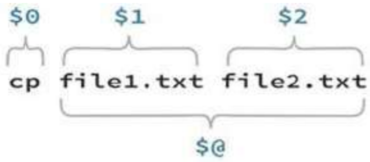
$ echo $z

Note : For third statement, read Shell Arithmetic.

POSITIONAL PARAMETER / COMMAND LINE ARGUMENTS

Command line arguments are important part of writing scripts. Command line arguments define the expected input into a shell script. For example, we may want to pass a file name or folder name or some other type of argument to a shell script.

A **positional parameter** is a variable within a **shell** program; its value is set from an **argument** specified on the **command** line that invokes the program. **Positional parameters** are numbered and are referred to with a preceding ``$'': $1, $2, $3, and so on. A **shell** program may reference up to nine **positional parameters**.



Consider the following bash command. Let the command name is mycommand. The command line has three parameters: one, two, and three four.

$ mycommand one two “three four”

Variable name	Value
$0	Mycommand
$1	One
$2	Two
$3	three four
$#	3
$@	one two three four
$*	one two three four
$!	This parameter represents the process number of the background that was executed last.
$?	This parameter represents exit status of the last command that was executed. Here 0 represents success and 1 represents failure.
$_	This parameter represents the command which is being executed previously.
$-	This parameter will print the current options flags where the set command can be used to modify the options flags.

```
$ cat  program.sh
echo "The File Name is: $0"
echo "The First argument is: $1"
echo "The Second argument is: $2"
```

```
$ sh program.sh ab cd
The File Name: program.sh
The First argument is: ab
The Second argument is: cd
```

If any arguments are supplied, they become the positional parameters when filename is executed. Otherwise, the positional parameters remain unchanged.

if-then-else

Like most programming languages, shell script supports the if statement, with or without an else. The general form is below:

```
if (condition)
        simple commands

if (condition)
then
        commands
        …
fi

if (condition)
then
```

... Else

...fi

commands

Commands

You can have nested if-else-fi

The condition used as the predicate can be any program or expression. The results areevaluated with a 0 return being true and a non-0 return being false.

If ever there is the need for an empty if-block, the null command, a :, can be used in place ofa command to keep the syntax legal.

The following is a nice, quick example of an if-else:

```
if [ "$LOGNAME"="guna" ]then
elsefi                    printf "%s is logged in" $LOGNAMEprintf "Intruder! Intruder!"
```

The elif construct

Shell scripting also has another construct that is very helpful in reducing deep nesting. It is unfamilar to those of us who come from languages like C and Perl. It is the elif, the "else if". This probably made its way into shell scripting because it drastically reduces the nesting that would otherwise result from the many special cases that real-world situatins present --without functions to hide complexity (shell does have functions, but not parameters -- and they are more frequently used by csh shell scripters than traniditonalists).

```
if conditionthen
commandcommand
...
commandelif condition then
commandcommand
...
commandelif condition then
commandcommand
...
command
fi
```

The switch statement

Much like C, C++, or Java, shell has a case/switch case statement. The form is as follows:

```
case “$variable” inpattern1)
command

command
...
command
;; # Two ;;'s serve as the breakPattern2)
commandcommand
...
command
;; # Two ;;'s serve as the breakPattern3)
```

```
commandcommand
...
command
;; # Two ;;'s serve as the breakesac
```

Here's a quick example:

```
#!/bin/sh case "$char"in
"+")
                ans=`expr $1 + $3`
                printf "%d %s %d = %d\n" $1 $2 $3 $ans
                ;;
"-")
                ans=`expr $1 - $3`
                printf "%d %s %d = %d\n" $1 $2 $3 $ans
                ;;
"\*")
                ans=`expr "$1 * $3"`
                printf "%d %s %d = %d\n" $1 $2 $3 $ans
                ;;
"/")
                ans=`expr $1 / $3`
                printf "%d %s %d = %d\n" $1 $2 $3 $ans
                ;;
# Notice this: the default case is a simple *
*)



esac
printf "Don't know how
to do that.\n"
;
```

The for Loop

The for loop provides a tool for processing a list of input. The input to for loop is a list of values. Every iteration through the loop extracts one value into a variable and then enters the body of the loop. The loop stops when the extract fails because there are no more values in the list.

for loop operates on a list and repeats commands in the block for each element on the list.

```
for x in [ list ]do
commands
done
```

Let's consider the following example which prints each of the command line arguments, oneat a time. We'll extract them from "$@" into $arg:

```
for var in "$@"do
printf "%s\n" $var
done
```

Much like C or Java, shell has a break command, also. As you might guess, it can be used to get out of a loop. Consider this example which stops printing command line arguments, whenit gets to one whose value is "quit":

```
for var in "$@"do
                              if [ "$var" = "quit" ]break
thenfi                        printf "%s\n" $var
done
```

Similarly, shell has a continue that works just like it does in C or Java.

```
for var in "$@"do
                              if [ "$var" = "me" ]
then
                              continue
elif [ "$var" = "mine" ]then
continue
elif [ "$var" = "myself" ]
```

```
then
                    continue
fi
if [ "$var" = "quit" ]then
break
fi
printf "%s\n" $vardone
```

The while and until Loops

Shell has a while loop similar to that seen in C or Java. It continues until the predicate is false. And, like the other loops within shell, break and continue can be used. Here's an example of a simple while loop:

```
# This lists the files in a directory in alphabetical order

# It continues until the read fails because it has reached the end of inputls | sort |
while read filedo
echo $file
done
```

In the above code, | called a “pipe” directs output from one process to another process. For example, ls | sort takes the output from ls command, and sort the output stream received. Pipeis a form of inter process communication which we will discuss later.

There is a similar loop, the until loop that continues until the condition is successful -- inother words, while the command failes. This will pound the user for input until it gets it:

```
printf "ANSWER ME! "until read $answer
do
printf "ANSWER ME! "
done
```

PRACTICE QUESTIONS

Program1: Write a shell script program to display “HELLO WORLD”.Program2: Write a shell script to find the factorial of given integer

Program3: Write a shell Script program to check whether the given number is even **or odd.**

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University
(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-II
Course: DATA STRUCTURE & ALGORITHMS (MSCS-2-02T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

M.Sc. (Computer Science) Programme Coordinator:
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:
Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Business Management & Commerce:
Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Social Sciences and Liberal Arts:
Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

PROGRAMME COORDINATOR

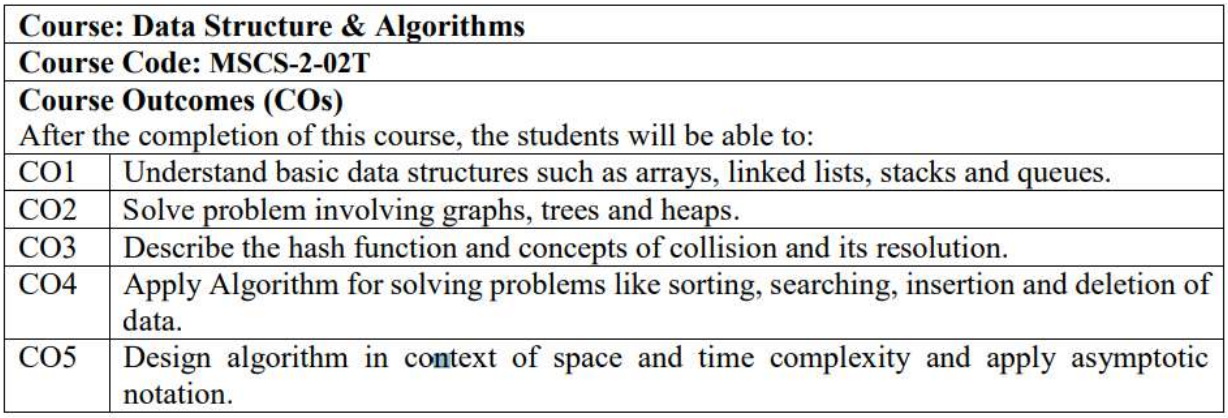
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

COURSE COORDINATOR & CONTENT WRITER:

Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE OUTCOMES (COs)

After Completion of the course, learners will be able to



Course: Data Structure & Algorithms	
Course Code: MSCS-2-02T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Understand basic data structures such as arrays, linked lists, stacks and queues.
CO2	Solve problem involving graphs, trees and heaps.
CO3	Describe the hash function and concepts of collision and its resolution.
CO4	Apply Algorithm for solving problems like sorting, searching, insertion and deletion of data.
CO5	Design algorithm in context of space and time complexity and apply asymptotic notation.

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

DATA STRUCTURE & ALGORITHMS

TotalMarks:100
External Marks:70
Internal Marks:30
Credits:4
Pass Percentage:40%

Section A

Unit I: Data Structure: Introduction to data structure and algorithm, various phases of algorithms, Pointers, working with pointers, pointers and function, structure, union, classification of data structures Algorithm analysis: Time space trade offalgorithms and Big O notation.

Unit II: Arrays: Introduction, one dimensional and multidimensional arrays, memory representation of arrays, operations on arrays, sparse arrays and sparse matrices and their implementation, Advantages and limitation of arrays.

Unit II: Linked List: Introduction; operation on linked list, circular linked list, doubly linked list, headerlinked list, implementation of linked list, application of linked lists.

Unit III: Stacks: Introduction; array representation of stacks, Operation on stacks; Linked representation of stacks Implementation of stacks, Application of stacks: matching parenthesis, evaluation of arithmetic expressions, and conversion from infix to post fix, recursion.

SECTION B

Unit IV: Queues: Introduction, operation on queues, linked representation & implementation of queue, Applications of queues, circular queue, memory representation of queues, dequeus, priority queues, Multiple queues, application of queues.

Unit V: Trees: Introduction; Binary Tree; Complete Binary Trees, Extended Binary Trees, representation of binary trees in the memory , traversing a binary tree, Binary Search Tree, Operations on Binary Search Tree; Balanced Trees- AVL; B- Trees; Heap, Applications of trees

Unit VI: Graphs: Introduction Graph: Graph terminology, Memory Representation of Graphs: adjacency matrix representation of graphs, adjacency list or linked representation of graphs, graph traversal algorithms, Operations performed ongraphs.

Unit VII: Searching: Linear Search, Binary Search, Fibonacci Search, Sorting : Selection Sort, Insertion Sort, Merge Sort, Bucket Sort, Radix Sort, Quick Sort and Heap Sort

Reference Books:

1. A.Tanenbaum,Y.LanhgsamandA.J.Augenstein,"Data Structures UsingC",PHI.
2. Loomis,Marry,“Data Management and File Structures”, PHI
3. Seymour Lipschultz,“ Theory and Practice of Data Structures" Mc Graw-Hill.
4. E.Horowitz and S.Sahni, “Data Structures with Pascal",Galgotia.
5. M.J.Folk, B.Zoellick, G Riccardi,“File Structures”, Pearson Education.

M.Sc. (Computer Science)

SEMESTER-2

COURSE: DATA STRUCTURE AND ALGORITHMS

UNIT 1: INTRODUCTION TO DATA STRUCTURE

1.1 INTRODUCTION

- Data Structures is the concept of set of algorithms used to structure the information.
- These algorithms are implemented using C, C++, Java, etc
- Structure the information means store and process data in an efficient manner.
- To store and process data we may use the following operations
    - create()
    - sorting()
    - insert()
    - merging()
    - delete()
    - splitting()
    - display()
    - traversal()
    - searching()
- So data structure may contain algorithms, use for different operations implement these algorithms by aprogramming language
- For example for stack data structure write algorithms for different operations
    - Push
    - Pop
    - Display

1.2 CLASSIFICATION OF DATA STRUCTURES:

- Data structures are normally classified into two types.
- They are primitive data structures and non-primitive data structures.

(i) Primitive data structures:

- Primitive data structures are built in types in most programming languages. They are
- Integer: It is whole numbers. i.e. negative values,0,positive values
- Float: It is fractional numbers
- Character: It is character values
- Boolean: it represents true or false.

(ii) Non-primitive data structures:

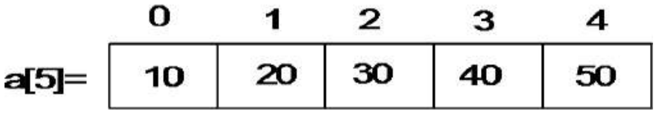
- These are derived from primitive data structures.
- They are Array, Structure, Union, Files etc
- A Non-primitive data type is further divided into Linear and Non-Linear data structure.

(a) Linear data structures:

- Here the data elements are connected in a sequence manner.
- Examples are Arrays, Linked List, Stacks and Queues.

Array:

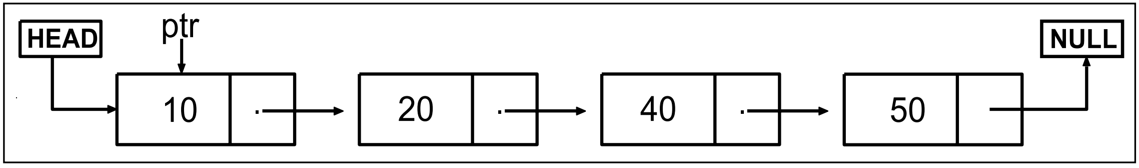
➢ It is collection of elements of the same type



Linked List:

- *linked list* or single linked list is a sequence of elements in which every element has *link* to its nextelement in the sequence.
- Every element is called as a "node". Every "node" contains two fields, *data* and link. The data is a valueor string and link is an address of next node.

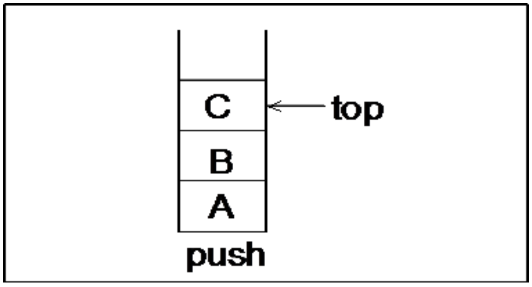
- The first node is called HEAD which is an empty node contains an address of the first node so it link tothe first node.
- The first node link to the second node and so on.
- The last node does not link to address but link to NULL. Let ptr be a pointer to the linked list. Theexample is given below



Stack:

- A stack is a data structure in which additions and deletions are made at the top of the stack. So we canperform two operations on stack.

1. Adding elements into the stack known as push;2.Deleting elements from the stack known as pop

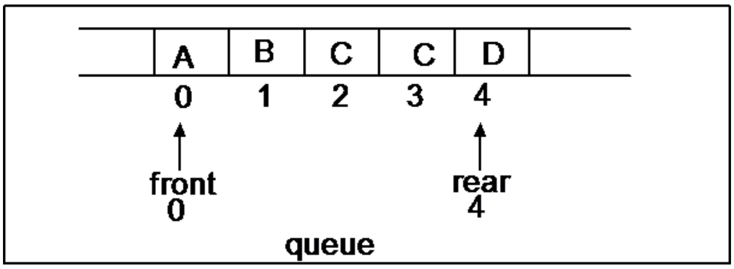


Queue:

 A queue is a data structure in which additions are made at one end and deletions are made at the otherend. We can represent a queue in an array.

 Here we can perform two operations on queue.

1. Adding elements into the queue known as insertion at rear 2.Deleting elements from the queue known as deletion from front



(b) Non-linear data structures:

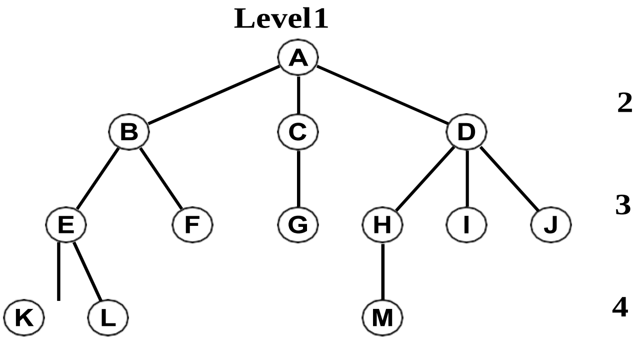
➢ Here data elements are not connected in a sequence manner.

➢ Examples are: Trees and Graphs.

Tree:

➢ The tree is defined as a finite set of one or more nodes such that1.One node is called a root node and

2. Remaining nodes partitioned into sub trees of the root.

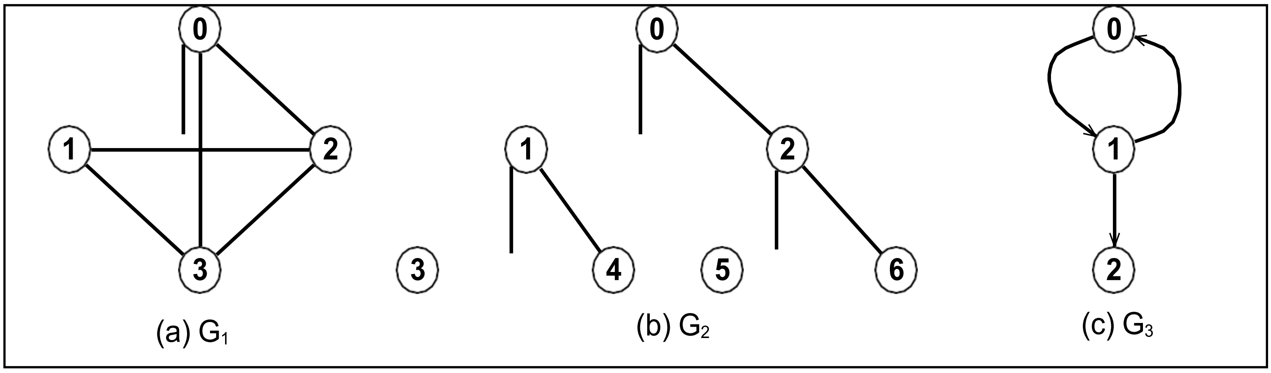


Graph:

❖ A graph is a pictorial representation of a set of <u>points</u> or nodes termed as <u>**vertices,**</u> and the <u>links</u> thatconnect the vertices are called <u>**edges**</u>.

❖ A Graph(G) consists of two sets V and E where V is called vertices and E is called edges. We also writeG = (V,E) to represent a graph.

❖ A Graph may be directed graph and undirected graph.



(a) G_1 (b) G_2 (c) G_3

❖ The Fig(a),Fig(b) are called undirected graph & Fig(c) is called directed graph.

Differences between Linear and Non Linear Data Structures:

Linear Data Structure	**Non-Linear Data Structure**
Every data element is connected to its previous & next one	Every data element is connected with many other data elements.
Data is arranged in a sequence manner	Data is not arranged in a sequence manner
Data can be traversed in a single run	Data cannot be traversed in a single run
Ex: Array, Stack, Queue, Linked List	Ex: Tree, Graph
Implementation is easy	Implementation is difficult

1.3 ALGORITHM DEVELOPMENT PROCESS

Every problem solution starts with a plan. That plan is called an algorithm.

1.3.1 An algorithm is a plan for solving a problem.

There are many ways to write an algorithm. Some are very informal, some are quite formal and mathematical in nature, and some are quite graphical. The instructions for connecting a DVD player to a television are an algorithm. A mathematical formula such as πR2 is a special case of an algorithm. The form is not particularly important as long as it provides a good way to describe and check the logic of the plan.

The development of an algorithm (a plan) is a key step in solving a problem. Once we have an algorithm, we can translate it into a computer program in some programming language. Our algorithm development process consists of five major steps.

Step 1: Obtain a description of the problem.

Step 2: Analyze the problem.

Step 3: Develop a high-level algorithm.

Step 4: Refine the algorithm by adding more detail.

Step 5: Review the algorithm.

Step 1: Obtain a description of the problem.

This step is much more difficult than it appears. In the following discussion, the word client refers to someone who wants to find a solution to a problem, and the word developer refers to someone who finds a way to solve the problem. The developer must create an algorithm that will solve the client's problem.

The client is responsible for creating a description of the problem, but this is often the weakest part of the process. It's quite common for a problem description to suffer from one or more of the following types of defects: (1) the description relies on unstated assumptions, (2) the description is ambiguous, (3) the description is incomplete, or (4) the description has internal contradictions. These defects are seldom due to carelessness by the client. Instead, they are due to the fact that natural languages (English, French, Korean, etc.) are rather imprecise. Part of the developer's responsibility is to identify defects in the description of a problem, and to work with the client to remedy those defects.

Step 2: Analyze the problem.

The purpose of this step is to determine both the starting and ending points for solving the problem. This process is analogous to a mathematician determining what is given and what must be proven. A good problem description makes it easier to perform this step.

When determining the starting point, we should start by seeking answers to the following questions:

What data are available?

Where is that data?

What formulas pertain to the problem?

What rules exist for working with the data?

What relationships exist among the data values?

When determining the ending point, we need to describe the characteristics of a solution. In other words, how will we know when we're done? Asking the following questions often helps to determine the ending point.

What new facts will we have?

What items will have changed?

What changes will have been made to those items?

What things will no longer exist?

Step 3: Develop a high-level algorithm.

An algorithm is a plan for solving a problem, but plans come in several levels of detail. It's usually better to start with a high-level algorithm that includes the major part of a solution, but leaves the details until later. We can use an everyday example to demonstrate a high-level algorithm.

Problem: I need a send a birthday card to my brother, Rohit .

Analysis: I don't have a card. I prefer to buy a card rather than make one myself.

High-level algorithm:

Go to a store that sells greeting cards

Select a card

Purchase a card

Mail the card

This algorithm is satisfactory for daily use, but it lacks details that would have to be added were a computer to carry out the solution. These details include answers to questions such as the following.

"Which store will I visit?"

"How will I get there: walk, drive, ride my bicycle, take the bus?"

"What kind of card does Rohit like: humorous, sentimental, risqué?"

These kinds of details are considered in the next step of our process.

Step 4: Refine the algorithm by adding more detail.

A high-level algorithm shows the major steps that need to be followed to solve a problem. Now we need to add details to these steps, but how much detail should we add? Unfortunately, the answer to this question depends on the situation. We have to consider who (or what) is going to implement the algorithm and how much that person (or thing) already knows how to do. If someone is going to purchase Rohit's birthday card on my behalf, my instructions have to be adapted to whether or not that person is familiar with the stores in the community and how well the purchaser known my brother's taste in greeting cards.

When our goal is to develop algorithms that will lead to computer programs, we need to consider the capabilities of the computer and provide enough detail so that someone else could use our algorithm to write a computer program that follows the steps in our algorithm. As with the birthday card problem, we need to adjust the level of detail to match the ability of the programmer. When in doubt, or when you are learning, it is better to have too much detail than to have too little.

Most of our examples will move from a high-level to a detailed algorithm in a single step, but this is not always reasonable. For larger, more complex problems, it is common to go through this process several times, developing intermediate level algorithms as we go. Each time, we add more detail to the previous algorithm, stopping when we see no benefit to further refinement. This technique of gradually working from a high-level to a detailed algorithm is often called stepwise refinement.

Stepwise refinement is a process for developing a detailed algorithm by gradually adding detail to a high-level algorithm.

Step 5: Review the algorithm.

The final step is to review the algorithm. What are we looking for? First, we need to work through the algorithm step by step to determine whether or not it will solve the original problem. Once we are satisfied that the algorithm does provide a solution to the problem, we start to look for other things. The following questions are typical of ones that should be asked whenever we review an algorithm. Asking these questions and seeking their answers is a good way to develop skills that can be applied to the next problem.

Does this algorithm solve a very specific problem or does it solve a more general problem? If it solves a very specific problem, should it be generalized.

Pointers

A **pointer** is a variable that stores the address of another variable. Unlike other variables that hold values of a certain type, pointer holds the address of a variable. For example, an integer variable holds (or you can say stores) an integer value, however an integer pointer holds the address of a integer variable.

A simple example to understand how to access the address of a variablewithout pointers?

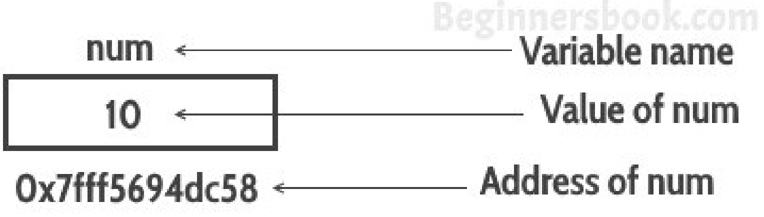
In this program, we have a variable num of int type. The value of num is 10 and this value must be stored somewhere in the memory, right? A memory space is allocated for each variable that holds thevalue of that variable, this memory space has an address. For example we live in a house and our house has an address, which helps other people to find our house. The same way the value of the variable is stored in a memory address, which helps the C program to find that value when it is needed.

So let's say the address assigned to variable num is 0x7fff5694dc58, which means whatever value wewould be assigning to num should be stored at the location: 0x7fff5694dc58. See the diagram below.

```
#include <stdio.h>
int main()
{
   int num = 10;
   printf("Value of variable num is: %d", num);
   /* To print the address of a variable we use %p
    * format specifier and ampersand (&) sign just
    * before the variable name like &num.
    */
   printf("\nAddress of variable num is: %p", &num);
   return 0;
}
```

Output:

```
Value of variable num is: 10
Address of variable num is: 0x7fff5694dc58
```



M.Sc. (Computer Science)

SEMESTER-2

COURSE: DATA STRUCTURE AND ALGORITHMS

UNIT 2: LINKED LIST

2.1 INTRODUCTION

A linked list is a collection of data elements called nodes in which the linear representation is given by links from one node to the next node. A linked list does not store its elements in consecutive memory locations and the user can add any number of elements to it.

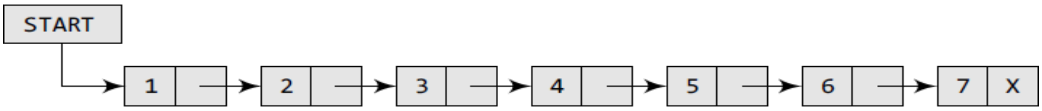
The elements in a linked list can be accessed only in a sequential manner. But like an array, insertions and deletions can be done at any point in the list in a constant time.

A linked list, in simple terms, is a linear collection of data elements. These data elements are called nodes. Linked list is a data structure which in turn can be used to implement other data structures.

Thus, it acts as a building block to implement data structures such as stacks, queues, and their variations.

Basic Terminology:

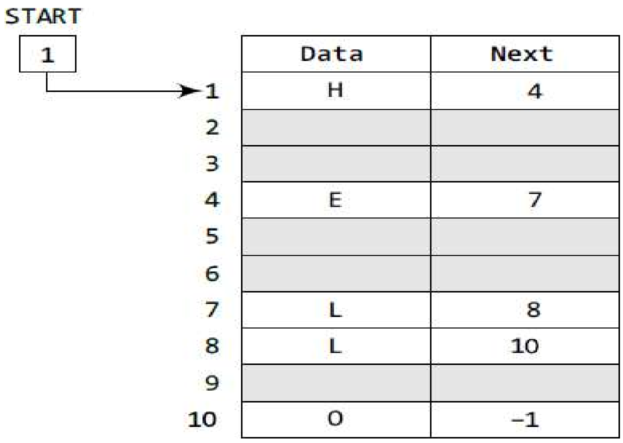
A linked list can be perceived as a train or a sequence of nodes in which each node contains one or more data fields and a pointer to the next node.



We can see a linked list in which every node contains two parts, an integer and a pointer to the next node. The last node will have no next node connected to it, so it will store a special value called NULL.

Since in a linked list, every node contains a pointer to another node which is of the same type, it is alsocalled a self-referential data type.

Let us see how a linked list is maintained in the memory. When we traverse DATA and NEXT in this manner, we finally see that the linked list in the above example stores characters that when put together form the word HELLO.



	Data	Next
1	H	4
2		
3		
4	E	7
5		
6		
7	L	8
8	L	10
9		
10	O	-1

Linked Lists versus Arrays:

Both arrays and linked lists are a linear collection of data elements. But unlike an array, a linked list does not store its nodes in consecutive memory locations. Another point of difference between an array and a linked list is that a linked list does not allow random access of data. Nodes in a linked list can be accessed only in a sequential manner.

Another advantage of a linked list over an array is that we can add any number of elements in the list. This is not possible in case of an array.

Memory Allocation and De-allocation for a Linked List:

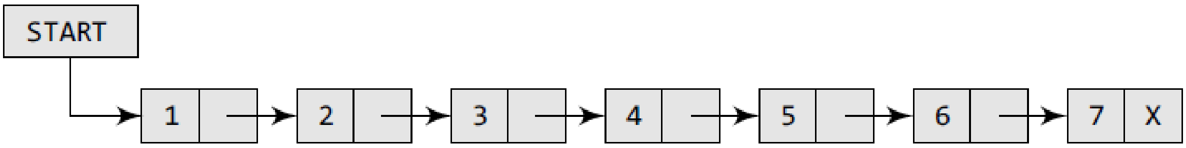
If we want to add a node to an already existing linked list in the memory, we first find free space in the memory and then use it to store the information.

Now, the question is which part of the memory is available and which part is occupied? When we delete a node from a linked list, then who changes the status of the memory occupied by it from occupied to available? The answer is the operating system.

The operating system scans through all the memory cells and marks those cells that are being used by some program. Then it collects all the cells which are not being used and adds their address to the free pool, so that these cells can be reused by other programs. This process is called garbage collection.

2.2 SINGLE LINKED LISTS:

A singly linked list is the simplest type of linked list in which every node contains some data and a pointer to the next node of the same data type.

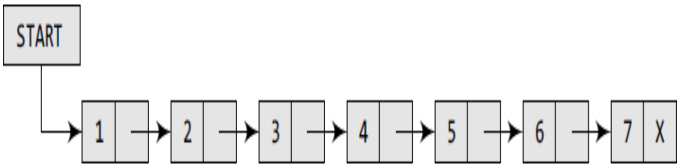


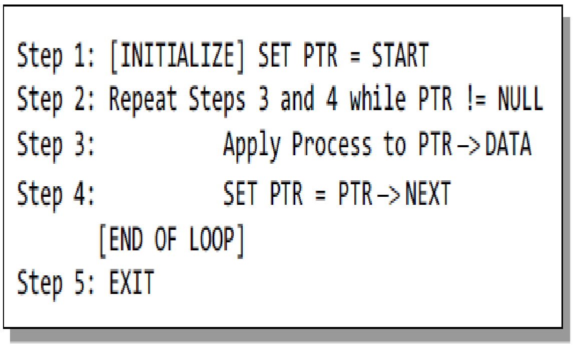
Traversing a linked list means accessing the nodes of the list in order to perform some processing on them. Remember a linked list always contains a pointer variable START which stores the address of the first node of the list. End of the list is marked by storing NULL or –1 in the NEXT field of the last node.

2.3 OPERATIONS

Traversing a Linked List:

For traversing the linked list, we also make use of another pointer variable PTR which points to the node that is currently being accessed. Algorithm for traversing a linked list



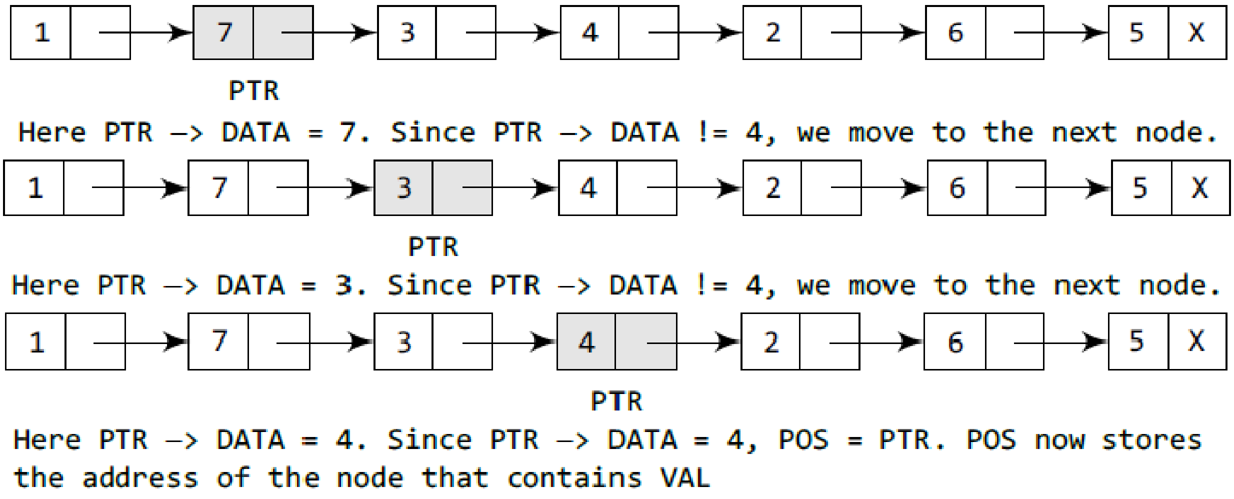


```
Step 1: [INITIALIZE] SET PTR = START
Step 2: Repeat Steps 3 and 4 while PTR != NULL
Step 3:         Apply Process to PTR->DATA
Step 4:         SET PTR = PTR->NEXT
        [END OF LOOP]
Step 5: EXIT
```

Searching for a Value in a Linked List:

Searching a linked list means to find a particular element in the linked list. So searching means finding whether a given value is present in the information part of the node or not. If it is present, the algorithm returns the address of the node that contains the value. However, if the search is unsuccessful, POS is set to NULL which indicates that VAL is not present in the linked list.

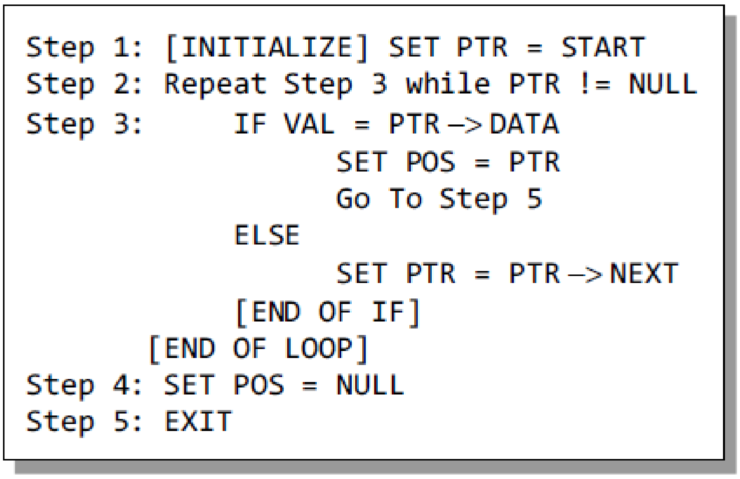
Consider the linked list shown in below. If we have VAL = 4, then the flow of the algorithm can be explained as shown in the figure.



Here PTR -> DATA = 7. Since PTR -> DATA != 4, we move to the next node.

Here PTR -> DATA = 3. Since PTR -> DATA != 4, we move to the next node.

Here PTR -> DATA = 4. Since PTR -> DATA = 4, POS = PTR. POS now stores the address of the node that contains VAL



```
Step 1: [INITIALIZE] SET PTR = START
Step 2: Repeat Step 3 while PTR != NULL
Step 3:      IF VAL = PTR->DATA
                   SET POS = PTR
                   Go To Step 5
             ELSE
                   SET PTR = PTR->NEXT
             [END OF IF]
        [END OF LOOP]
Step 4: SET POS = NULL
Step 5: EXIT
```

Inserting a New Node in a Linked List:

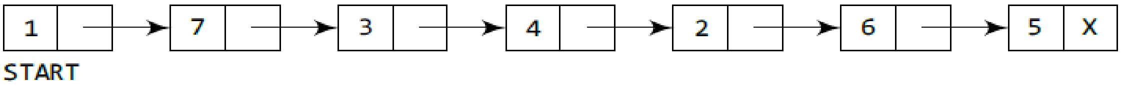
we will see how a new node is added into an already existing linked list. We will take four cases and then see how insertion is done in each case.

Case 1: The new node is inserted at the beginning.Case
2: The new node is inserted at the end.
Case 3: The new node is inserted after a given node.
Case 4: The new node is inserted before a given node.

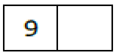
Let us first discuss an important term called OVERFLOW. Overflow is a condition that occurs when AVAIL = NULL or no free memory cell is present in the system. When this condition occurs, the program must give an appropriate message.

Case 1: Inserting a Node at the Beginning of a Linked List

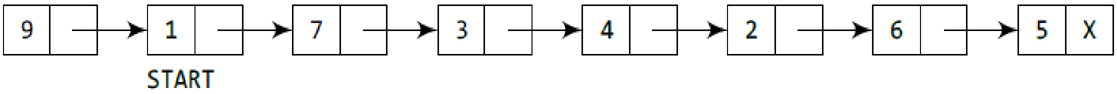
Inserting a Node at the Beginning of a Linked List. Consider the linked list shown in below figure. Suppose we want to add a new node with data 9 and add it as the first node of the list.



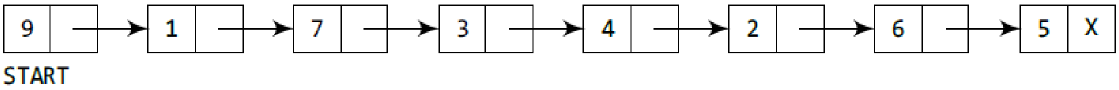
Allocate memory for the new node and initialize its DATA part to 9.

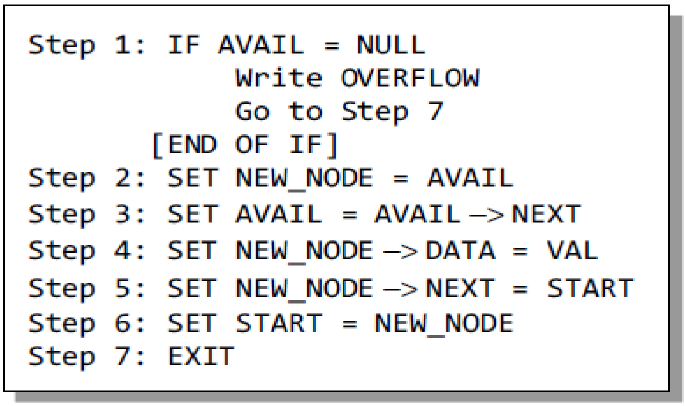


Add the new node as the first node of the list by making the NEXT part of the new node contain the address of START.



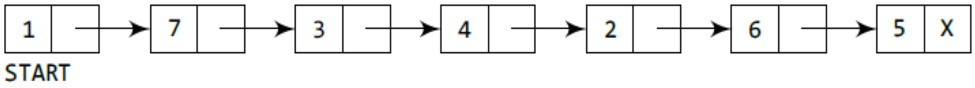
Now make START to point to the first node of the list.



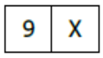


```
Step 1: IF AVAIL = NULL
                Write OVERFLOW
                Go to Step 7
          [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET NEW_NODE -> NEXT = START
Step 6: SET START = NEW_NODE
Step 7: EXIT
```

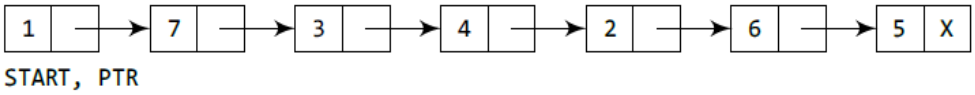
Case 2: Inserting a Node at the End of a Linked List



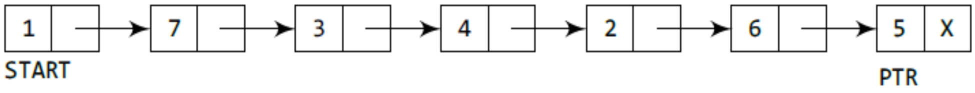
Allocate memory for the new node and initialize its DATA part to 9 and NEXT part to NULL.



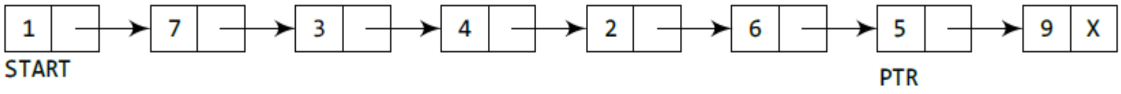
Take a pointer variable PTR which points to START.

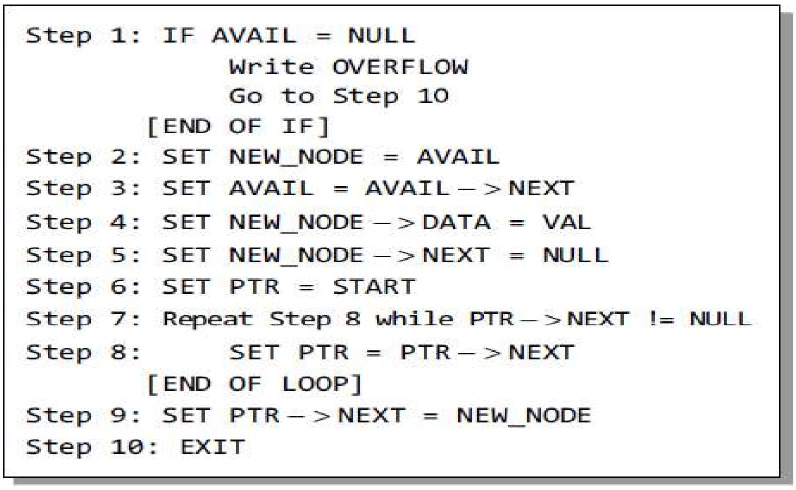


Move PTR so that it points to the last node of the list.



Add the new node after the node pointed by PTR. This is done by storing the address of the new node in the NEXT part of PTR.

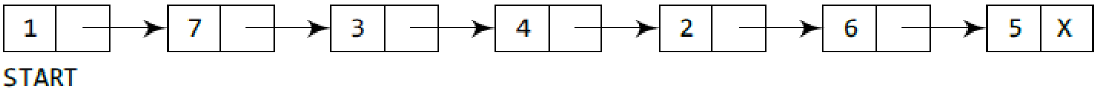




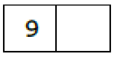
```
Step 1: IF AVAIL = NULL
                Write OVERFLOW
                Go to Step 10
        [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL->NEXT
Step 4: SET NEW_NODE->DATA = VAL
Step 5: SET NEW_NODE->NEXT = NULL
Step 6: SET PTR = START
Step 7: Repeat Step 8 while PTR->NEXT != NULL
Step 8:         SET PTR = PTR->NEXT
        [END OF LOOP]
Step 9: SET PTR->NEXT = NEW_NODE
Step 10: EXIT
```

Case 3: Inserting a Node After a Given Node in a Linked List

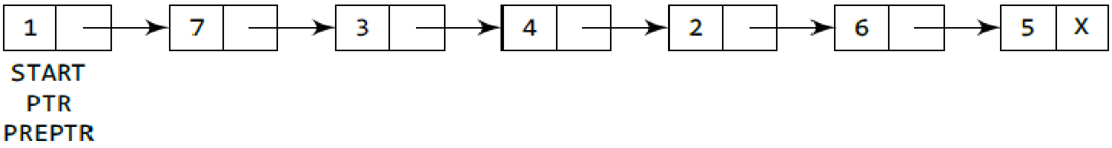
Consider the linked list shown in below figure. Suppose we want to add a new node with value 9 after the node containing 3.



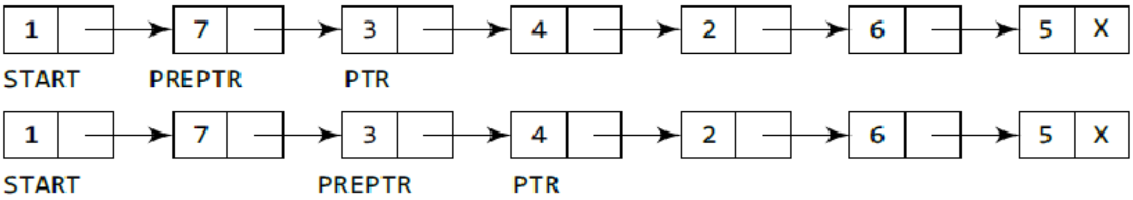
Allocate memory for the new node and initialize its DATA part to 9.



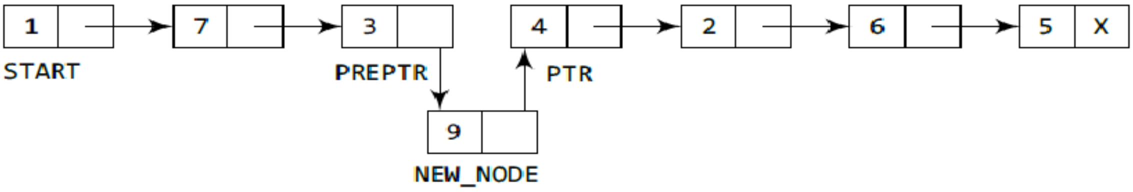
Take two pointer variables PTR and PREPTR and initialize them with START so that START, PTR, and PREPTR point to the first node of the list.

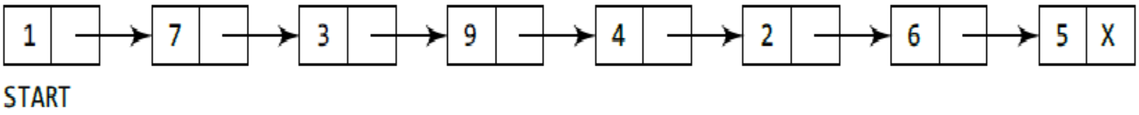


Move PTR and PREPTR until the DATA part of PREPTR = value of the node after which insertion has to be done. PREPTR will always point to the node just before PTR.



Add the new node in between the nodes pointed by PREPTR and PTR.

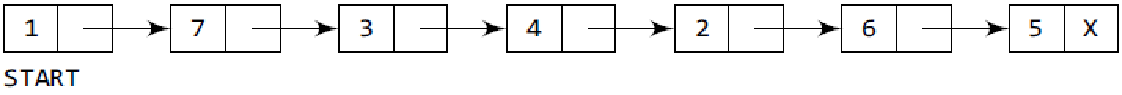




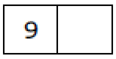
```
Step 1: IF AVAIL = NULL
                Write OVERFLOW
                Go to Step 10
            [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET NEW_NODE -> NEXT = NULL
Step 6: SET PTR = START
Step 7: Repeat Step 8 while PTR -> NEXT != NULL
Step 8:         SET PTR = PTR -> NEXT
            [END OF LOOP]
Step 9: SET PTR -> NEXT = NEW_NODE
Step 10: EXIT
```

Case 4: Inserting a Node Before a Given Node in a Linked List

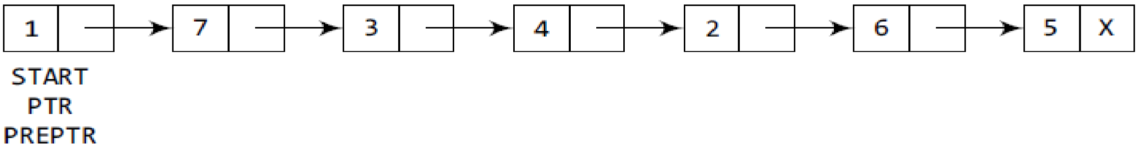
Consider the linked list shown in below figure. Suppose we want to add a new node with value 9 before the node containing 3.



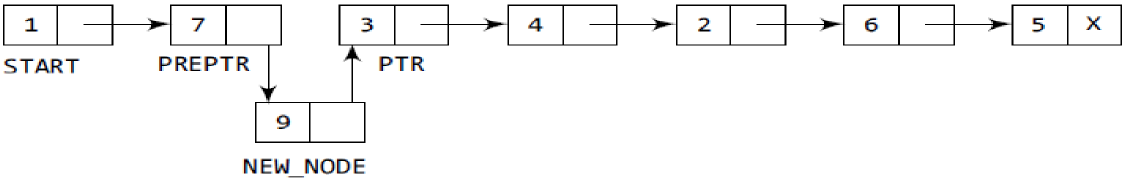
Allocate memory for the new node and initialize its DATA part to 9.

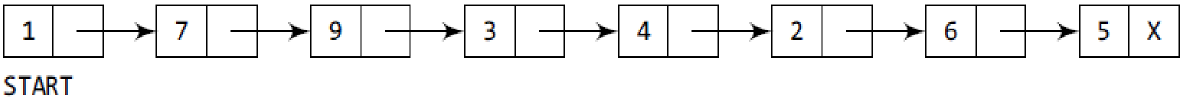


Initialize PREPTR and PTR to the START node.

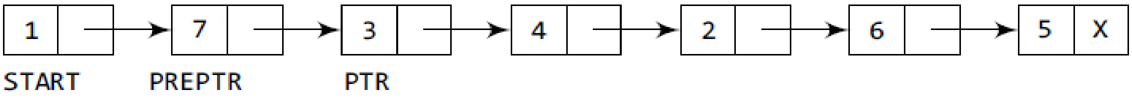


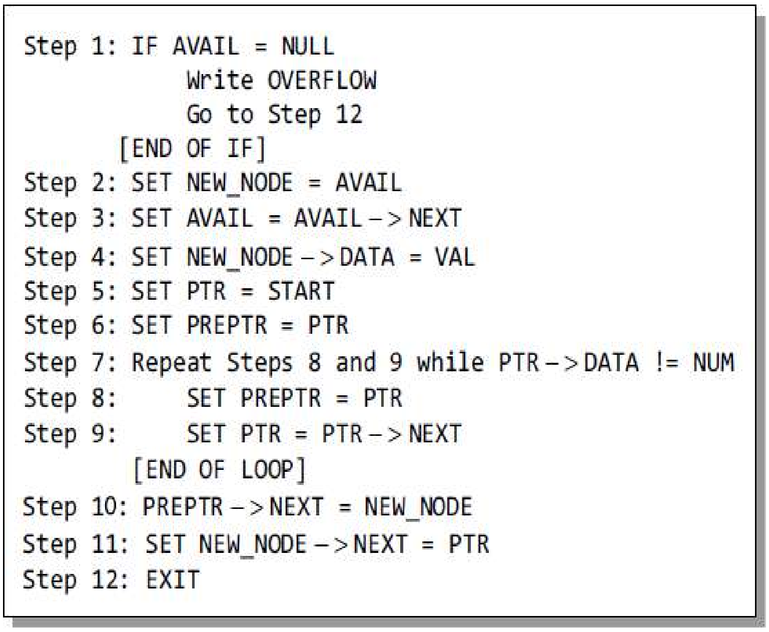
Insert the new node in between the nodes pointed by PREPTR and PTR.





Move PTR and PREPTR until the DATA part of PTR = value of the node before which insertion has to be done. PREPTR will always point to the node just before PTR.





```
Step 1: IF AVAIL = NULL
            Write OVERFLOW
            Go to Step 12
       [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL->NEXT
Step 4: SET NEW_NODE->DATA = VAL
Step 5: SET PTR = START
Step 6: SET PREPTR = PTR
Step 7: Repeat Steps 8 and 9 while PTR->DATA != NUM
Step 8:     SET PREPTR = PTR
Step 9:     SET PTR = PTR->NEXT
        [END OF LOOP]
Step 10: PREPTR->NEXT = NEW_NODE
Step 11: SET NEW_NODE->NEXT = PTR
Step 12: EXIT
```

Deleting a Node from a Linked List:

We will discuss how a node is deleted from an already existing linked list. We will consider three casesand then see how deletion is done in each case.

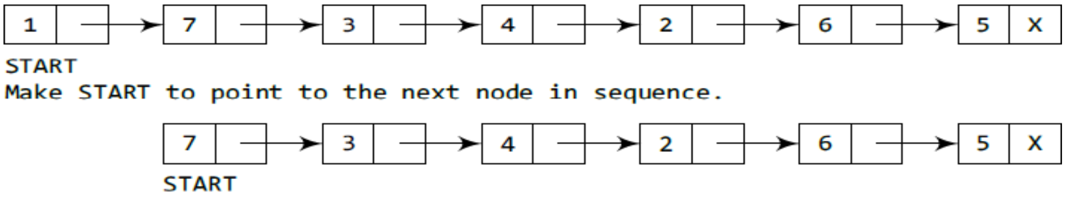
Case 1: The first node is

deleted.Case 2: The last node

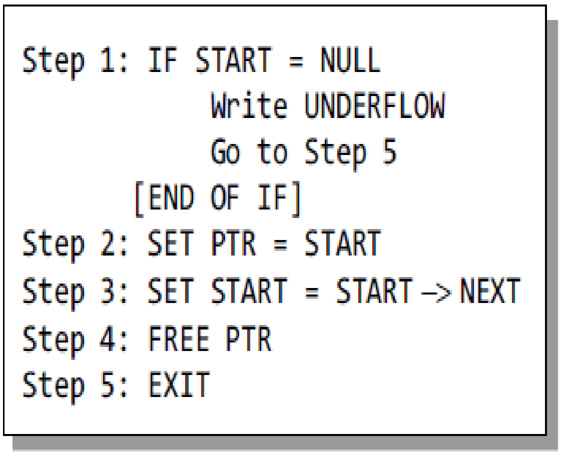
is deleted.

Case 3: The node after a given node is deleted.

Case 1: Deleting a First Node from a Linked List

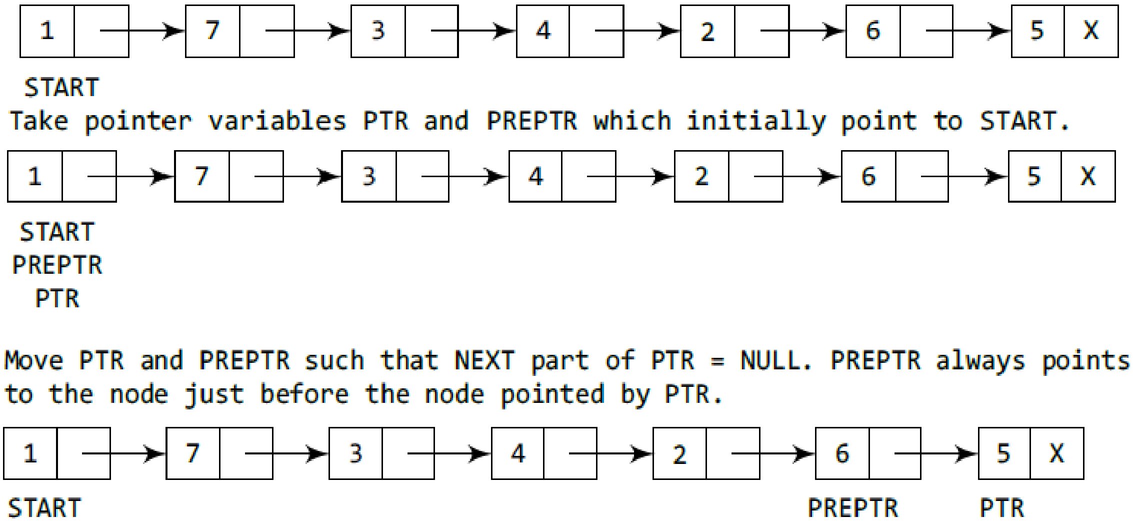


Make START to point to the next node in sequence.



```
Step 1: IF START = NULL
                Write UNDERFLOW
                Go to Step 5
          [END OF IF]
Step 2: SET PTR = START
Step 3: SET START = START -> NEXT
Step 4: FREE PTR
Step 5: EXIT
```

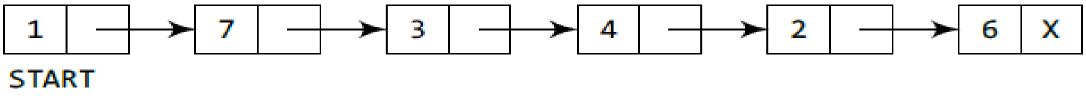
Case 2: Deleting the Last Node from a Linked List



Take pointer variables PTR and PREPTR which initially point to START.

Move PTR and PREPTR such that NEXT part of PTR = NULL. PREPTR always points to the node just before the node pointed by PTR.

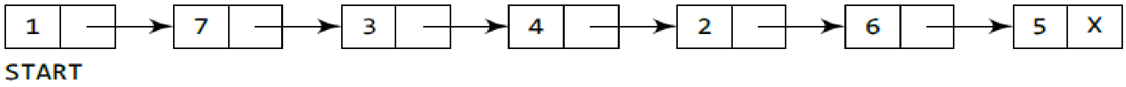
Set the NEXT part of PREPTR node to NULL.



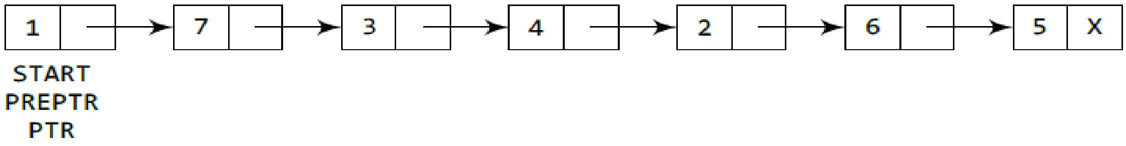
```
Step 1: IF START = NULL
              Write UNDERFLOW
              Go to Step 8
         [END OF IF]
Step 2: SET PTR = START
Step 3: Repeat Steps 4 and 5 while PTR->NEXT != NULL
Step 4:       SET PREPTR = PTR
Step 5:       SET PTR = PTR->NEXT
         [END OF LOOP]
Step 6: SET PREPTR->NEXT = NULL
Step 7: FREE PTR
Step 8: EXIT
```

Case 3: Deleting After a Given Node in a Linked List

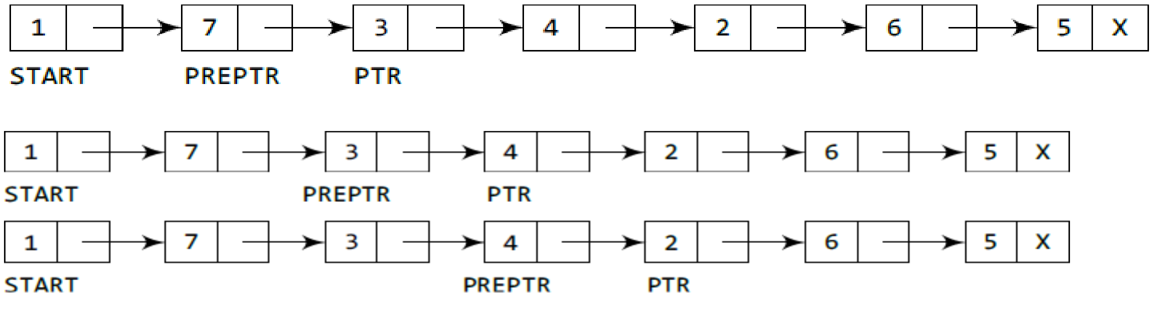
Consider the linked list shown in below figure. Suppose we want to delete the node that succeeds the node which contains data value 4.



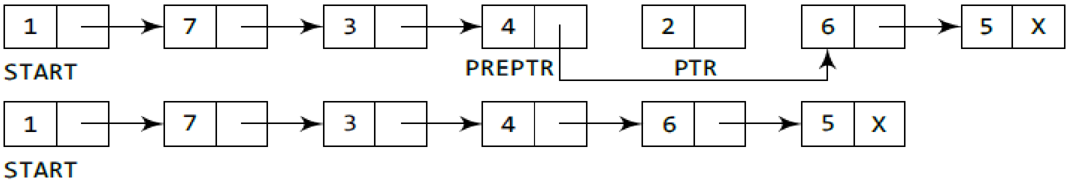
Take pointer variables PTR and PREPTR which initially point to START.

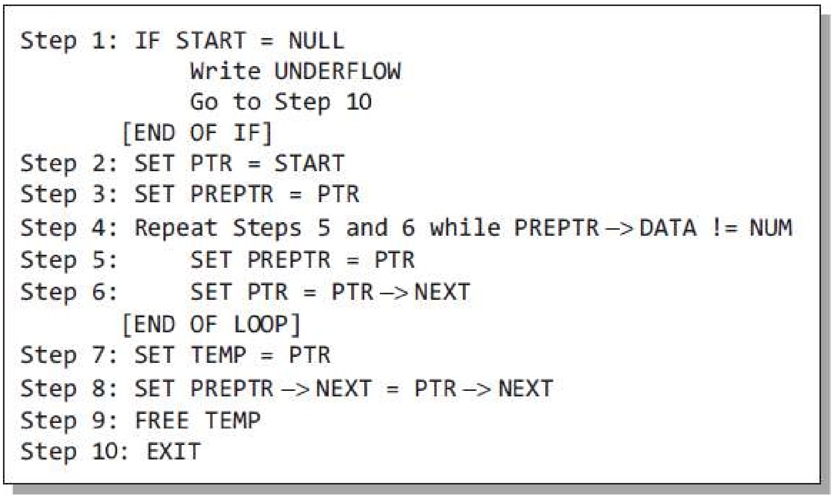


Move PREPTR and PTR such that PREPTR points to the node containing VAL and PTR points to the succeeding node.



Set the NEXT part of PREPTR to the NEXT part of PTR.





```
Step 1: IF START = NULL
            Write UNDERFLOW
            Go to Step 10
        [END OF IF]
Step 2: SET PTR = START
Step 3: SET PREPTR = PTR
Step 4: Repeat Steps 5 and 6 while PREPTR -> DATA != NUM
Step 5:     SET PREPTR = PTR
Step 6:     SET PTR = PTR -> NEXT
        [END OF LOOP]
Step 7: SET TEMP = PTR
Step 8: SET PREPTR -> NEXT = PTR -> NEXT
Step 9: FREE TEMP
Step 10: EXIT
```

2.4 APPLICATIONS ON SINGLE LINKED LIST

- ✓ Implementation of stacks and queues.
- ✓ Implementation of graphs: Adjacency list representation of graphs is most popular which useslinked list to store adjacent vertices.
- ✓ Dynamic memory allocation: We use linked list of free blocks.
- ✓ Maintaining directory of names. Performing arithmetic operations on long integers
- ✓ Manipulation of polynomials by storing constants in the node of linked list. Representing sparsematrices

Advantages of Single Linked list:

- Insertions and Deletions can be done easily.
- It does not need movement of elements for insertion and deletion.
- Space is not wasted as we can get space according to our requirements.
- Its size is not fixed. It can be extended or reduced according to requirements.
- Elements may or may not be stored in consecutive memory available, even then we can storethe data in computer.
- It is less expensive.

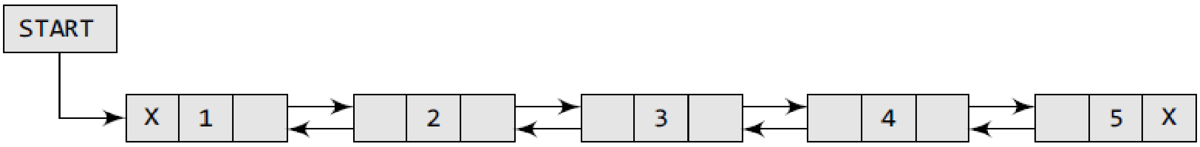
Disadvantages of Single Linked list:

- It requires more space as pointers are also stored with information.
- Different amount of time is required to access each element.
- If we have to go to a particular element then we have to go through all those elements thatcome before that element.
- We cannot traverse it from last & only from the beginning.
- It is not easy to sort the elements stored in the linear linked list.

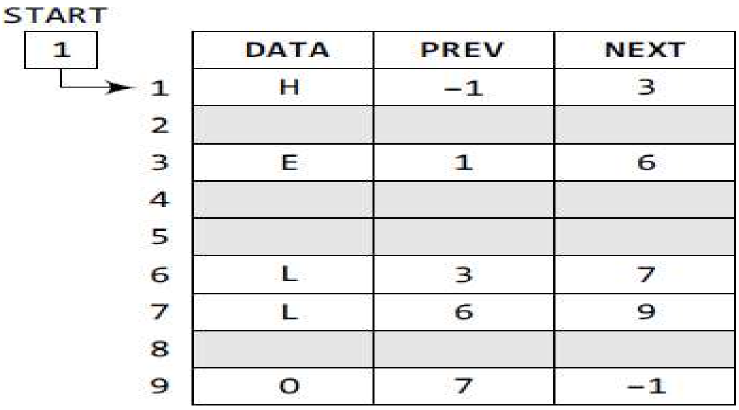
2.5 DOUBLY LINKED LIST

A doubly linked list or a two-way linked list is a more complex type of linked list which contains apointer to the next as well as the previous node in the sequence.

Therefore, it consists of three parts—data, a pointer to the next node, and a pointer to the previous node.



A doubly linked list provides the ease to manipulate the elements of the list as it maintains pointers to nodes in both the directions (forward and backward).
The main advantage of using a doubly linked list is that it makes searching twice as efficient.Let us view how a doubly linked list is maintained in the memory.



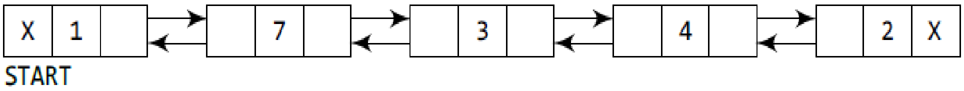
	DATA	PREV	NEXT
1	H	−1	3
2			
3	E	1	6
4			
5			
6	L	3	7
7	L	6	9
8			
9	O	7	−1

Inserting a New Node in a Doubly Linked List:

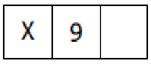
In this section, we will discuss how a new node is added into an already existing doubly linked list. Wewill take four cases and then see how insertion is done in each case.

Case 1: The new node is inserted at the beginning.Case 2: The new node is inserted at the end.
Case 3: The new node is inserted after a given node.
Case 4: The new node is inserted before a given node.

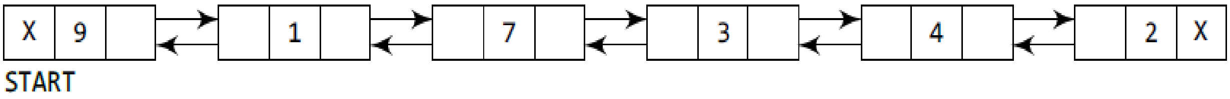
Case 1: Inserting a Node at the Beginning of a Doubly Linked List



Allocate memory for the new node and initialize its DATA part to 9 and PREV field to NULL.

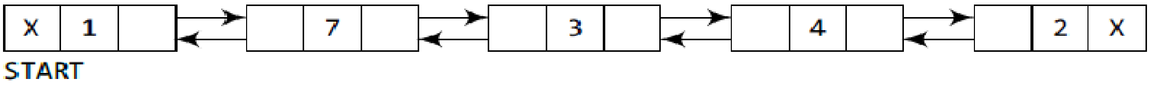


Add the new node before the START node. Now the new node becomes the first node of the list.

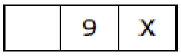


```
Step 1: IF AVAIL = NULL
                Write OVERFLOW
                Go to Step 9
        [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET NEW_NODE -> PREV = NULL
Step 6: SET NEW_NODE -> NEXT = START
Step 7: SET START -> PREV = NEW_NODE
Step 8: SET START = NEW_NODE
Step 9: EXIT
```

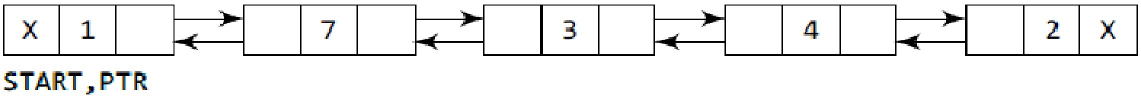
Case 2: Inserting a Node at the end of a Doubly Linked List



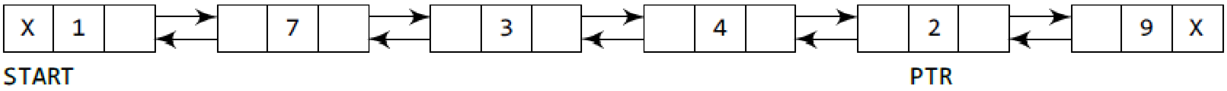
Allocate memory for the new node and initialize its DATA part to 9 and its NEXT field to NULL.



Take a pointer variable PTR and make it point to the first node of the list.

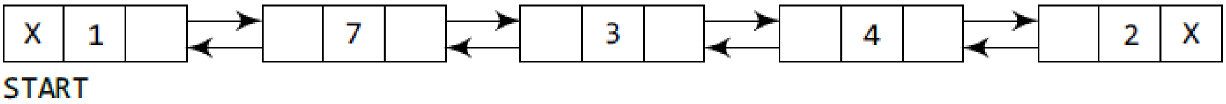


Move PTR so that it points to the last node of the list. Add the new node after the node pointed by PTR.

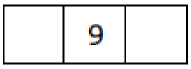


```
Step 1: IF AVAIL = NULL
                Write OVERFLOW
                Go to Step 11
        [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET NEW_NODE -> NEXT = NULL
Step 6: SET PTR = START
Step 7: Repeat Step 8 while PTR -> NEXT != NULL
Step 8:         SET PTR = PTR -> NEXT
        [END OF LOOP]
Step 9: SET PTR -> NEXT = NEW_NODE
Step 10: SET NEW_NODE -> PREV = PTR
Step 11: EXIT
```

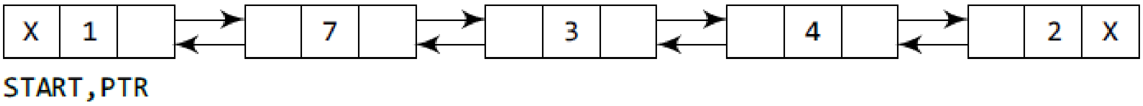
Case 3: Inserting a Node After a Given Node in a Doubly Linked List



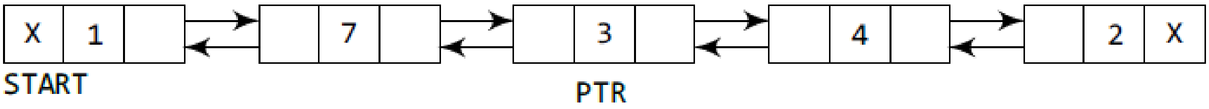
Allocate memory for the new node and initialize its DATA part to 9.



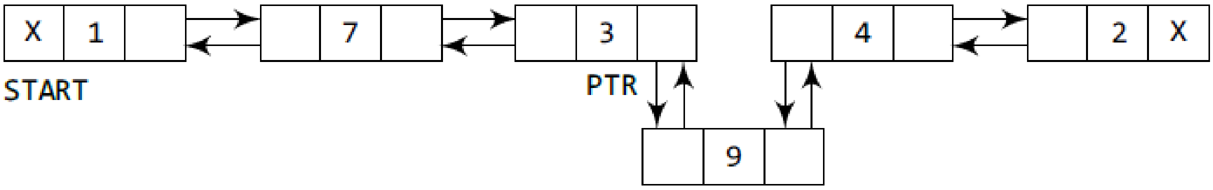
Take a pointer variable PTR and make it point to the first node of the list.

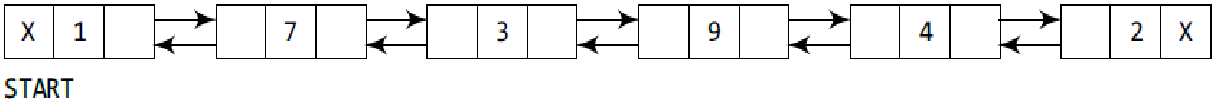


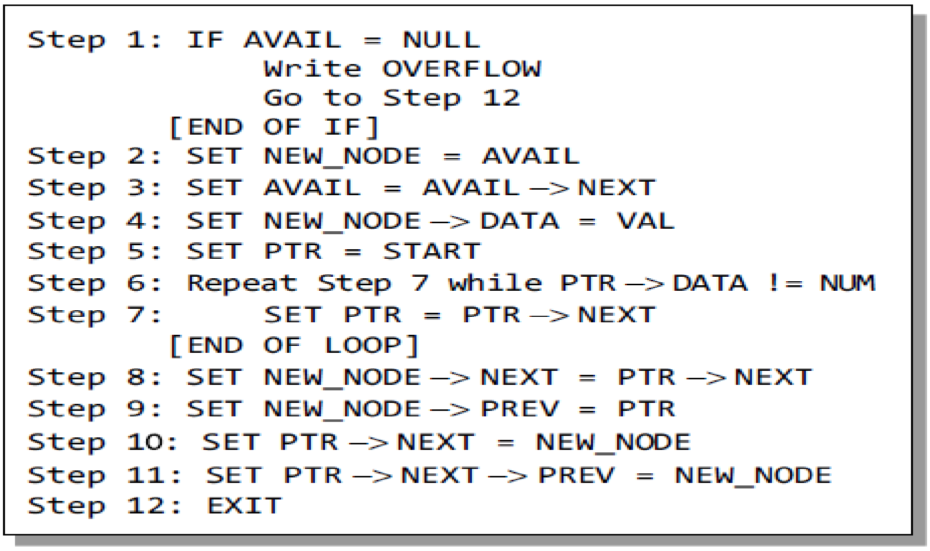
Move PTR further until the data part of PTR = value after which the node has to be inserted.



Insert the new node between PTR and the node succeeding it.

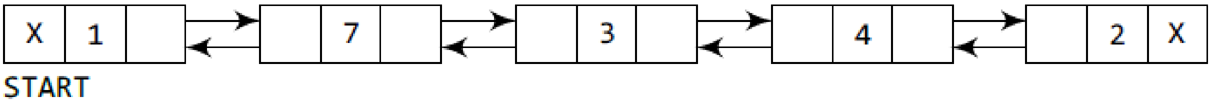




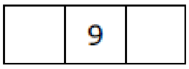


```
Step 1: IF AVAIL = NULL
              Write OVERFLOW
              Go to Step 12
        [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET PTR = START
Step 6: Repeat Step 7 while PTR -> DATA != NUM
Step 7:       SET PTR = PTR -> NEXT
        [END OF LOOP]
Step 8: SET NEW_NODE -> NEXT = PTR -> NEXT
Step 9: SET NEW_NODE -> PREV = PTR
Step 10: SET PTR -> NEXT = NEW_NODE
Step 11: SET PTR -> NEXT -> PREV = NEW_NODE
Step 12: EXIT
```

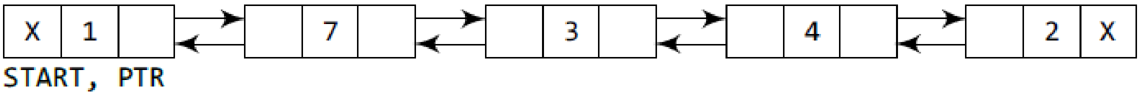
Case 4: Inserting a Node Before a Given Node in a Doubly Linked List



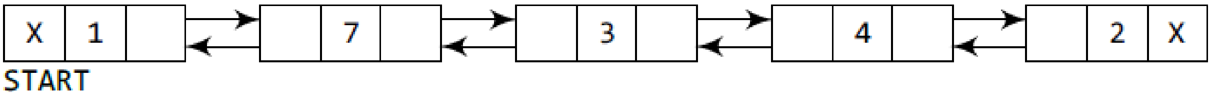
Allocate memory for the new node and initialize its DATA part to 9.



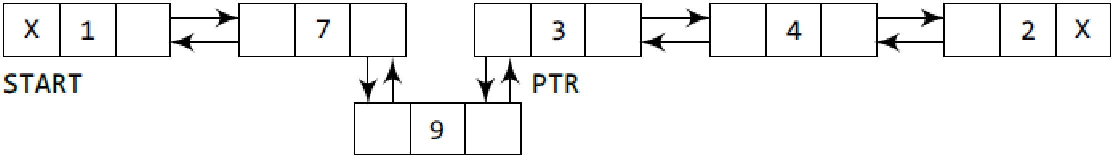
Take a pointer variable PTR and make it point to the first node of the list.

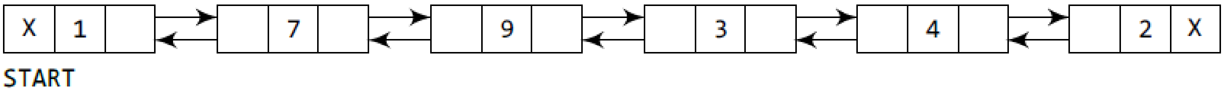


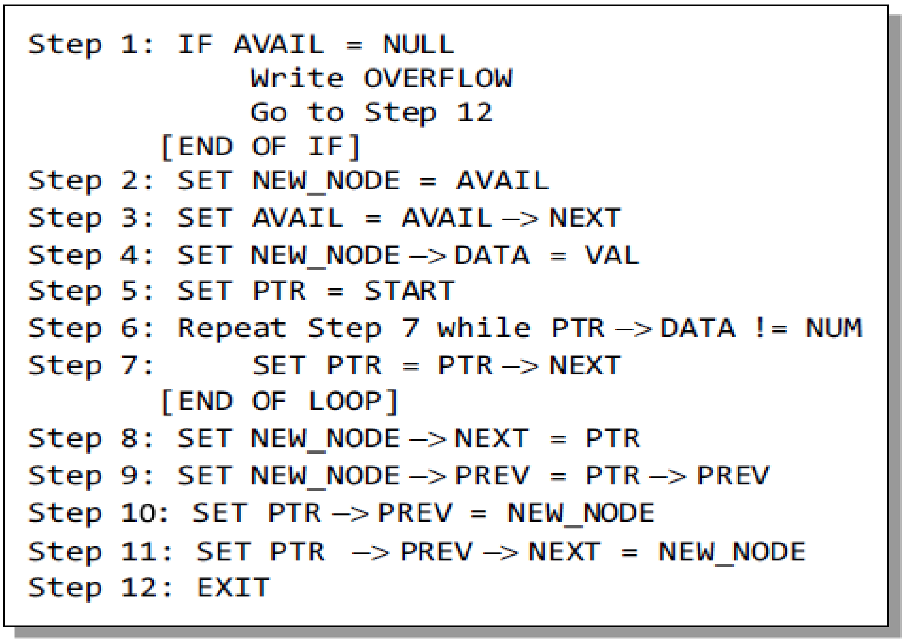
Move PTR further so that it now points to the node whose data is equal to the value before which the node has to be inserted.



Add the new node in between the node pointed by PTR and the node preceding it.







```
Step 1: IF AVAIL = NULL
              Write OVERFLOW
              Go to Step 12
        [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET PTR = START
Step 6: Repeat Step 7 while PTR -> DATA != NUM
Step 7:       SET PTR = PTR -> NEXT
        [END OF LOOP]
Step 8: SET NEW_NODE -> NEXT = PTR
Step 9: SET NEW_NODE -> PREV = PTR -> PREV
Step 10: SET PTR -> PREV = NEW_NODE
Step 11: SET PTR  -> PREV -> NEXT = NEW_NODE
Step 12: EXIT
```

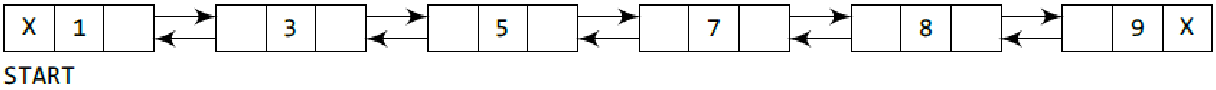
Deleting a Node from a Doubly Linked List

In this section, we will see how a node is deleted from an already existing doubly linked list. We will take four cases and then see how deletion is done in each case.

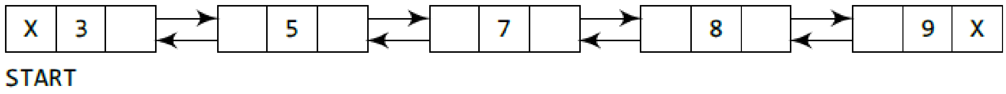
Case 1: The first node is deleted.Case 2: The last node is deleted.

Case 3: The node after a given node is deleted. Case 4: The node before a given node is deleted.

Case 1: Deleting the First Node from a Doubly Linked List

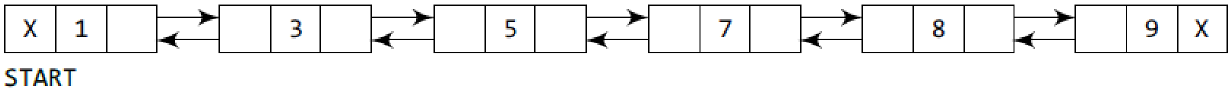


Free the memory occupied by the first node of the list and make the second node of the list as the START node.

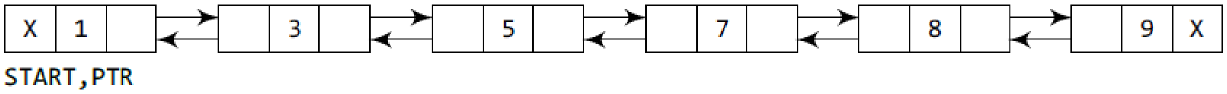


```
Step 1: IF START = NULL
                Write UNDERFLOW
                Go to Step 6
         [END OF IF]
Step 2: SET PTR = START
Step 3: SET START = START -> NEXT
Step 4: SET START -> PREV = NULL
Step 5: FREE PTR
Step 6: EXIT
```

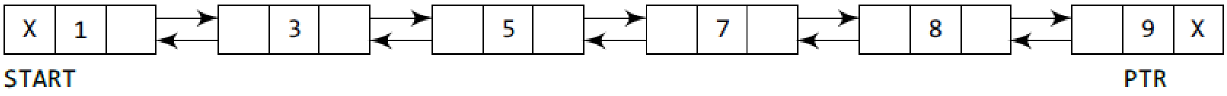
Case 2: Deleting the Last Node from a Doubly Linked List



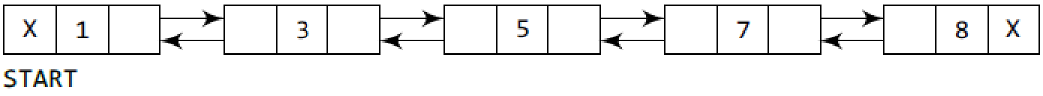
Take a pointer variable PTR that points to the first node of the list.



Move PTR so that it now points to the last node of the list.

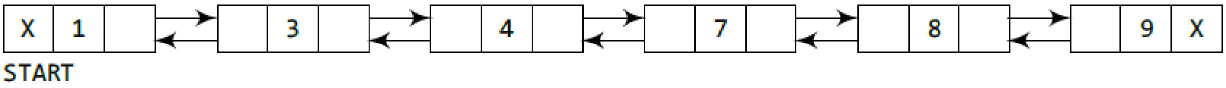


Free the space occupied by the node pointed by PTR and store NULL in NEXT field of its preceding node.

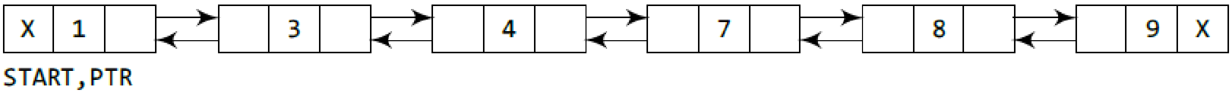


```
Step 1: IF START = NULL
              Write UNDERFLOW
              Go to Step 7
         [END OF IF]
Step 2: SET PTR = START
Step 3: Repeat Step 4 while PTR -> NEXT != NULL
Step 4:       SET PTR = PTR -> NEXT
         [END OF LOOP]
Step 5: SET PTR -> PREV -> NEXT = NULL
Step 6: FREE PTR
Step 7: EXIT
```

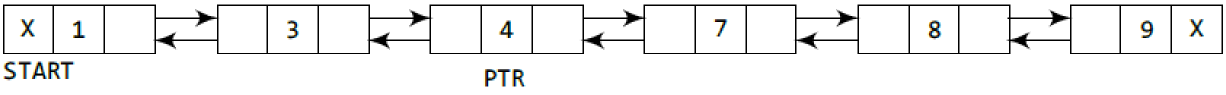
Case 3: Deleting the Node After a Given Node in a Doubly Linked List



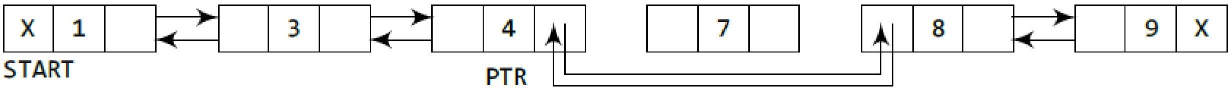
Take a pointer variable PTR and make it point to the first node of the list.

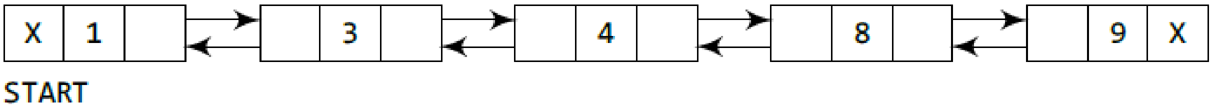


Move PTR further so that its data part is equal to the value after which the node has to be inserted.



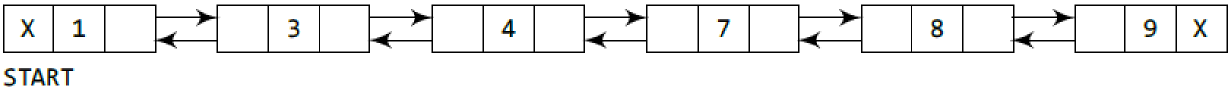
Delete the node succeeding PTR.



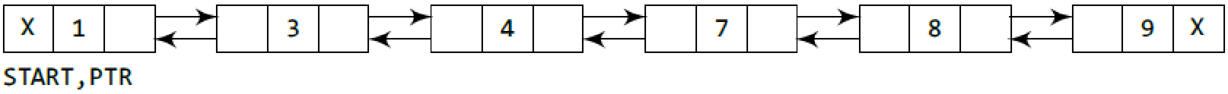


```
Step 1: IF START = NULL
              Write UNDERFLOW
              Go to Step 9
        [END OF IF]
Step 2: SET PTR = START
Step 3: Repeat Step 4 while PTR -> DATA != NUM
Step 4:        SET PTR = PTR -> NEXT
        [END OF LOOP]
Step 5: SET TEMP = PTR -> NEXT
Step 6: SET PTR -> NEXT = TEMP -> NEXT
Step 7: SET TEMP -> NEXT -> PREV = PTR
Step 8: FREE TEMP
Step 9: EXIT
```

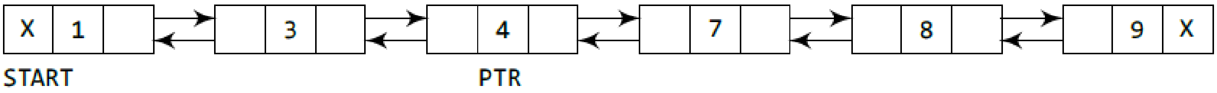
Case 4: Deleting the Node Before a Given Node in a Doubly Linked List



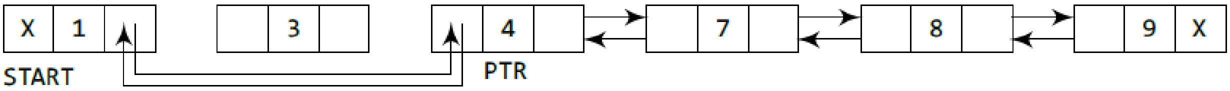
Take a pointer variable PTR that points to the first node of the list.

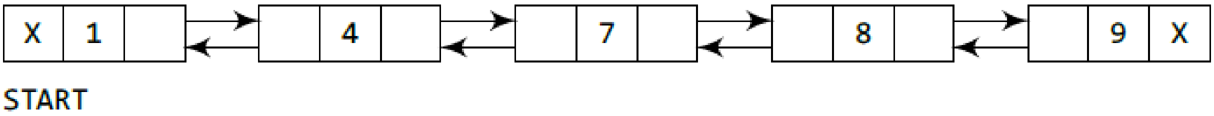


Move PTR further till its data part is equal to the value before which the node has to be deleted.



Delete the node preceding PTR.





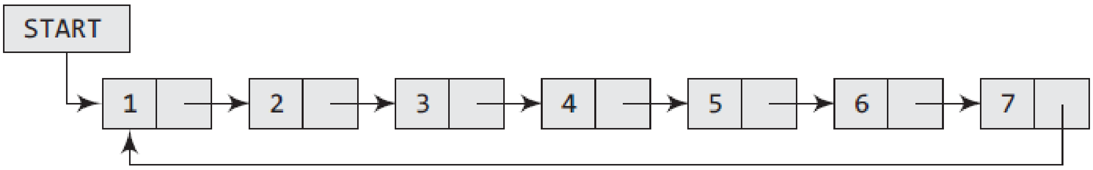
```
Step 1: IF START = NULL
              Write UNDERFLOW
              Go to Step 9
        [END OF IF]
Step 2: SET PTR = START
Step 3: Repeat Step 4 while PTR -> DATA != NUM
Step 4:       SET PTR = PTR -> NEXT
        [END OF LOOP]
Step 5: SET TEMP = PTR -> PREV
Step 6: SET TEMP -> PREV -> NEXT = PTR
Step 7: SET PTR -> PREV = TEMP -> PREV
Step 8: FREE TEMP
Step 9: EXIT
```

2.6 CIRCULAR LINKED LIST

In a circular linked list, the last node contains a pointer to the first node of the list. We can have a circular singly linked list as well as a circular doubly linked list.

While traversing a circular linked list, we can begin at any node and traverse the list in any direction,forward or backward, until we reach the same node where we started. Thus, a circular linked list hasno beginning and no ending.

Note that there are no NULL values in the NEXT part of any of the nodes of list.



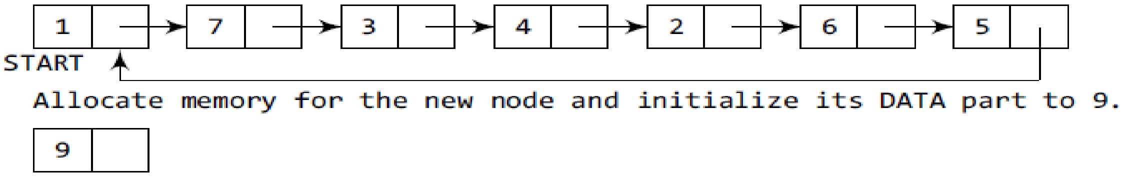
Operation:

Inserting a New Node in a Circular Linked List

In this section, we will see how a new node is added into an already existing linked list. We will taketwo cases and then see how insertion is done in each case.

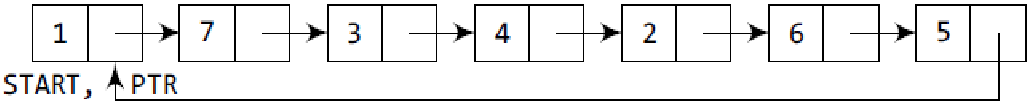
Case 1: The new node is inserted at the beginning of the circular linked list.Case 2: The new node is inserted at the end of the circular linked list.

Case 1: Inserting a Node at the Beginning of a Circular Linked List

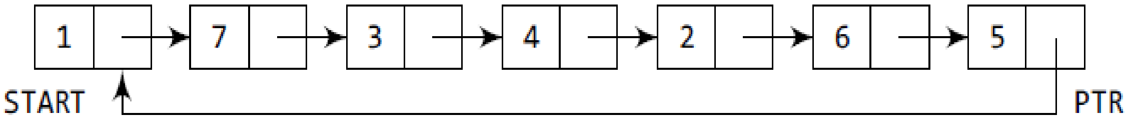


Allocate memory for the new node and initialize its DATA part to 9.

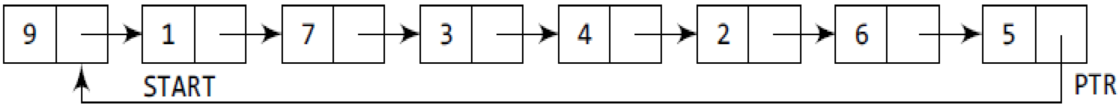
Take a pointer variable PTR that points to the START node of the list.

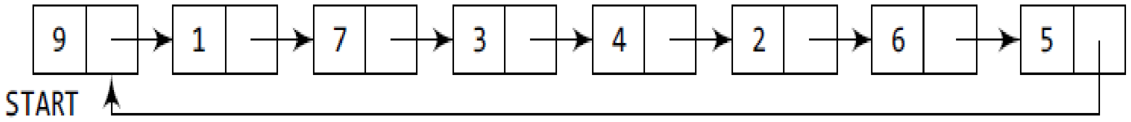


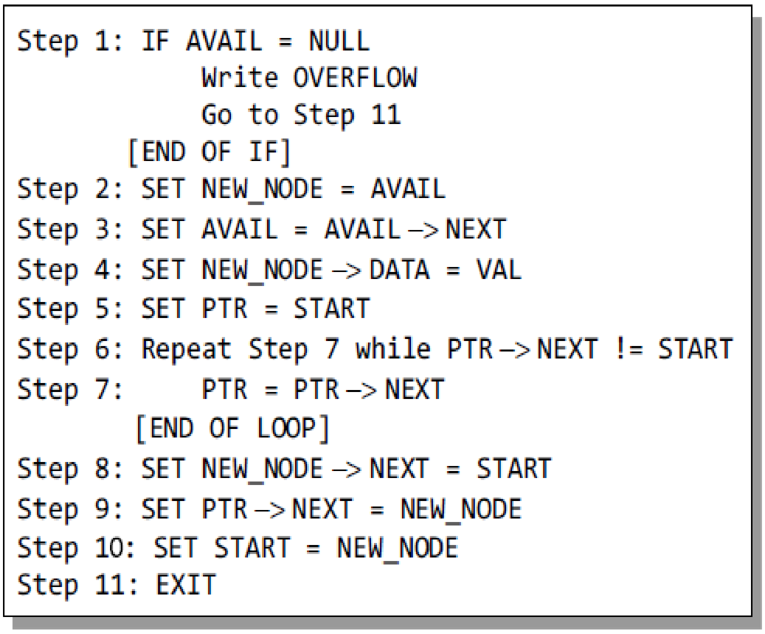
Move PTR so that it now points to the last node of the list.



Add the new node in between PTR and START.

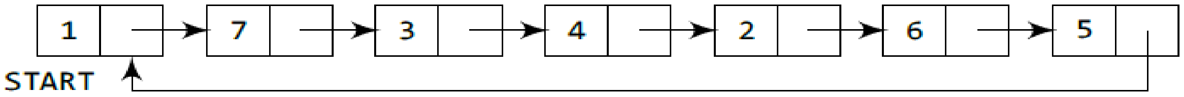




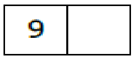


```
Step 1: IF AVAIL = NULL
                Write OVERFLOW
                Go to Step 11
        [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET PTR = START
Step 6: Repeat Step 7 while PTR -> NEXT != START
Step 7:         PTR = PTR -> NEXT
        [END OF LOOP]
Step 8: SET NEW_NODE -> NEXT = START
Step 9: SET PTR -> NEXT = NEW_NODE
Step 10: SET START = NEW_NODE
Step 11: EXIT
```

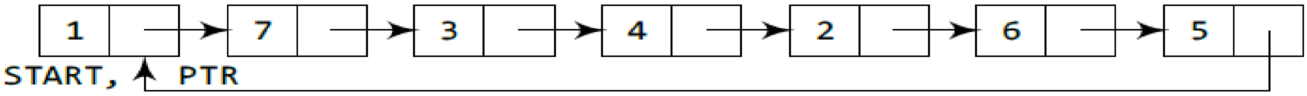
Case 2: Inserting a Node at the End of a Circular Linked List



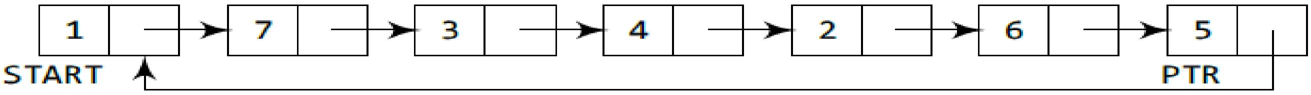
Allocate memory for the new node and initialize its DATA part to 9.



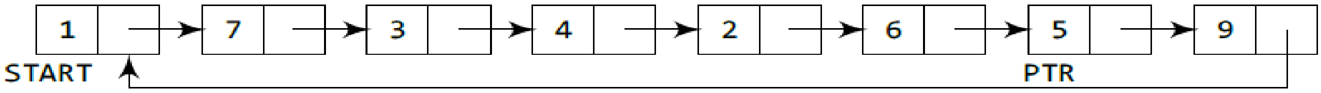
Take a pointer variable PTR which will initially point to START.

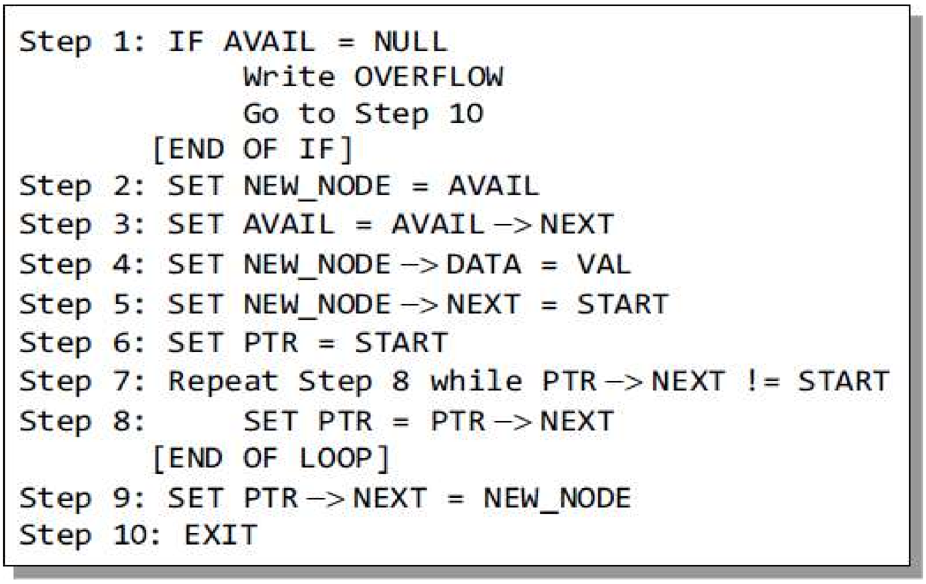


Move PTR so that it now points to the last node of the list.



Add the new node after the node pointed by PTR.





```
Step 1: IF AVAIL = NULL
            Write OVERFLOW
            Go to Step 10
        [END OF IF]
Step 2: SET NEW_NODE = AVAIL
Step 3: SET AVAIL = AVAIL -> NEXT
Step 4: SET NEW_NODE -> DATA = VAL
Step 5: SET NEW_NODE -> NEXT = START
Step 6: SET PTR = START
Step 7: Repeat Step 8 while PTR -> NEXT != START
Step 8:     SET PTR = PTR -> NEXT
        [END OF LOOP]
Step 9: SET PTR -> NEXT = NEW_NODE
Step 10: EXIT
```

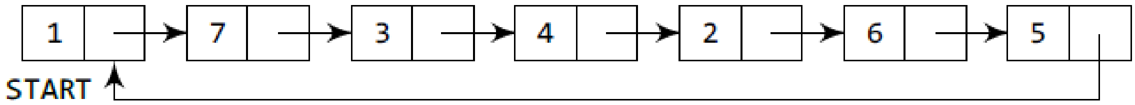
Deleting a Node from a Circular Linked List

In this section, we will discuss how a node is deleted from an already existing circular linked list. We will take two cases and then see how deletion is done in each case. Rest of the cases of deletion are same as that given for singly linked lists.

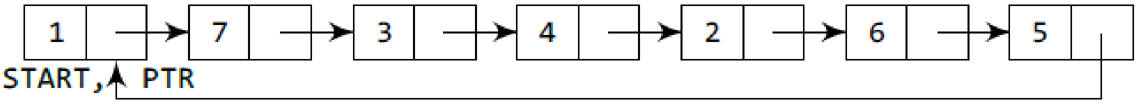
Case 1: The first node is

deleted.Case 2: The last node
is deleted.

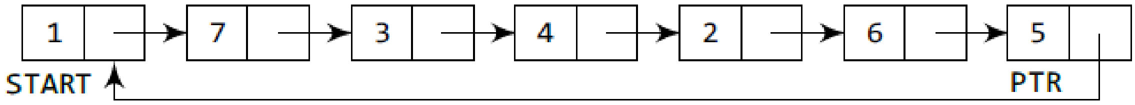
Case 1: Deleting the First Node from a Circular Linked List



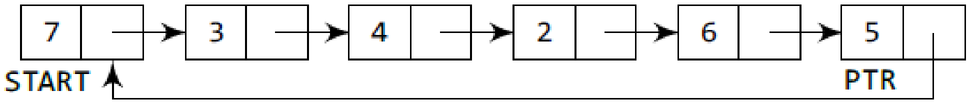
Take a variable PTR and make it point to the START node of the list.

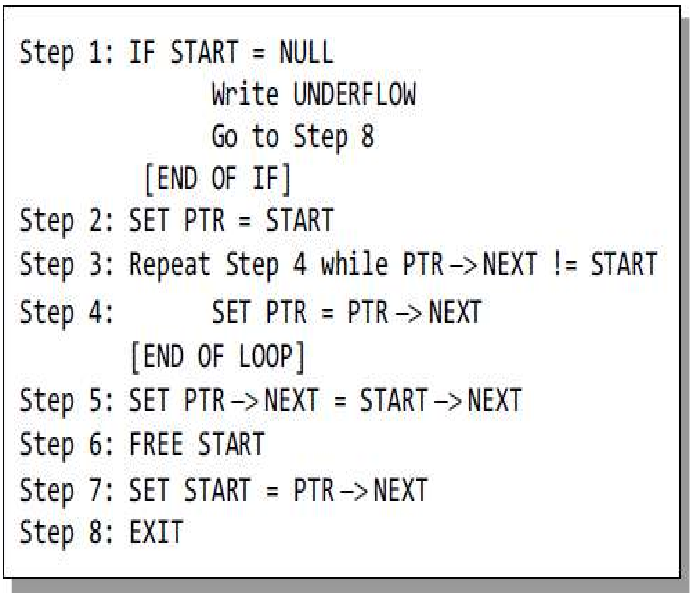


Move PTR further so that it now points to the last node of the list.



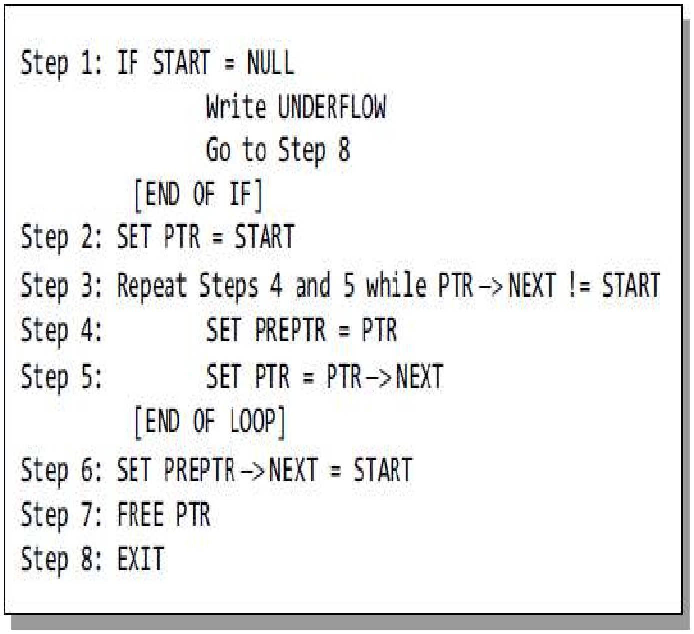
The NEXT part of PTR is made to point to the second node of the list and the memory of the first node is freed. The second node becomes the first node of the list.



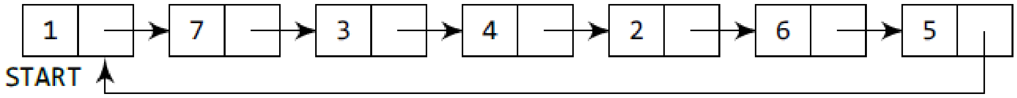


```
Step 1: IF START = NULL
              Write UNDERFLOW
              Go to Step 8
        [END OF IF]
Step 2: SET PTR = START
Step 3: Repeat Step 4 while PTR->NEXT != START
Step 4:       SET PTR = PTR->NEXT
        [END OF LOOP]
Step 5: SET PTR->NEXT = START->NEXT
Step 6: FREE START
Step 7: SET START = PTR->NEXT
Step 8: EXIT
```

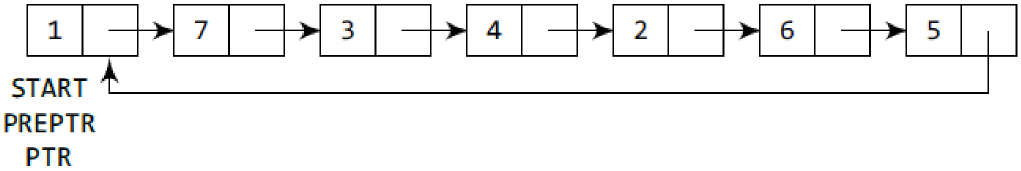
Case 2: Deleting the Last Node from a Circular Linked List



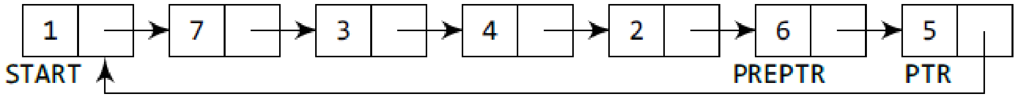
```
Step 1: IF START = NULL
                Write UNDERFLOW
                Go to Step 8
          [END OF IF]
Step 2: SET PTR = START
Step 3: Repeat Steps 4 and 5 while PTR->NEXT != START
Step 4:         SET PREPTR = PTR
Step 5:         SET PTR = PTR->NEXT
          [END OF LOOP]
Step 6: SET PREPTR->NEXT = START
Step 7: FREE PTR
Step 8: EXIT
```



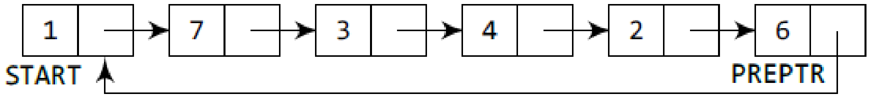
Take two pointers PREPTR and PTR which will initially point to START.



Move PTR so that it points to the last node of the list. PREPTR will always point to the node preceding PTR.



Make the PREPTR's next part store START node's address and free the space allocated for PTR. Now PREPTR is the last node of the list.



M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 3: QUEUE

3.1 INTRODUCTION

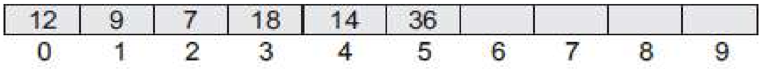
Let us explain the concept of queues using the analogies given below.

- People moving on an escalator. The people who got on the escalator first will be the first one to step out of it.
- People waiting for a bus. The first person standing in the line will be the first one to get into the bus.
- People standing outside the ticketing window of a cinema hall. The first person in the line will get the ticket first and thus will be the first one to move out of it.
- Luggage kept on conveyor belts. The bag which was placed first will be the first to come out at the other end.
- Cars lined at a toll bridge. The first car to reach the bridge will be the first to leave.

In all these examples, we see that the element at the first position is served first. Same is the case with queue data structure. A queue is a **FIFO (First-In, First-Out)** data structure in which the element that is inserted first is the first one to be taken out. The elements in a queue are added at one end called the REAR and removed from the other end called the FRONT. Queues can be implemented by using either arrays or linked lists. In this section, we will see how queues are implemented using each of these data structures.

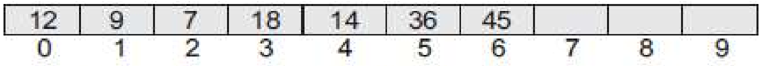
3.2 ARRAY REPRESENTATION OF QUEUES

Queues can be easily represented using linear arrays. As stated earlier, every queue has front and rear variables that point to the position from where deletions and insertions can be done, respectively.
The array representation of a queue is shown in Fig. 3.1.



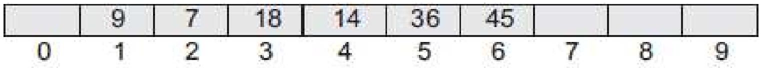
12	9	7	18	14	36				
0	1	2	3	4	5	6	7	8	9

Figure 3.1 Queue



12	9	7	18	14	36	45			
0	1	2	3	4	5	6	7	8	9

Figure 3.2 Queue after insertion of a new element



	9	7	18	14	36	45			
0	1	2	3	4	5	6	7	8	9

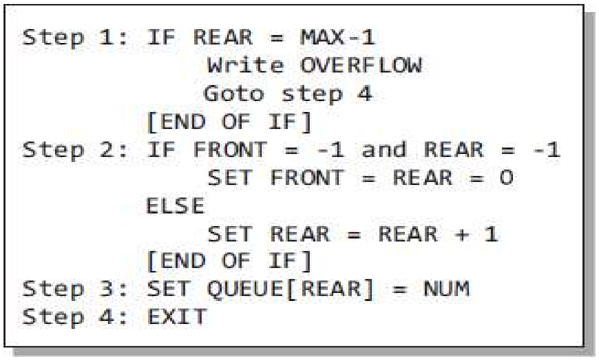
Operations on Queues

In Fig. 3.1, FRONT = 0 and REAR = 5. Suppose we want to add another element with value 45, then REAR would be incremented by 1 and the value would be stored at the position pointed by REAR.

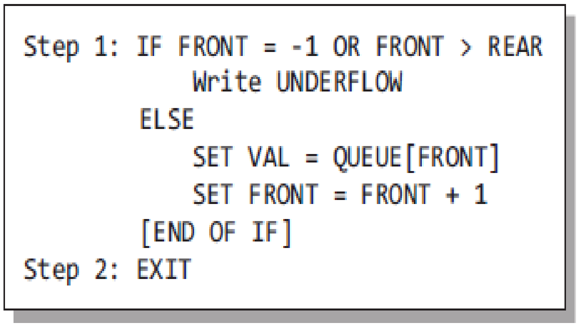
The queue after addition would be as shown in Fig. 3.2. Here, FRONT = 0 and REAR = 6. Every time a new element has to be added, we repeat the same procedure.

If we want to delete an element from the queue, then the value of FRONT will be incremented. Deletions are done from only this end of the queue. The queue after deletion will be as shown in Fig. 3.3.
Here, FRONT = 1 and REAR = 6.

However, before inserting an element in a queue, we must check for overflow conditions. An overflow will occur when we try to insert an element into a queue that is already full. When REAR = MAX – 1, where MAX is the size of the queue, we have an overflow condition. Note that we have written MAX – 1 because the index starts from 0. Similarly, before deleting an element from a queue, we must check for underflow conditions. An underflow condition occurs when we try to delete an element from a queue that is already empty. If FRONT = –1 and REAR = –1, it means there is no element in the queue.



```
Step 1: IF REAR = MAX-1
                Write OVERFLOW
                Goto step 4
            [END OF IF]
Step 2: IF FRONT = -1 and REAR = -1
                SET FRONT = REAR = 0
            ELSE
                SET REAR = REAR + 1
            [END OF IF]
Step 3: SET QUEUE[REAR] = NUM
Step 4: EXIT
```



```
Step 1: IF FRONT = -1 OR FRONT > REAR
                Write UNDERFLOW
            ELSE
                SET VAL = QUEUE[FRONT]
                SET FRONT = FRONT + 1
            [END OF IF]
Step 2: EXIT
```

Algorithm to insert an element in a queue

Algorithm to delete an element from a queue

NOTE: The process of inserting an element in the queue is called enqueue, and the process of deleting an element from the queue is called dequeue.

3.3 LINKED REPRESENTATION OF QUEUES

We have seen how a queue is created using an array. Although this technique of creating a queue is easy, its drawback is that the array must be declared to have some fixed size. If we allocate space for 50 elements in the queue and it hardly uses 20–25 locations, then half of the space will be wasted.

And in case we allocate less memory locations for a queue that might end up growing large and large, then a lot of re-allocations will have to be done, thereby creating a lot of overhead and consuming a lot of time.

In case the queue is a very small one or its maximum size is known in advance, then the array implementation of the queue gives an efficient implementation. But if the array size cannot be determined in advance, the other alternative, i.e., the linked representation is used. The storage requirement of linked representation of a queue with n elements is O(n) and the typical time requirement for operations is O(1).

In a linked queue, every element has two parts, one that stores the data and another that stores the address of the next element. The START pointer of the linked list is used as FRONT. Here, we will also use another pointer called REAR, which will store the address of the last element in the queue.

All insertions will be done at the rear end and all the deletions will be done at the front end. If FRONT = REAR = NULL, then it indicates that the queue is empty. The linked representation of a queue is shown in Fig. 3.4.

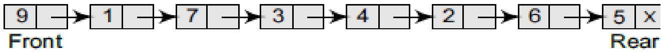


Figure 3.4 Linked queue

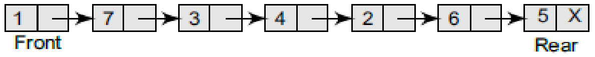
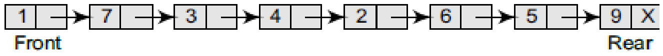
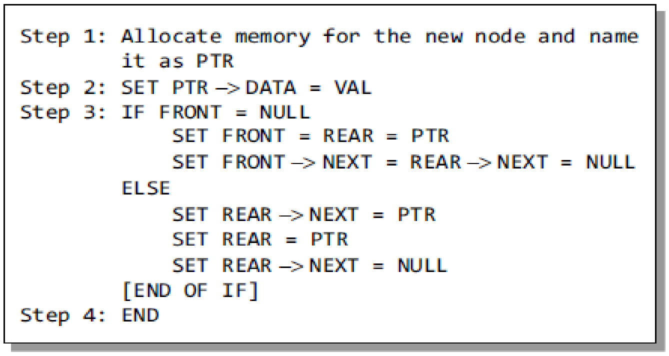


Figure 3.5 Linked queue





```
Step 1: Allocate memory for the new node and name
            it as PTR
Step 2: SET PTR -> DATA = VAL
Step 3: IF FRONT = NULL
                SET FRONT = REAR = PTR
                SET FRONT -> NEXT = REAR -> NEXT = NULL
            ELSE
                SET REAR -> NEXT = PTR
                SET REAR = PTR
                SET REAR -> NEXT = NULL
            [END OF IF]
Step 4: END
```

Algorithm to insert an element in a linked queue

Insert Operation

The insert operation is used to insert an element into a queue. The new element is added as the last element of the queue. Consider the linked queue shown in Fig. 3.5.

To insert an element with value 9, we first check if FRONT=NULL. If the condition holds, then the queue is empty. So, we allocate memory for a new node, store the value in its data part and NULL in its next part. The new node will then be called both FRONT and rear. However, if FRONT != NULL, then we will insert the new node at the rear end of the linked queue and name this new node as rear. Thus, the updated queue becomes as shown in Fig. 3.6.

The algorithm shows that inserting an element in a linked queue. In Step 1, the memory is allocated for the new node. In Step 2, the DATA part of the new node is initialized with the value to be stored in the node. In Step 3, we check if the new node is the first node of the linked queue. This is done by checking if FRONT = NULL. If this is the case, then the new node is tagged as FRONT as well as REAR. Also NULL is stored in the NEXT part of the node (which is also the FRONT and the REAR node). However, if the new node is not the first node in the list, then it is added at the REAR end of the linked queue (or the last node of the queue).

Delete Operation

The delete operation is used to delete the element that is first inserted in a queue, i.e., the element whose address is stored in FRONT. However, before deleting the value, we must first check if FRONT=NULL because if this is the case, then the queue is empty and no more deletions can be done. If an attempt is made todelete a value from a queue that is already empty, an underflow message is printed. Consider the queue shownin Fig. 3.7.

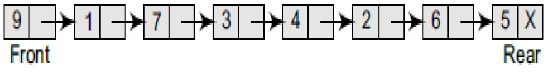
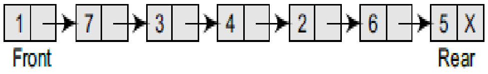
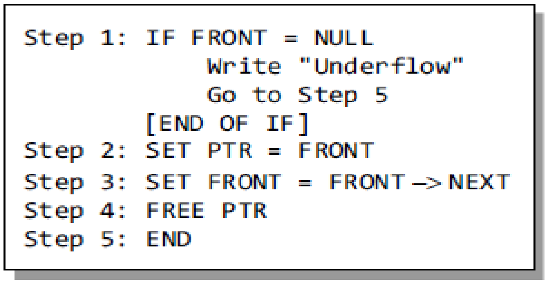


Figure 3.7 Linked queue





```
Step 1: IF FRONT = NULL
            Write "Underflow"
            Go to Step 5
        [END OF IF]
Step 2: SET PTR = FRONT
Step 3: SET FRONT = FRONT -> NEXT
Step 4: FREE PTR
Step 5: END
```

3.4 APPLICATIONS OF QUEUES

- Queues are widely used as waiting lists for a single shared resource like printer, disk, CPU.
- Queues are used to transfer data asynchronously (data not necessarily received at same rate as sent) between two processes (IO buffers), e.g., pipes, file IO, sockets.
- Queues are used as buffers on MP3 players and portable CD players, iPod playlist.
- Queues are used in Playlist for jukebox to add songs to the end, play from the front of the list.
- Queues are used in operating system for handling interrupts. When programming a real-time system that can be interrupted, for example, by a mouse click, it is necessary to process the interrupts immediately, before proceeding with the current job. If the interrupts have to be handled in the order ofarrival, then a FIFO queue is the appropriate data structure.

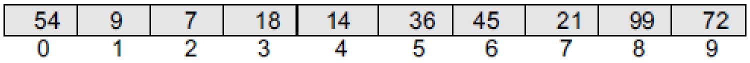
3.5 TYPES OF QUEUES

A queue data structure can be classified into the following types:

1. Circular Queue
2. Deque
3. Priority Queue
4. Multiple Queue

3.5.1 Circular Queues

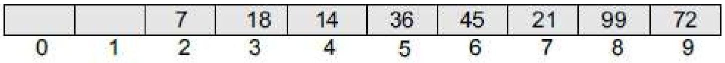
In linear queues, we have discussed so far that insertions can be done only at one end called the REAR and deletions are always done from the other end called the FRONT. Look at the queue shown in Fig. 3.9.



54	9	7	18	14	36	45	21	99	72
0	1	2	3	4	5	6	7	8	9

Here, FRONT = 0 and REAR = 9.

Now, if you want to insert another value, it will not be possible because the queue is completely full. There is no empty space where the value can be inserted. Consider a scenario in which two successive deletions are made. The queue will then be given as shown in Fig. 3.10.



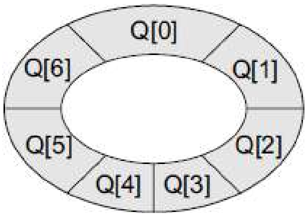
		7	18	14	36	45	21	99	72
0	1	2	3	4	5	6	7	8	9

Here, front = 2 and REAR = 9.

Suppose we want to insert a new element in the queue shown in Fig. 3.10. Even though there is space available, the overflow condition still exists because the condition rear = MAX – 1 still holds true. This is a major drawback of a linear queue.

To resolve this problem, we have two solutions. First, shift the elements to the left so that the vacant space canbe occupied and utilized efficiently. But this can be very time-consuming, especially when the queue is quite large.

The second option is to use a circular queue. In the circular queue, the first index comes right after the last index. Conceptually, you can think of a circular queue as shown in Fig. 3.11.



The circular queue will be full only when front = 0 and rear = Max – 1. A circular queue is implemented in the same manner as a linear queue is implemented. The only difference will be in the code that performs insertion and deletion operations.

For insertion, we now have to check for the following three conditions:

- If front = 0 and rear = MAX – 1, then the circular queue is full. Look at the queue given in Fig. 3.12 which illustrates this point.
- If rear != MAX – 1, then rear will be incremented and the value will be inserted as illustrated in Fig. 3.13.
- If front != 0 and rear = MAX – 1, then it means that the queue is not full. So, set rear = 0 and insert the new element there, as shown in Fig. 3.14.

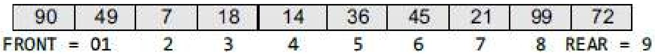
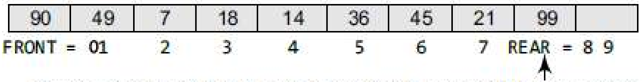
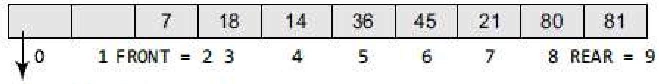


Figure 3.12 Full queue



Increment rear so that it points to location 9 and insert the value here

Figure 3.13 Queue with vacant locations



Set REAR = 0 and insert the value here

```
Step 1: IF FRONT = 0 and Rear = MAX - 1
             Write "OVERFLOW"
             Goto step 4
        [End OF IF]
Step 2: IF FRONT = -1 and REAR = -1
             SET FRONT = REAR = 0
        ELSE IF REAR = MAX - 1 and FRONT != 0
             SET REAR = 0
        ELSE
             SET REAR = REAR + 1
        [END OF IF]
Step 3: SET QUEUE[REAR] = VAL
Step 4: EXIT
```

Let us look at the algorithm to insert an element in a circular queue. In Step 1, we check for the overflow condition. In Step 2, we make two checks. First to see if the queue is empty, and second to see if the REAR end has already reached the maximum capacity while there are certain free locations before the FRONT end. In Step 3, the value is stored in the queue at the location pointed by REAR.

Let us now discuss how **deletions** are performed in this case. To delete an element, again we check for three conditions.

- Look at Fig. 3.15. If front = –1, then there are no elements in the queue. So, an underflow condition will be reported.
- If the queue is not empty and front = rear, then after deleting the element at the front the queue becomes empty and so front and rear are set to –1. This is illustrated in Fig. 3.16.
- If the queue is not empty and front = MAX–1, then after deleting the element at the front, front is set to 0. This is shown in Fig. 3.17.

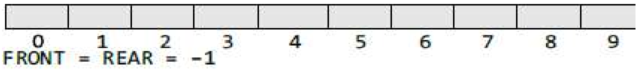
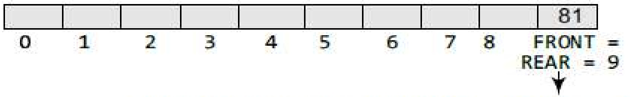
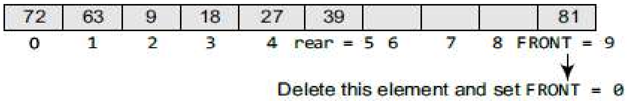


Figure 3.15 Empty queue

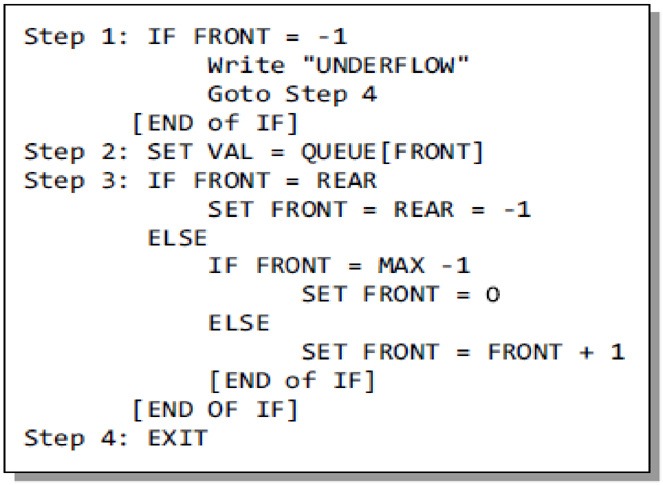


Delete this element and set REAR = FRONT = -1

Figure 3.16 Queue with a single element



Delete this element and set FRONT = 0



```
Step 1: IF FRONT = -1
             Write "UNDERFLOW"
             Goto Step 4
        [END of IF]
Step 2: SET VAL = QUEUE[FRONT]
Step 3: IF FRONT = REAR
             SET FRONT = REAR = -1
         ELSE
             IF FRONT = MAX -1
                   SET FRONT = 0
             ELSE
                   SET FRONT = FRONT + 1
             [END of IF]
        [END OF IF]
Step 4: EXIT
```

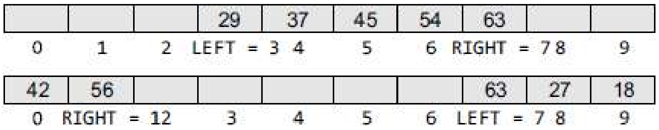
Let us look at the algorithm to delete an element from a circular queue. In Step 1, we check for the underflow condition. In Step 2, the value of the queue at the location pointed by FRONT is stored in VAL. In Step 3, we make two checks. First to see if the queue has become empty after deletion and second to see if FRONT has reached the maximum capacity of the queue. The value of FRONT is then updated based on the outcome of these checks.

3.5.2 Deques

A deque (pronounced as ‘deck’ or ‘dequeue’) is a list in which the elements can be inserted or deleted at eitherend. It is also known as a head-tail linked list because elements can be added to or removed from either the front (head) or the back (tail) end.

However, no element can be added and deleted from the middle. In the computer’s memory, a deque is implemented using either a circular array or a circular doubly linked list.

In a deque, two pointers are maintained, LEFT and RIGHT, which point to either end of the deque. The elements in a deque extend from the LEFT end to the RIGHT end and since it is circular, Dequeue[N–1] is followed by Dequeue[0]. Consider the deques shown in Fig. 3.18.


29 37 45 54 63
0 1 2 LEFT = 3 4 5 6 RIGHT = 7 8 9
42 56 63 27 18
0 RIGHT = 1 2 3 4 5 6 LEFT = 7 8 9

There are two variants of a double-ended queue. They include

- *Input restricted deque* In this dequeue, insertions can be done only at one of the ends, while deletions can be done from both ends.
- *Output restricted deque* In this dequeue, deletions can be done only at one of the ends, while insertions can be done on both ends.

3.5.3 Priority Queues

A priority queue is a data structure in which each element is assigned a priority. The priority of the element will be used to determine the order in which the elements will be processed. The general rules of processing the elements of a priority queue are

- An element with higher priority is processed before an element with a lower priority.
- Two elements with the same priority are processed on a first-come-first-served (FCFS) basis.

A priority queue can be thought of as a modified queue in which when an element has to be removed from the queue, the one with the highest-priority is retrieved first. The priority of the element can be set based on various factors. Priority queues are widely used in operating systems to execute the highest priority process first. The priority of the process may be set based on the CPU time it requires to get executed completely.

Implementation of a Priority Queue

There are two ways to implement a priority queue. We can either use a sorted list to store the elements so that when an element has to be taken out, the queue will not have to be searched for the element with the highest priority or we can use an unsorted list so that insertions are always done at the end of the list.

Every time when an element has to be removed from the list, the element with the highest priority will be searched and removed. While a sorted list takes O(n) time to insert an element in the list, it takes only O(1) time to delete an element. On the contrary, an unsorted list will take O(1) time to insert an element and O(n) time to delete an element from the list.

Practically, both these techniques are inefficient and usually a blend of these two approaches is adopted that takes roughly O(log n) time or less.

3.5.4 Multiple Queues

When we implement a queue using an array, the size of the array must be known in advance. If the queue is allocated less space, then frequent overflow conditions will be encountered.

To deal with this problem, the code will have to be modified to reallocate more space for the array. In case we allocate a large amount of space for the queue, it will result in sheer wastage of the memory.

Thus, there lies a tradeoff between the frequency of overflows and the space allocated. So a better solution to deal with this problem is to have multiple queues or to have more than one queue in the same array of sufficient size. Figure 3.19 illustrates this concept.

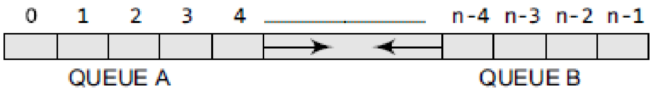
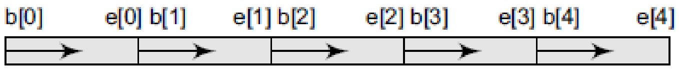


Figure 3.19 Multiple queues



In the figure, an array Queue[n] is used to represent two queues, Queue A and Queue B. The value of n is suchthat the combined size of both the queues will never exceed n. While operating on these queues, it is importantto note one thing—queue

A will grow from left to right, whereas queue B will grow from right to left at the same time. Extending the concept to multiple queues, a queue can also be used to represent n number of queues in the same array. That is, if we have a QUEUE[n], then each queue I will be allocated an equal amount of space bounded by indices b[i] and e[i]. This is shown in Fig. 3.20.

Applications of Queue

Queues are an important data structure that follows the First-In-First-Out (FIFO) principle. They are used in various applications, including:

Operating Systems: Queues are used in operating systems to manage the scheduling of processes and threads. Each process or thread is added to a queue, and the operating system determines which process or thread should be executed next based on scheduling algorithms.

Networking: Queues are used in networking to manage the flow of data packets between different devices on a network. For example, routers use queues to store packets that are waiting to be transmitted to their destination.

Print Spooling: Print jobs that are sent to a printer are stored in a queue until the printer is ready to print them. This ensures that the printer can handle multiple print jobs without getting overwhelmed.

Traffic Management: Queues are used in traffic management systems to manage the flow of vehicles at intersections, toll booths, and other traffic control points. Vehicles are queued up in lanes and allowed to proceed through the control point in a first-come, first-served order.

Call Center Management: Queues are used in call centers to manage the flow of incoming calls. Calls are placed in a queue and assigned to available agents based on various routing algorithms.

Event-driven Programming: Queues are used in event-driven programming to manage the flow of events. Events are added to a queue, and the program processes them in a first-come, first-served order.

Job Scheduling: Queues are used in job scheduling to manage the order in which tasks are executed. Jobs are added to a queue and processed in the order in which they were added.

Breadth-First Search: Queues are used in graph traversal algorithms, such as Breadth-First Search, to keep track of nodes that need to be explored. Nodes are added to a queue and explored in the order in which they were added.

Overall, queues are a fundamental data structure used in many applications that require the efficient management of a collection of data elements.

Multiple-choice Questions

1. A line in a grocery store represents a

(a) Stack (b) Queue

(c) Linked List (d) Array

2. In a queue, insertion is done at

(a) Rear (b) Front

(c) Back (d) Top

3. The function that deletes values from a queue iscalled

(a) enqueue (b) dequeue

(c) pop (d) peek

4. Typical time requirement for operations onqueues is

(a) O(1) (b) O(n)

(c) O(log n) (d) O(n2)

5. The circular queue will be full only when

(a) FRONT = MAX –1 and REAR = Max –1

(b) FRONT = 0 and REAR = Max –1

(c) FRONT = MAX –1 and REAR = 0

(d) FRONT = 0 and REAR = 0

Fill in the Blanks

1. New nodes are added at__________of the queue.

2. _______allows insertion of elements at either ends but not in the middle.

3. The typical time requirement for operations in a linked queue is____________.

4. In_______, insertions can be done only at one end, while deletions can be done from both the ends.

5. Dequeue is implemented using__________.

6. _______ are appropriate data structures to process batch computer programs submitted to the computercentre.

7. _______ are appropriate data structures to process a list of employees having a contract for a senioritysystem for hiring and firing.

M.Sc. (Computer Science)

SEMESTER-2

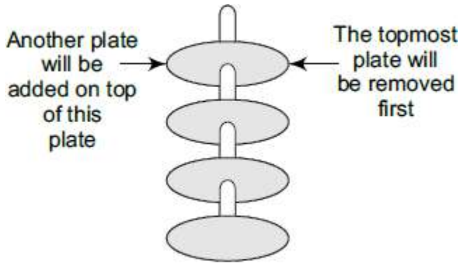
COURSE: DATA STRUCTURE AND ALGORITHMS

UNIT 4: STACKS

4.1 INTRODUCTION

Stack is an important data structure which stores its elements in an ordered manner. We will explain the concept of stacks using an analogy. You must have seen a pile of plates where one plate is placed on top of another as shown in Fig. 3.21.

Now, when you want to remove a plate, you remove the topmost plate first. Hence, you can add and removean element (i.e., a plate) only at/from one position which is the topmost position.


Another plate will be added on top of this plate
The topmost plate will be removed first

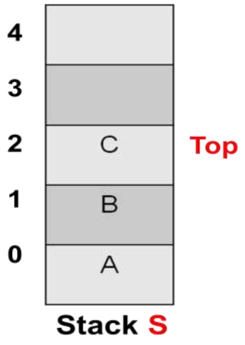
A stack is a linear data structure which uses the same principle, i.e., the elements in a stack are added and removed only from one end, which is called the TOP.

Hence, a stack is called a LIFO (Last-In-First-Out) data structure, as the element that was inserted last is the first one to be taken out.

4.2 ARRAY REPRESENTATION OF STACKS

In the computer’s memory, stacks can be represented as a linear array. Every stack has a variable called TOP associated with it, which is used to store the address of the topmost element of the stack. It is this position where the element will be added to or deleted from.

There is another variable called MAX, which is used to store the maximum number of elements that the stack can hold. If TOP = NULL, then it indicates that the stack is empty and if TOP = MAX–1, then the stack is full. (You must be wondering why we have written MAX–1. It is because array indices start from 0.) Look at Fig. 3.22.



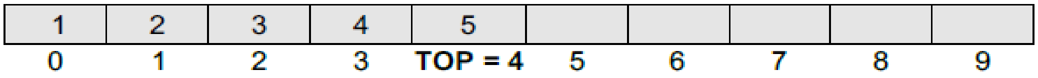
4.3 OPERATIONS ON A STACK

A stack supports three basic operations: push, pop, and peek.

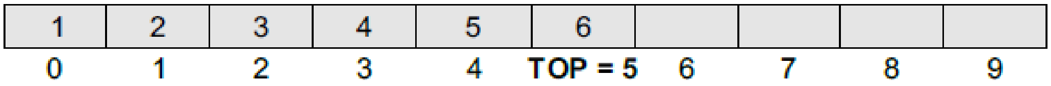
The **push** operation adds an element to the top of the stack and the **pop** operation removes the element fromthe top of the stack. The **peek** operation returns the value of the topmost element of the stack.

Push Operation

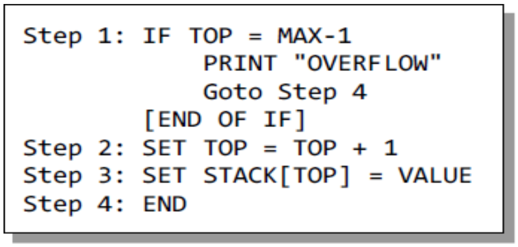
- The push operation is used to insert an element into the stack.
- The new element is added at the topmost position of the stack.
- To insert an element with value 6, we first check if TOP=MAX–1.
- If the condition is false, then we increment the value of TOP and store the new element at theposition given by stack[TOP].



1	2	3	4	5					
0	1	2	3	TOP = 4	5	6	7	8	9



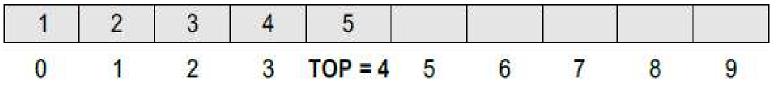
1	2	3	4	5	6				
0	1	2	3	4	TOP = 5	6	7	8	9



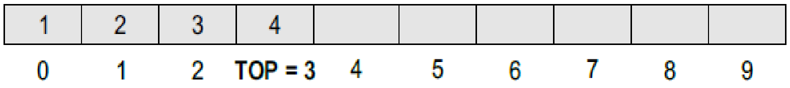
```
Step 1: IF TOP = MAX-1
              PRINT "OVERFLOW"
              Goto Step 4
        [END OF IF]
Step 2: SET TOP = TOP + 1
Step 3: SET STACK[TOP] = VALUE
Step 4: END
```

Pop Operation

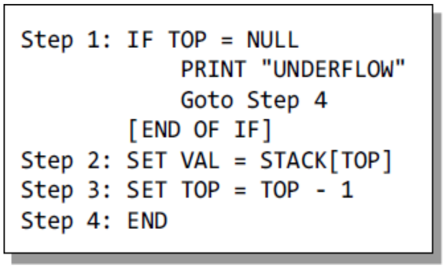
- The pop operation is used to delete the topmost element from the stack.
- However, before deleting the value, we must first check if TOP=NULL because if that is the case, thenit means the stack is empty and no more deletions can be done.
- To delete the topmost element, we first check if TOP=NULL. If the condition is false, then wedecrement the value pointed by TOP.



1	2	3	4	5					
0	1	2	3	TOP = 4	5	6	7	8	9



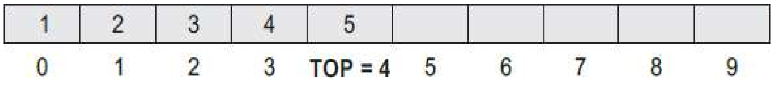
1	2	3	4						
0	1	2	TOP = 3	4	5	6	7	8	9

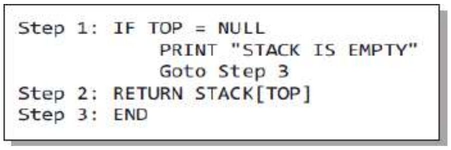


```
Step 1: IF TOP = NULL
              PRINT "UNDERFLOW"
              Goto Step 4
        [END OF IF]
Step 2: SET VAL = STACK[TOP]
Step 3: SET TOP = TOP - 1
Step 4: END
```

Peek Operation

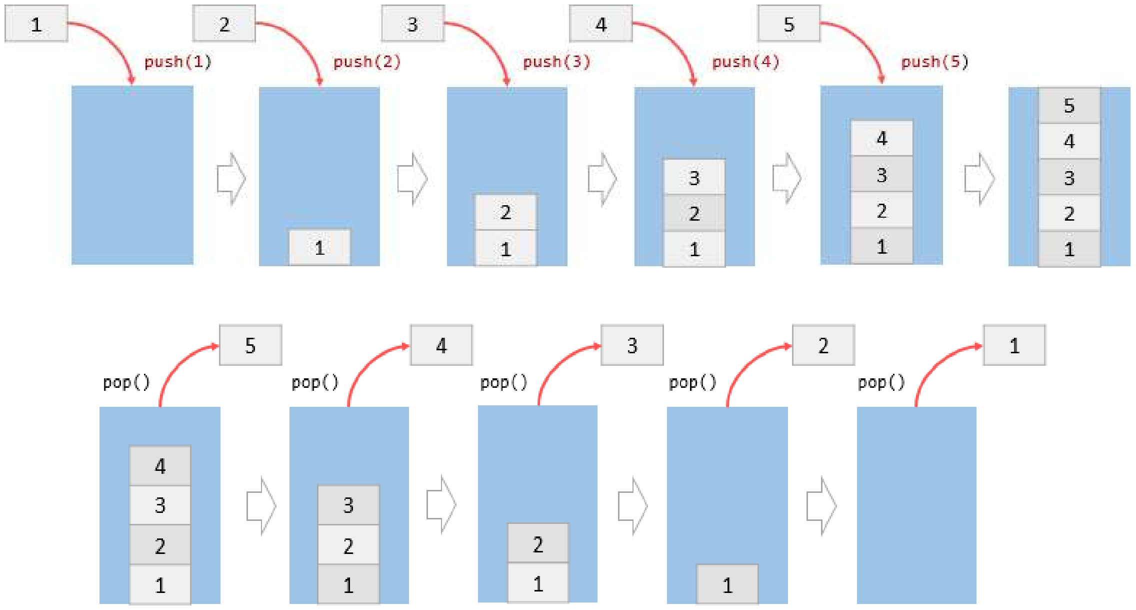
- Peek is an operation that returns the value of the topmost element of the stack without deleting it fromthe stack.
- However, the Peek operation first checks if the stack is empty, i.e., if TOP = NULL, then anappropriate message is printed, else the value is returned.
- Here, the Peek operation will return 5,as it is the value of the topmost element of the stack.





```
Step 1: IF TOP = NULL
            PRINT "STACK IS EMPTY"
            Goto Step 3
Step 2: RETURN STACK[TOP]
Step 3: END
```

Example:

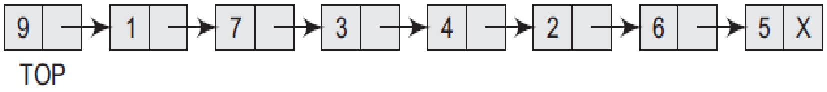


4.4 LINKED REPRESENTATION OF STACKS

We have seen how a stack is created using an array. This technique of creating a stack is easy, but the drawback is that the array must be declared to have some fixed size. In case the stack is a very small one or its maximum size is known in advance, then the array implementation of the stack gives an efficient implementation. But if the array size cannot be determined in advance, then the other alternative, i.e., linked representation, is used.

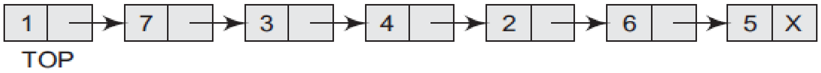
The storage requirement of linked representation of the stack with n elements is O(n), and the typical time requirement for the operations is O(1).

In a linked stack, every node has two parts—one that stores data and another that stores the address of the nextnode. The START pointer of the linked list is used as TOP. All insertions and deletions are done at the node pointed by TOP. If TOP = NULL, then it indicates that the stack is empty. The linked representation of a stackis shown in below figure.

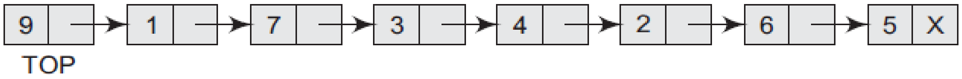


Push Operation

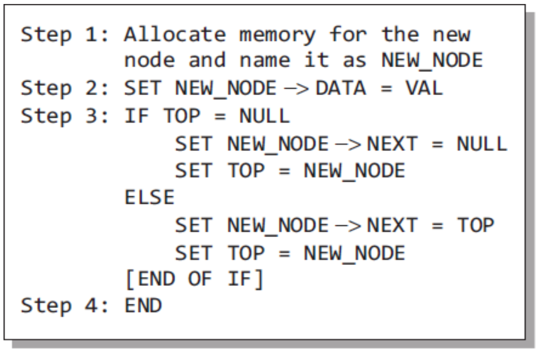
The push operation is used to insert an element into the stack. The new element is added at the topmostposition of the stack. Consider the linked stack shown in below figure.



To insert an element with value 9, we first check if TOP=NULL. If this is the case, then we allocate memory for a new node, store the value in its DATA part and NULL in its NEXT part. The new node will then be called TOP. However, if TOP!=NULL, then we insert the new node at the beginning of the linked stack and name this new node as TOP.



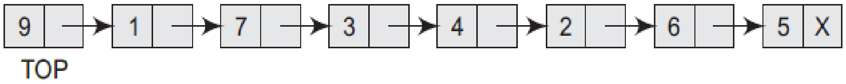
the algorithm to push an element into a linked stack. In Step 1, memory is allocated for the new node. In Step2, the DATA part of the new node is initialized with the value to be stored in the node. In Step 3, we check ifthe new node is the first node of the linked list. is done by checking if TOP = NULL. In case the IF statementevaluates to true, then NULL is stored in the NEXT part of the node and the new node is called TOP. However, if the new node is not the first node in the list, then it is added before the first node of the list (that is, the TOP node) and termed as TOP.



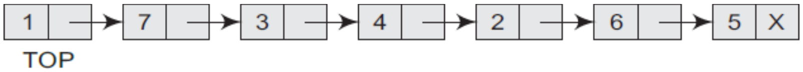
```
Step 1: Allocate memory for the new
        node and name it as NEW_NODE
Step 2: SET NEW_NODE -> DATA = VAL
Step 3: IF TOP = NULL
            SET NEW_NODE -> NEXT = NULL
            SET TOP = NEW_NODE
        ELSE
            SET NEW_NODE -> NEXT = TOP
            SET TOP = NEW_NODE
        [END OF IF]
Step 4: END
```

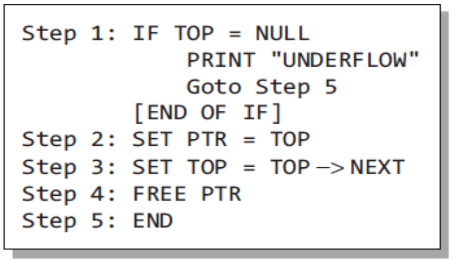
Pop Operation

The pop operation is used to delete the topmost element from a stack. However, before deleting the value, we must first check if TOP=NULL, because if this is the case, then it means that the stack is empty and no more deletions can be done. If an attempt is made to delete a value from a stack that is already empty, an UNDERFLOW message is printed. Consider the stack shown in below figure.



In case TOP!=NULL, then we will delete the node pointed by TOP, and make TOP point to the second element of the linked stack. Thus, the updated stack becomes as shown in below figure.





```
Step 1: IF TOP = NULL
            PRINT "UNDERFLOW"
            Goto Step 5
        [END OF IF]
Step 2: SET PTR = TOP
Step 3: SET TOP = TOP -> NEXT
Step 4: FREE PTR
Step 5: END
```

The algorithm to delete an element from a stack.

In Step 1, we first check for the UNDERFLOW condition.
In Step 2, we use a pointer PTR that points to TOP.
In Step 3, TOP is made to point to the next node in sequence.
In Step 4, the memory occupied by PTR is given back to the free pool.

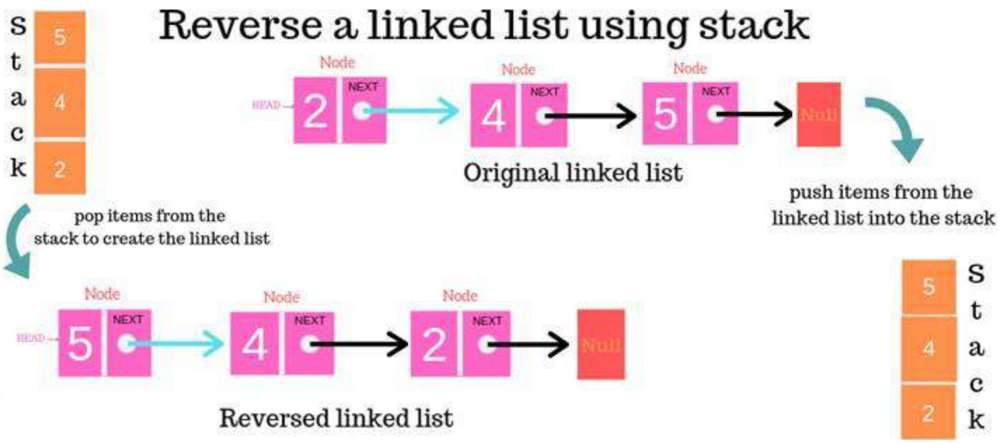
4.5 APPLICATIONS OF STACKS

In this section we will discuss typical problems where stacks can be easily applied for a simple and efficient solution. The topics that will be discussed in this section include the following:

Reversing a list
Parentheses checker
Conversion of an infix expression into a postfix expression
Evaluation of a postfix expression
Conversion of an infix expression into a prefix expression
Evaluation of a prefix expression
Recursion
Tower of Hanoi

4.6 REVERSING LIST

A list of numbers can be reversed by reading each number from an array starting from the first index and pushing it on a stack. Once all the numbers have been read, the numbers can be popped one at a time and then stored in the array starting from the first index.


Reverse a linked list using stack
s t a c k
5
4
2
Node
NEXT
HEAD
2
4
5
Null
Original linked list
push items from the linked list into the stack
pop items from the stack to create the linked list
5
4
2
Null
Reversed linked list
s t a c k

4.7 EVALUATION OF ARITHMETIC EXPRESSIONS

Polish Notations:

Infix, postfix, and prefix notations are three different but equivalent notations of writing algebraic expressions.But before learning about prefix and postfix notations, let us first see what an infix notation is. We all are familiar with the infix notation of writing algebraic expressions.

While writing an arithmetic expression using infix notation, the operator is placed in between the operands. For example, A+B; here, plus operator is placed between the two operands A and B. Although it is easy for us to write expressions using infix notation, computers find it difficult to parse as the computer needs a lot of information to evaluate the expression. Information is needed about operator precedence and associativity rules, and brackets which override these rules.

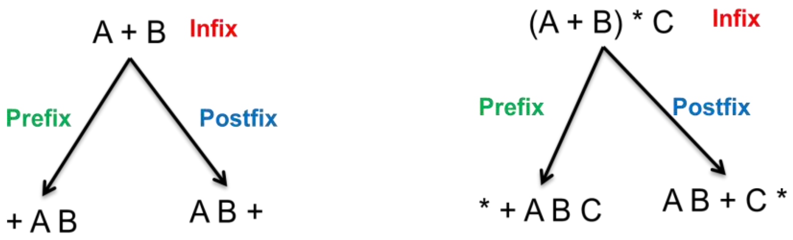
So, computers work more efficiently with expressions written using prefix and postfix notations. Postfix notation was developed by Jan Łukasiewicz who was a Polish logician, mathematician, and philosopher. His aim was to develop a parenthesis-free prefix notation (also known as Polish notation) and a postfix notation, which is better known as Reverse Polish Notation or RPN.

In postfix notation, as the name suggests, the operator is placed after the operands. For example, if an expression is written as A+B in infix notation, the same expression can be written as AB+ in postfix notation. The order of evaluation of a postfix expression is always from left to right. Even brackets cannot alter the order of evaluation.

The expression (A + B) * C can be written as:
[AB+]*C
AB+C* in the postfix notation

A postfix operation does not even follow the rules of operator precedence. The operator which occurs first inthe expression is operated first on the operands.

For example, given a postfix notation AB+C*. While evaluation, addition will be performed prior to multiplication. Thus we see that in a postfix notation, operators are applied to the operands that are immediately left to them. In the example, AB+C*, + is applied on A and B, then * is applied on the result of addition and C.



Conversion of an Infix Expression into a Postfix Expression:

Let I be an algebraic expression written in infix notation. I may contain parentheses, operands, and operators. For simplicity of the algorithm we will use only +, –, *, /, % operators.

The precedence of these operators can be given as follows:

- Higher priority *, /, %
- Lower priority +, –

No doubt, the order of evaluation of these operators can be changed by making use of parentheses. For example, if we have an expression A + B * C, then first B * C will be done and the result will be added to A. But the same expression if written as, (A + B) * C, will evaluate A + B first and then the result will be multiplied with C.

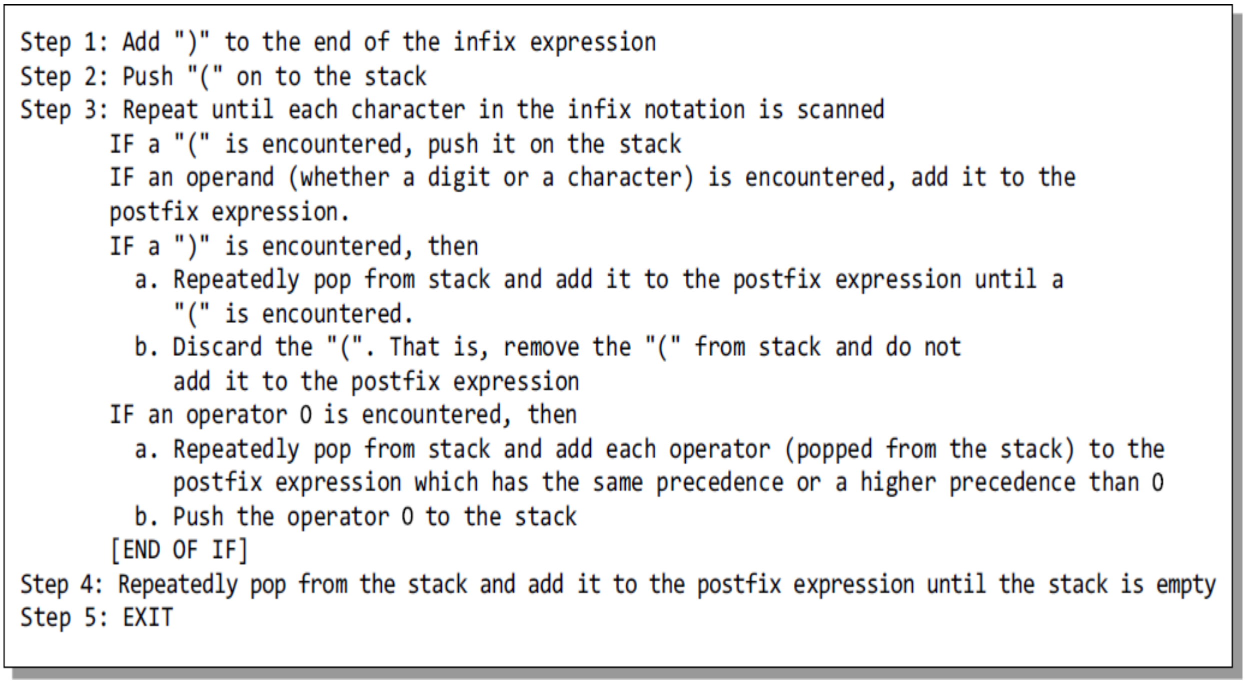
Example: Convert the following infix expressions into postfix expressions.
Solution:
(a) (A–B) * (C+D)
[AB–] * [CD+]
AB–CD+*

(b) (A + B) / (C + D) – (D * E)
[AB+] / [CD+] – [DE*]
[AB+CD+/] – [DE*]
AB+CD+/DE*–

The algorithm given below transforms an infix expression into postfix expression. The algorithm accepts an infix expression that may contain operators, operands, and parentheses.

For simplicity, we assume that the infix operation contains only modulus (%), multiplication (*), division (/), addition (+), and subtraction (—) operators and that operators with same precedence are performed from left- to-right.

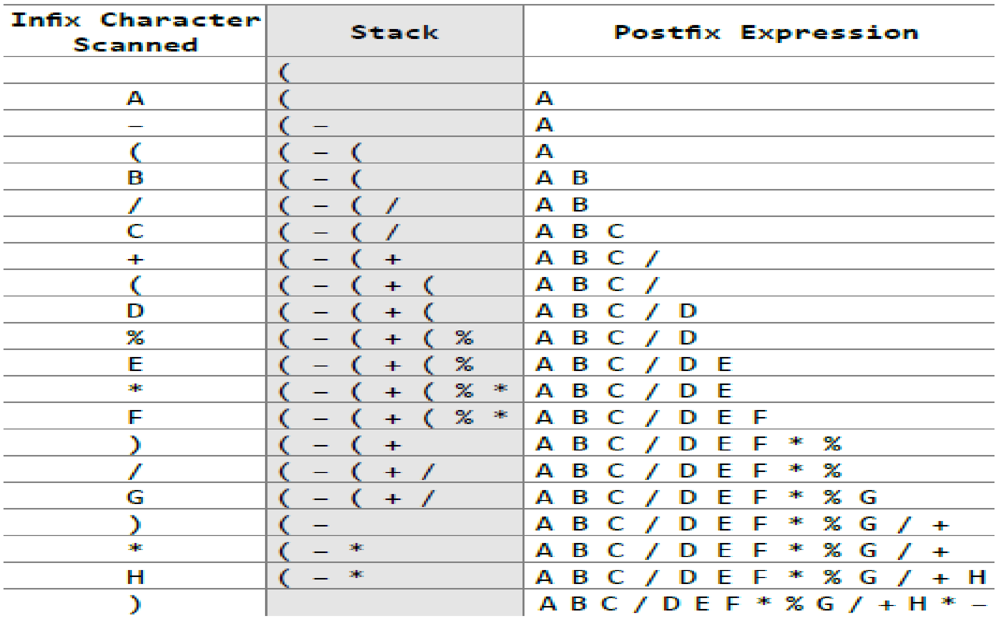
The algorithm uses a stack to temporarily hold operators. The postfix expression is obtained from left-to-right using the operands from the infix expression and the operators which are removed from the stack. The first step in this algorithm is to push a left parenthesis on the stack and to add a corresponding right parenthesis at the end of the infix expression. The algorithm is repeated until the stack is empty.



```
Step 1: Add ")" to the end of the infix expression
Step 2: Push "(" on to the stack
Step 3: Repeat until each character in the infix notation is scanned
        IF a "(" is encountered, push it on the stack
        IF an operand (whether a digit or a character) is encountered, add it to the
        postfix expression.
        IF a ")" is encountered, then
          a. Repeatedly pop from stack and add it to the postfix expression until a
             "(" is encountered.
          b. Discard the "(". That is, remove the "(" from stack and do not
             add it to the postfix expression
        IF an operator O is encountered, then
          a. Repeatedly pop from stack and add each operator (popped from the stack) to the
             postfix expression which has the same precedence or a higher precedence than O
          b. Push the operator O to the stack
        [END OF IF]
Step 4: Repeatedly pop from the stack and add it to the postfix expression until the stack is empty
Step 5: EXIT
```

Example: Convert the following infix expression into postfix expression using the algorithmA – (B / C + (D % E * F) / G)* H

A – (B / C + (D % E * F) / G)* H)



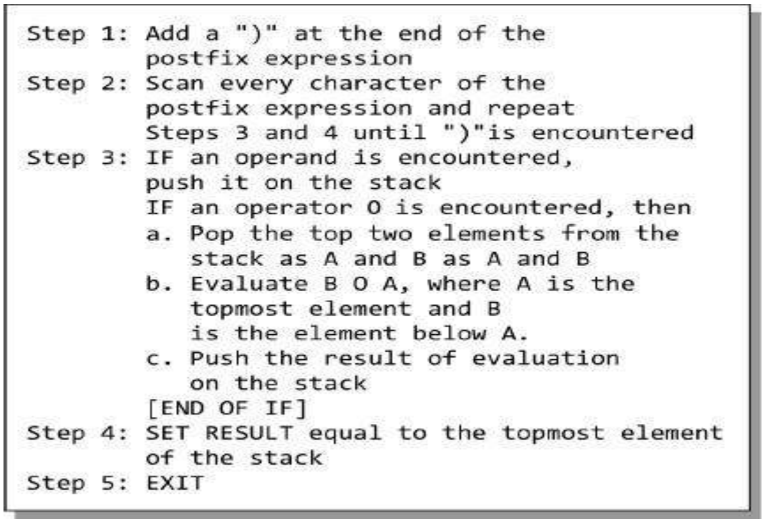
Infix Character Scanned	Stack	Postfix Expression
	(	
A	(	A
–	(–	A
(	(– (	A
B	(– (	A B
/	(– (/	A B
C	(– (/	A B C
+	(– (+	A B C /
(	(– (+ (	A B C /
D	(– (+ (	A B C / D
%	(– (+ (%	A B C / D
E	(– (+ (%	A B C / D E
*	(– (+ (% *	A B C / D E
F	(– (+ (% *	A B C / D E F
)	(– (+	A B C / D E F * %
/	(– (+ /	A B C / D E F * %
G	(– (+ /	A B C / D E F * % G
)	(–	A B C / D E F * % G / +
*	(– *	A B C / D E F * % G / +
H	(– *	A B C / D E F * % G / + H
)		A B C / D E F * % G / + H * –

Evaluation of a Postfix Expression:

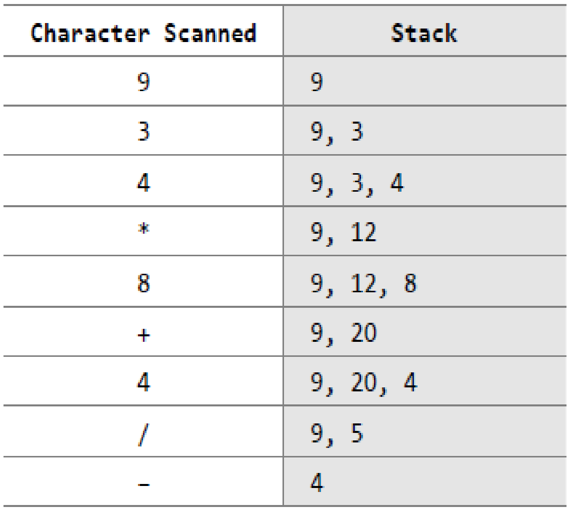
The ease of evaluation acts as the driving force for computers to translate an infix notation into a postfix notation. That is, given an algebraic expression written in infix notation, the computer first converts the expression into the equivalent postfix notation and then evaluates the postfix expression.

Both these tasks—converting the infix notation into postfix notation and evaluating the postfix expression— make extensive use of stacks as the primary tool.

Using stacks, any postfix expression can be evaluated very easily. Every character of the postfix expression is scanned from left to right. If the character encountered is an operand, it is pushed on to the stack. However, if an operator is encountered, then the top two values are popped from the stack and the operator is applied on these values. The result is then pushed on to the stack.



```
Step 1: Add a ")" at the end of the
        postfix expression
Step 2: Scan every character of the
        postfix expression and repeat
        Steps 3 and 4 until ")"is encountered
Step 3: IF an operand is encountered,
        push it on the stack
        IF an operator O is encountered, then
        a. Pop the top two elements from the
           stack as A and B as A and B
        b. Evaluate B O A, where A is the
           topmost element and B
           is the element below A.
        c. Push the result of evaluation
           on the stack
        [END OF IF]
Step 4: SET RESULT equal to the topmost element
        of the stack
Step 5: EXIT
```

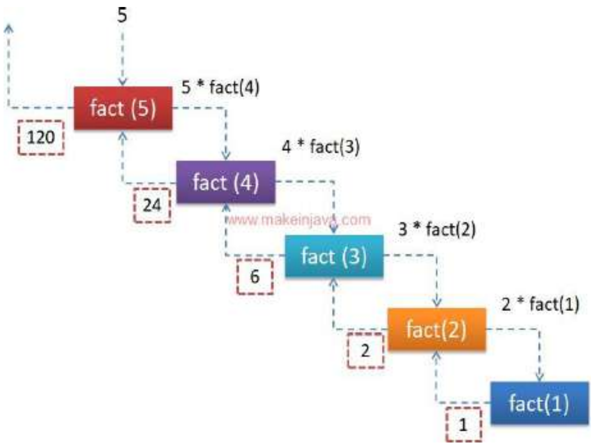


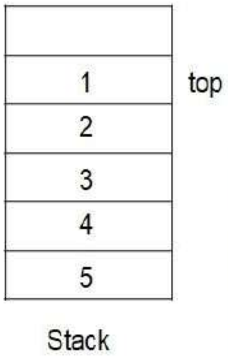
Character Scanned	Stack
9	9
3	9, 3
4	9, 3, 4
*	9, 12
8	9, 12, 8
+	9, 20
4	9, 20, 4
/	9, 5
–	4

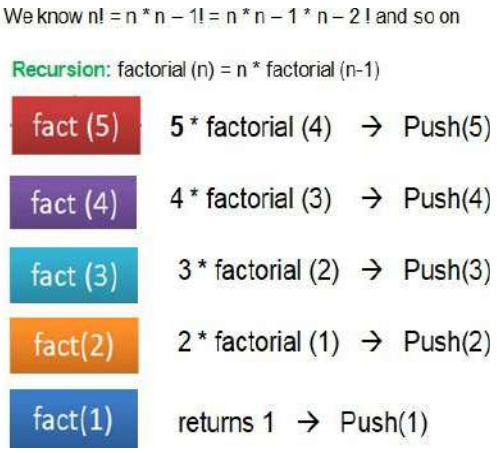
Algorithm to evaluate a postfix expression   Evaluation of a

postfix expressionLet us now take an example that makes use of this algorithm.
Consider the infix expression given as 9 – ((3 * 4) + 8) / 4. Evaluate the expression.
The infix expression 9 – ((3 * 4) + 8) / 4 can be written as 9 3 4 * 8 + 4 / – using postfix notation.

Factorial Calculation:






We know n! = n * n – 1! = n * n – 1 * n – 2 ! and so on
Recursion: factorial (n) = n * factorial (n-1)
fact (5)
5 * factorial (4) → Push(5)
fact (4)
4 * factorial (3) → Push(4)
fact (3)
3 * factorial (2) → Push(3)
fact(2)
2 * factorial (1) → Push(2)
fact(1)
returns 1 → Push(1)

Stack

Pop(1)	Factorial value = 1
Pop(2)	Factorial value = 1 * 2 = 2
Pop(3)	Factorial value = 2 * 3 = 6
Pop(4)	Factorial value = 6 * 4 = 24
Pop(5)	Factorial value = 24 * 5 = 120

Factorial value = 120

M.Sc. (Computer Science)

SEMESTER-2

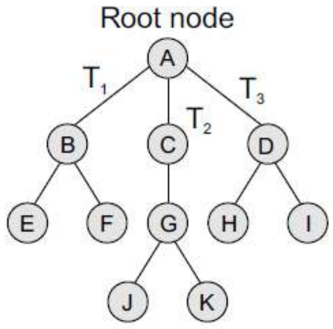
COURSE: DATA STRUCTURE AND ALGORITHMS

UNIT 5: TREE

5.1 BASIC TERMINOLOGY IN TREES

A tree is recursively defined as a set of one or more nodes where one node is designated as the root of the tree and all the remaining nodes can be partitioned into non-empty sets each of which is a sub-tree of the root.

Figure 4.1 shows a tree where node A is the root node; nodes B, C, and D are children of the root node and form sub-trees of the tree rooted at node A.



Root node The root node R is the topmost node in the tree. If R = NULL, then it means the tree is

empty.**Sub-trees** If the root node R is not NULL, then the trees T1, T2, and T3 are called the sub-

trees of R. **Leaf node** A node that has no children is called the leaf node or the terminal node.

Path A sequence of consecutive edges is called a path. For example, in Fig. 9.1, the path from the root node Ato node I is given as: A, D, and I.

Ancestor node An ancestor of a node is any predecessor node on the path from root to that node. The rootnode does not have any ancestors. In the tree given in Fig. 9.1, nodes A, C, and G are the ancestors of node K.

Descendant node A descendant node is any successor node on any path from the node to a leaf node. Leaf nodes do not have any descendants. In the tree given in Fig. 9.1, nodes C, G, J, and K are the descendants of node A.

Level number Every node in the tree is assigned a level number in such a way that the root node is at level 0, children of the root node are at level number 1. Thus, every node is at one level higher than its parent. So, all child nodes have a level number given by parent's level number + 1.

Degree Degree of a node is equal to the number of children that a node has. The degree of a leaf node is zero.

✓ **In-degree** In-degree of a node is the number of edges arriving at that node.

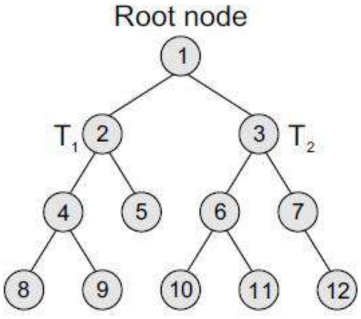
✓ **Out-degree** Out-degree of a node is the number of edges leaving that node.

5.2 BINARY TREES

A binary tree is a data structure that is defined as a collection of elements called nodes. In a binary tree, the topmost element is called the root node, and each node has 0, 1, or at the most 2 children.

A node that has zero children is called a leaf node or a terminal node. Every node contains a data element, a left pointer which points to the left child, and a right pointer which points to the right child. The root elementis pointed by a 'root' pointer. If root = NULL, then it means the tree is empty.

Figure 4.2 shows a binary tree. In the figure, R is the root node and the two trees T1 and T2 are called the left and right sub-trees of R. T1 is said to be the left successor of R. Likewise, T2 is called the right successor ofR.



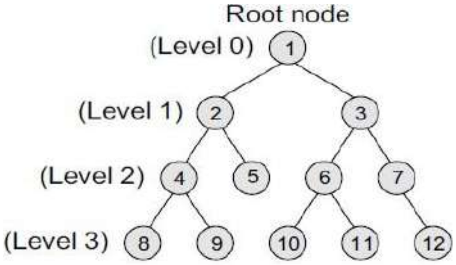
Note that the left sub-tree of the root node consists of the nodes: 2, 4, 5, 8, and 9. Similarly, the right sub-tree of the root node consists of nodes: 3, 6, 7, 10, 11, and 12.

In the tree, root node 1 has two successors: 2 and 3. Node 2 has two successor nodes: 4 and 5. Node 4 has two successors: 8 and 9. Node 5 has no successor. Node 3 has two successor nodes: 6 and 7. Node 6 has two
successors: 10 and 11. Finally, node 7 has only one successor: 12.

A binary tree is recursive by definition as every node in the tree contains a left sub-tree and a right sub-tree. Even the terminal nodes contain an empty left sub-tree and an empty right sub-tree.
Look at Fig. 4.2, nodes 5, 8, 9, 10, 11, and 12 have no successors and thus said to have empty sub-trees.

Terminology:

Parent If N is any node in T that has left successor S1 and right successor S2, then N is called the parent of S1 and S2. Correspondingly, S1 and S2 are called the left child and the right child of N. Every node other than the root node has a parent.



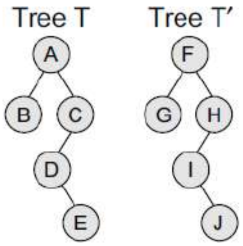
Level number Every node in the binary tree is assigned a level number (refer Fig. 4.3). The root node is defined to be at level 0. The left and the right child of the root node have a level number 1. Similarly, every node is at one level higher than its parents. So all child nodes are defined to have level number as parent's level number + 1.

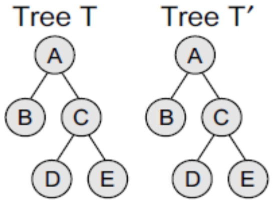
Degree of a node It is equal to the number of children that a node has. The degree of a leaf node is zero. For example, in the tree, degree of node 4 is 2, degree of node 5 is zero and degree of node 7 is 1.

Sibling All nodes that are at the same level and share the same parent are called siblings (brothers). For example, nodes 2 and 3; nodes 4 and 5; nodes 6 and 7; nodes 8 and 9; and nodes 10 and 11 are siblings.

Leaf node A node that has no children is called a leaf node or a terminal node. The leaf nodes in the tree are: 8, 9, 5, 10, 11, and 12.

Similar binary trees Two binary trees T and T’ are said to be similar if both these trees have the same structure. Figure 4.4 shows two similar binary trees.





Copies Two binary trees T and T’ are said to be copies if they have similar structure and if they have same content at the corresponding nodes. Figure 4.5 shows that T’ is a copy of T.

Edge It is the line connecting a node N to any of its successors. A binary tree of n nodes has exactly n – 1 edges because every node except the root node is connected to its parent via an edge.

Path A sequence of consecutive edges. For example, in Fig. 4.3, the path from the root node to the node 8 is given as: 1, 2, 4, and 8.

Depth The depth of a node N is given as the length of the path from the root R to the node N. The depth of theroot node is zero.

Height of a tree It is the total number of nodes on the path from the root node to the deepest node in the tree. A tree with only a root node has a height of 1.

A binary tree of height h has at least h nodes and at most 2^h – 1 nodes. This is because every level will have at least one node and can have at most 2 nodes. So, if every level has two nodes then a tree with height h will have at the most 2^h – 1 nodes as at level 0, there is only one element called the root. The height of a binary tree with n nodes is at least $\log_2(n+1)$ and at most n.

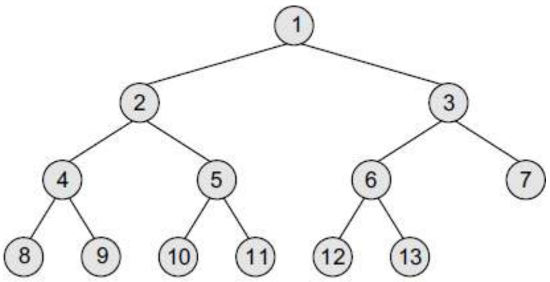
In-degree/out-degree of a node It is the number of edges arriving at a node. The root node is the only nodethat has an in-degree equal to zero. Similarly, out-degree of a node is the number of edges leaving that node.

Binary trees are commonly used to implement binary search trees, expression trees, tournament trees, andbinary heaps.

Complete Binary Trees

A complete binary tree is a binary tree that satisfies two properties. First, in a complete binary tree, everylevel, except possibly the last, is completely filled. Second, all nodes appear as far left as possible.

In a complete binary tree T_n, there are exactly n nodes and level r of T can have at most 2^r nodes. Figure 9.7shows a complete binary tree.



Note that in Fig. 4.6, level 0 has 2^0 = 1 node, level 1 has 2^1 = 2 nodes, level 2 has 2^2 = 4 nodes, level 3 has 6 nodes which is less than the maximum of 2^3 = 8 nodes.

In Fig. 4.6, tree T_{13} has exactly 13 nodes. They have been purposely labelled from 1 to 13, so that it is easy for the reader to find the parent node, the right child node, and the left child node of the given node.

The formula can be given as—if K is a parent node, then its left child can be calculated as 2 × K and its rightchild can be calculated as 2 × K + 1.

For example, the children of the node 4 are 8 (2 × 4) and 9 (2 × 4 + 1).Similarly, the parent of the node K can be calculated as | K/2 |.

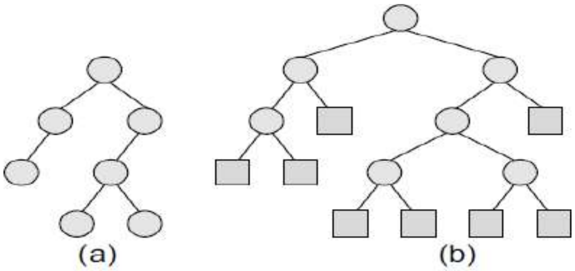
Given the node 4, its parent can be calculated as | 4/2 | = 2. The height of a tree T_n having exactly n nodes isgiven as: H_n = | $\log_2$ (n + 1) |

NOTE: This means, if a tree T has 10,00,000 nodes, then its height is 21

Extended Binary Trees

A binary tree T is said to be an extended binary tree (or a 2-tree) if each node in the tree has either no child orexactly two children. Figure 4.7 shows how an ordinary binary tree is converted into an extended binary tree. In an extended binary tree, nodes having two children are called internal nodes and nodes having no childrenare called external nodes. In Fig. 4.7, the internal nodes are represented using circles and the external nodesare represented using squares.

To convert a binary tree into an extended tree, every empty sub-tree is replaced by a new node. The originalnodes in the tree are the internal nodes, and the new nodes added are called the external nodes.



5.3 REPRESENTATION OF BINARY TREES IN THE MEMORY

In the computer's memory, a binary tree can be maintained either by using a linked representation or by usinga sequential representation.

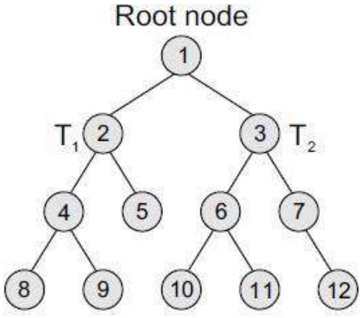
Linked representation of binary trees In the linked representation of a binary tree, every node will havethree parts: the data element, a pointer to the left node, and a pointer to the right node.

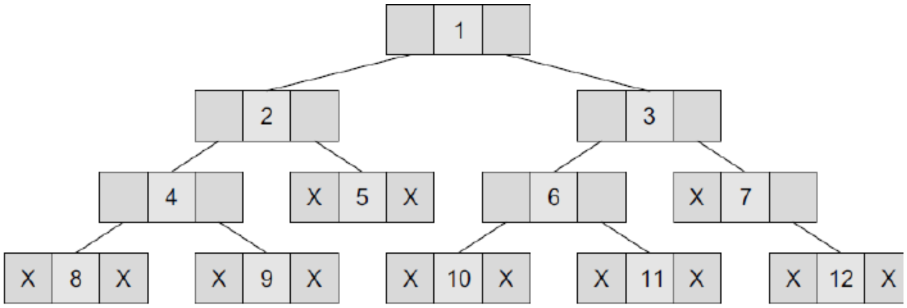
So in C, the binary tree is built with a node type given below.struct node

```
{
struct node *left;int data;
struct node *right;
};
```

Every binary tree has a pointer ROOT, which points to the root element (topmost element) of the tree. If ROOT = NULL, then the tree is empty. Consider the binary tree given in Fig. 4.2. The schematic diagram of the linked representation of the binary tree is shown in Fig. 4.8.

In Fig. 4.8, the left position is used to point to the left child of the node or to store the address of the left child of the node. The middle position is used to store the data. Finally, the right position is used to point to the rightchild of the node or to store the address of the right child of the node. Empty sub-trees are represented using X(meaning NULL).





Sequential representation of binary trees Sequential representation of trees is done using single or one-dimensional arrays. Though it is the simplest technique for memory representation, it is inefficient as it requires a lot of memory space.

A sequential binary tree follows the following rules:

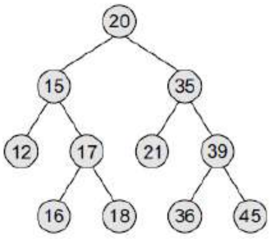
✓ A one-dimensional array, called TREE, is used to store the elements of tree.

✓ The root of the tree will be stored in the first location. That is, TREE[1] will store the data of the rootelement.

✓ The children of a node stored in location K will be stored in locations (2 × K) and (2 × K+1).

✓ The maximum size of the array TREE is given as (2h–1), where h is the height of the tree.

✓ An empty tree or sub-tree is specified using NULL. If TREE[1] = NULL, then the tree is empty. Figure 4.9 shows a binary tree and its corresponding sequential representation. The tree has 11 nodes and itsheight is 4.



1	20
2	15
3	35
4	12
5	17
6	21
7	39
8	
9	
10	16
11	18
12	
13	
14	36
15	45

5.4 TRAVERSING A BINARY TREE

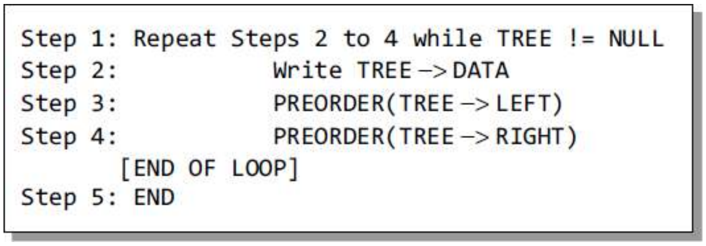
Traversing a binary tree is the process of visiting each node in the tree exactly once in a systematic way. Unlike linear data structures in which the elements are traversed sequentially, tree is a nonlinear data structure in which the elements can be traversed in many different ways. There are different algorithms for tree traversals. These algorithms differ in the order in which the nodes are visited. In this section, we will discuss these algorithms.

Pre-order Traversal

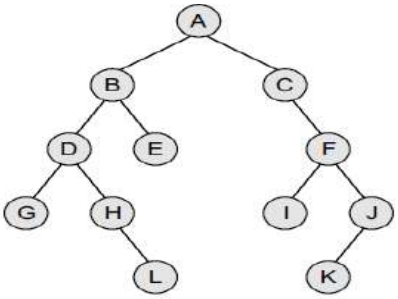
To traverse a non-empty binary tree in pre-order, the following operations are performed recursively at each node. The algorithm works by:

1. Visiting the root node,
2. Traversing the left sub-tree, and finally
3. Traversing the right sub-tree.

Pre-order traversal is also called as depth-first traversal. In this algorithm, the left sub-tree is always traversedbefore the right sub-tree. The word ‘pre’ in the pre-order specifies that the root node is accessed prior to anyother nodes in the left and right sub-trees. Pre-order algorithm is also known as the NLR traversal algorithm (Node-Left-Right).



```
Step 1: Repeat Steps 2 to 4 while TREE != NULL
Step 2:             Write TREE -> DATA
Step 3:             PREORDER(TREE -> LEFT)
Step 4:             PREORDER(TREE -> RIGHT)
        [END OF LOOP]
Step 5: END
```

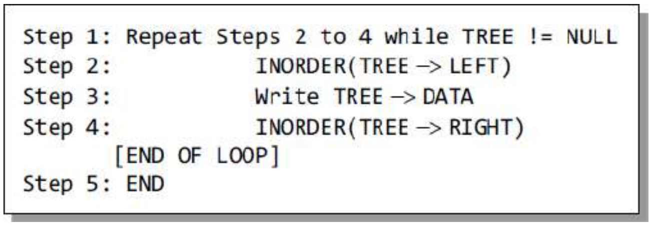


TRAVERSAL ORDER:
A, B, D, G, H, L, E, C, F, I, J, and K

In-order Traversal

To traverse a non-empty binary tree in in-order, the following operations are performed recursively at eachnode. The algorithm works by:

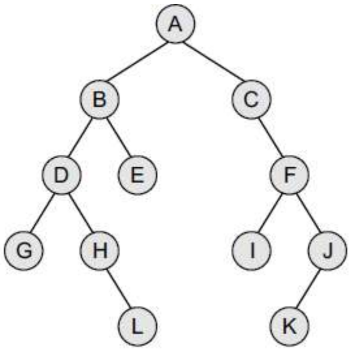
1. Traversing the left sub-tree,
2. Visiting the root node, and finally
3. Traversing the right sub-tree.



```
Step 1: Repeat Steps 2 to 4 while TREE != NULL
Step 2:             INORDER(TREE -> LEFT)
Step 3:             Write TREE -> DATA
Step 4:             INORDER(TREE -> RIGHT)
        [END OF LOOP]
Step 5: END
```

In-order traversal is also called as symmetric traversal. In this algorithm, the left sub-tree is always traversed before the root node and the right sub-tree.

The word ‘in’ in the in-order specifies that the root node is accessed in between the left and the right sub-trees. In-order algorithm is also known as the LNR traversal algorithm (Left-Node-Right).



TRAVERSAL ORDER:
G, D, H, L, B, E, A, C, I, F, K, and J

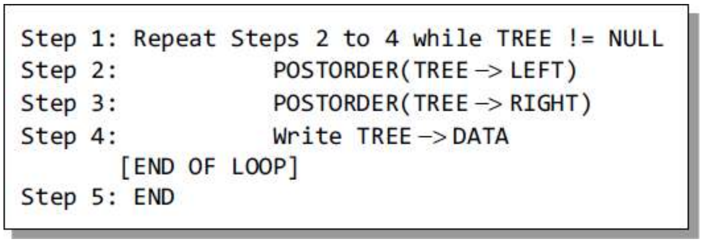
Post-order Traversal

To traverse a non-empty binary tree in post-order, the following operations are performed recursively at eachnode. The algorithm works by:

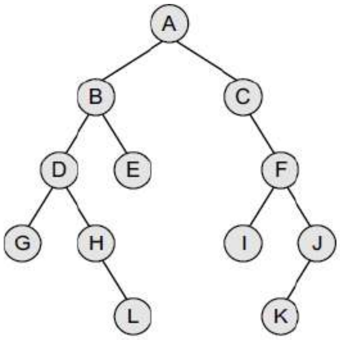
1. Traversing the left sub-tree,
2. Traversing the right sub-tree, and finally
3. Visiting the root node.

In this algorithm, the left sub-tree is always traversed before the right sub-tree and the root node. The word'post' in the post-order specifies that the root node is accessed after the left and the right sub-trees.

Post-order algorithm is also known as the LRN traversal algorithm (Left-Right-Node).



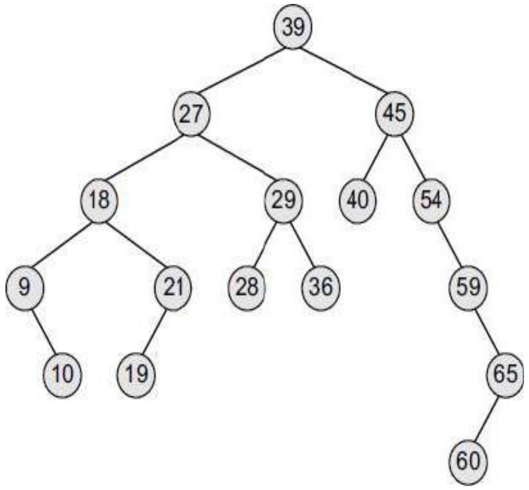
```
Step 1: Repeat Steps 2 to 4 while TREE != NULL
Step 2:           POSTORDER(TREE -> LEFT)
Step 3:           POSTORDER(TREE -> RIGHT)
Step 4:           Write TREE -> DATA
       [END OF LOOP]
Step 5: END
```



TRAVERSAL ORDER:
G, L, H, D, E, B, I, K, J, F, C, and A

5.5 BINARY SEARCH TREES

A binary search tree, also known as an ordered binary tree, is a variant of binary trees in which the nodes are arranged in an order. In a binary search tree, all the nodes in the left sub-tree have a value less than that of theroot node. Correspondingly, all the nodes in the right sub-tree have a value either equal to or greater than the root node. The same rule is applicable to every sub-tree in the tree. (Note that a binary search tree may or maynot contain duplicate values, depending on its implementation.)



The root node is 39. The left sub-tree of the root node consists of nodes 9, 10, 18, 19, 21, 27, 28, 29, and 36.

All these nodes have smaller values than the root node. The right sub-tree of the root node consists of nodes 40, 45, 54, 59, 60, and 65.

Recursively, each of the sub-trees also obeys the binary search tree constraint.

For example, in the left sub-tree of the root node, 27 is the root and all elements in its left sub-tree (9, 10, 18, 19, 21) are smaller than 27, while all nodes in its right sub-tree (28, 29, and 36) are greater than the root node's value.

Binary search trees also speed up the insertion and deletion operations. The tree has a speed advantage whenthe data in the structure changes rapidly.

Binary search trees are considered to be efficient data structures especially when compared with sorted linear arrays and linked lists. In a sorted array, searching can be done in O(log2n) time, but insertions and deletions are quite expensive. In contrast, inserting and deleting elements in a linked list is easier, but searching for an element is done in O(n) time.

However, in the worst case, a binary search tree will take O(n) time to search for an

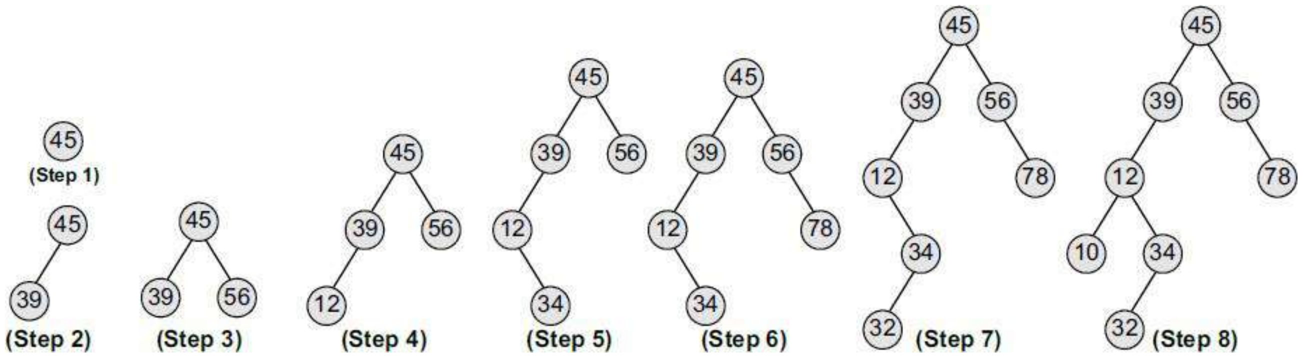
element.To summarize, a binary search tree is a binary tree with the following

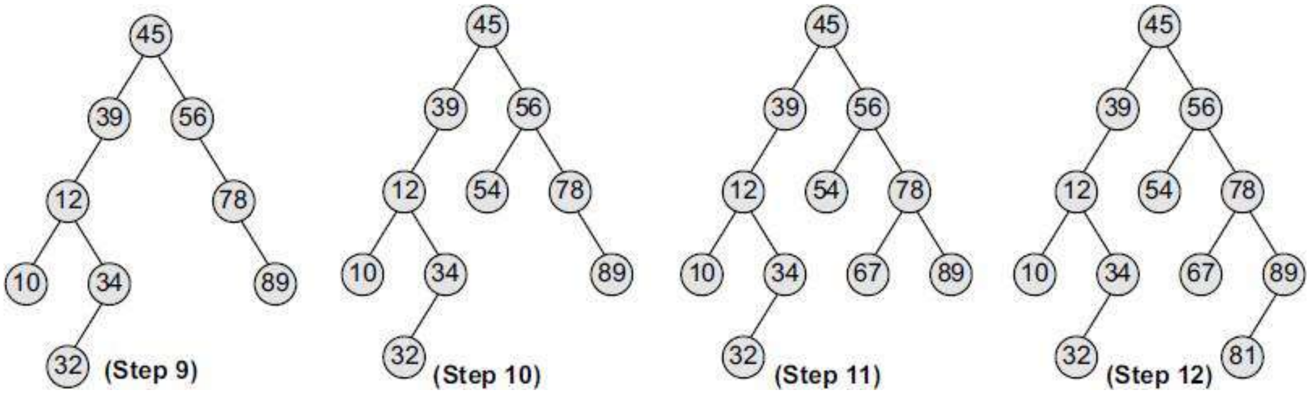
properties:

The left sub-tree of a node N contains values that are less than N's value.
The right sub-tree of a node N contains values that are greater than N's value.
Both the left and the right binary trees also satisfy these properties and, thus, are binary search trees.

Example:

Create a binary search tree using the following data elements: 45, 39, 56, 12, 34, 78, 32, 10, 89, 54, 67, and 81.

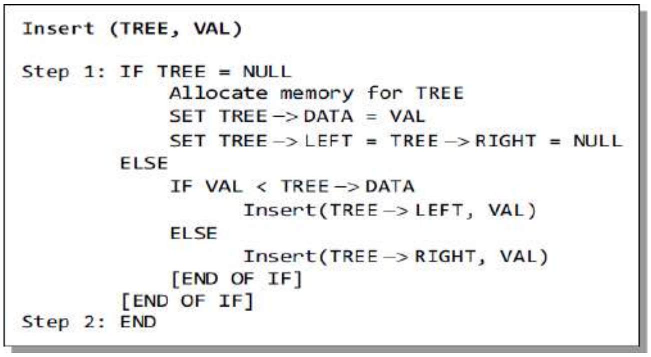




5.6 OPERATIONS ON BINARY SEARCH TREES

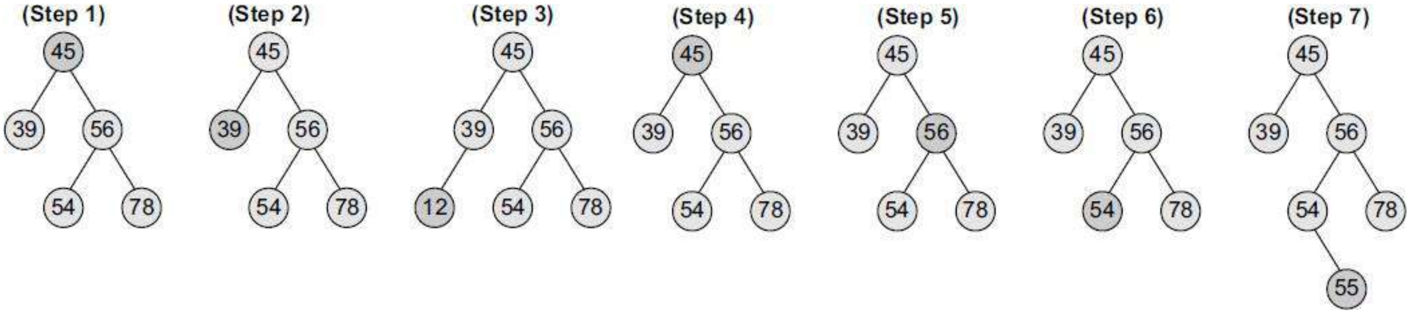
Inserting a New Node in a Binary Search Tree

The insert function is used to add a new node with a given value at the correct position in the binary search tree. Adding the node at the correct position means that the new node should not violate the properties of the binary search tree. The initial code for the insert function is similar to the search function. This is because we first find the correct position where the insertion has to be done and then add the node at that position. The insertion function changes the structure of the tree. Therefore, when the insert function is called recursively, the function should return the new tree pointer.
The insert function requires time proportional to the height of the tree in the worst case. It takes O(log n) time to execute in the average case and O(n) time in the worst case.



```
Insert (TREE, VAL)

Step 1: IF TREE = NULL
              Allocate memory for TREE
              SET TREE -> DATA = VAL
              SET TREE -> LEFT = TREE -> RIGHT = NULL
        ELSE
              IF VAL < TREE -> DATA
                    Insert(TREE -> LEFT, VAL)
              ELSE
                    Insert(TREE -> RIGHT, VAL)
              [END OF IF]
        [END OF IF]
Step 2: END
```

Example: Inserting nodes with values 12 and 55 in the given binary search tree

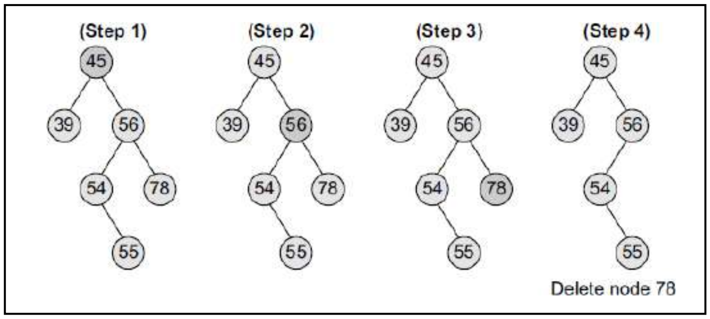


Deleting a Node from a Binary Search Tree

The delete function deletes a node from the binary search tree. However, utmost care should be taken that theproperties of the binary search tree are not violated and nodes are not lost in the process.

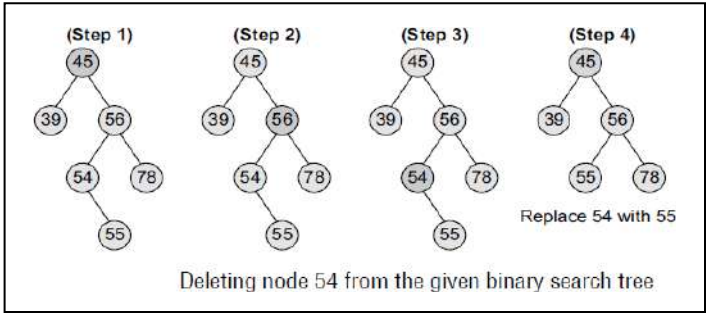
Case 1: Deleting a Node that has No ChildrenCase 2: Deleting a Node with One Child

Case 3: Deleting a Node with Two Children



If we have to delete node 78, we can simply remove this node without any issue.
This is the simplest case of deletion.

Case1: Deleting a Node that has No Children



Deleting node 54 from the given binary search tree

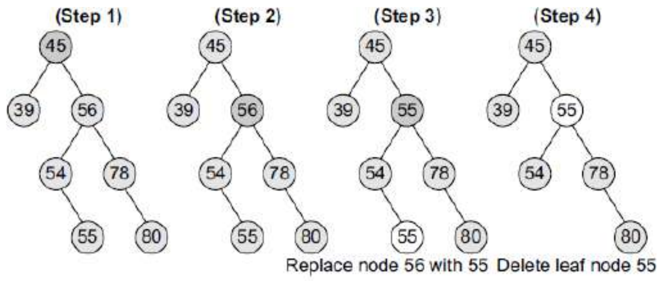
To handle this case, the node’s child is set as the child of the node’s parent. In other words, replace the node with its child. Now, if the node is the left child of its parent, the node’s child becomes the left child of the node’s parent. Correspondingly, if the node is the right child of its parent, the node’s child becomes the right child of the node’s parent.

Look at the binary search tree shown in figure and see how deletion of node 54 is handled.

Case 2: Deleting a Node with One Child

To handle this case, replace the node’s value with its in-order predecessor (largest value in the left sub-tree) or in-order successor (smallest value in the right sub-tree).
The in-order predecessor or the successor can then be deleted using any of the above cases.

Look at the binary search tree given in figure and see how deletion of node with value 56 is handled.



Deleting node 56 from the given binary search tree

5.7 BALANCED BINARY TREES- AVL TREES

AVL tree is a self-balancing binary search tree invented by G.M. Adelson-Velsky and E.M. Landis in 1962. The tree is named AVL in honour of its inventors. In an AVL tree, the heights of the two sub-trees of a node may differ by at most one. Due to this property, the AVL tree is also known as a height-balanced tree. The keyadvantage of using an AVL tree is that it takes O(log n) time to perform search, insert, and delete operations inan average case as well as the worst case because the height of the tree is limited to O(log n).

The structure of an AVL tree is the same as that of a binary search tree but with a little difference. In its structure, it stores an additional variable called the Balance Factor. Thus, every node has a balance factor associated with it. The balance factor of a node is calculated by subtracting the height of its right sub-tree fromthe height of its left sub-tree. A binary search tree in which every node has a balance factor of –1, 0, or 1 issaid to be height balanced. A node with any other balance factor is considered to be unbalanced and requires rebalancing of the tree.

Balance factor = Height (left sub-tree) – Height (right sub-tree)

- If the balance factor of a node is 1, then it means that the left sub-tree of the tree is one level higher than that of the right sub-tree. Such a tree is therefore called as a left-heavy tree.
- If the balance factor of a node is 0, then it means that the height of the left sub-tree (longest path in the left sub-tree) is equal to the height of the right sub-tree.
- If the balance factor of a node is –1, then it means that the left sub-tree of the tree is one level lower than that of the right sub-tree. Such a tree is therefore called as a right-heavy tree.

5.7.1 Operations on AVL Trees

Searching for a Node in an AVL Tree

Searching in an AVL tree is performed exactly the same way as it is performed in a binary search tree. Due to the height-balancing of the tree, the search operation takes O (log n) time to complete. Since the operationdoes not modify the structure of the tree, no special provisions are required.

Inserting a New Node in an AVL Tree

Insertion in an AVL tree is also done in the same way as it is done in a binary search tree. In the AVL tree, thenew node is always inserted as the leaf node. But the step of insertion is usually followed by an additional stepof rotation. Rotation is done to restore the balance of the tree.

However, if insertion of the new node does not disturb the balance factor, that is, if the balance factor of every node is still –1, 0, or 1, then rotations are not required.

To perform rotation, our first task is to find the critical node. Critical node is the nearest ancestor node on the path from the inserted node to the root whose balance factor is neither –1, 0, nor 1.

The second task in rebalancing the tree is to determine which type of rotation has to be done. There are four types of rebalancing rotations and application of these rotations depends on the position of the inserted node with reference to the critical node.

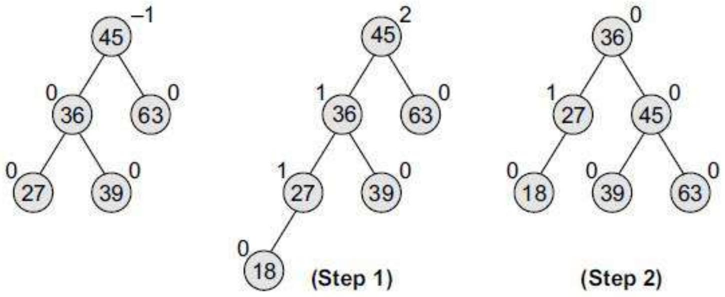
The four categories of rotations are:

- **LL rotation:** The new node is inserted in the left sub-tree of the left sub-tree of the critical node.
- **RR rotation:** The new node is inserted in the right sub-tree of the right sub-tree of the critical node.
- **LR rotation:** The new node is inserted in the right sub-tree of the left sub-tree of the critical node.
- **RL rotation:** The new node is inserted in the left sub-tree of the right sub-tree of the critical node.

LL Rotation

Example Consider the AVL tree given in Fig. and insert 18 into it.

Solution



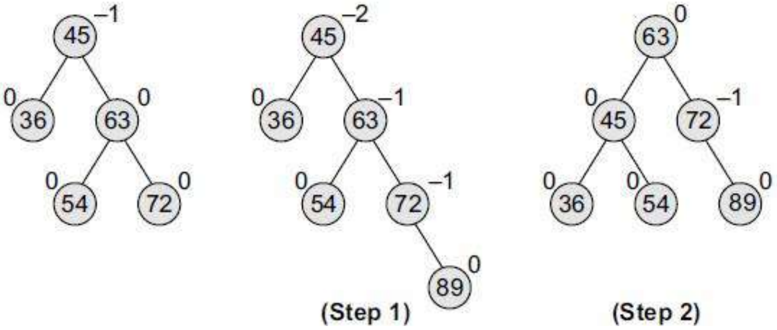
(Step 1) (Step 2)

Figure AVL tree

RR Rotation

Example Consider the AVL tree given in Fig. and insert 89 into it.

Solution



(Step 1) (Step 2)

LR and RL Rotations

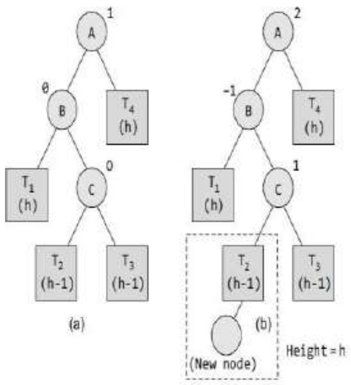


Figure LR rotation in an AVL tree

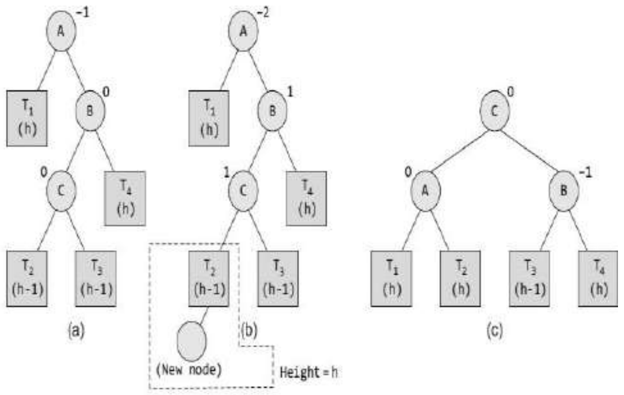
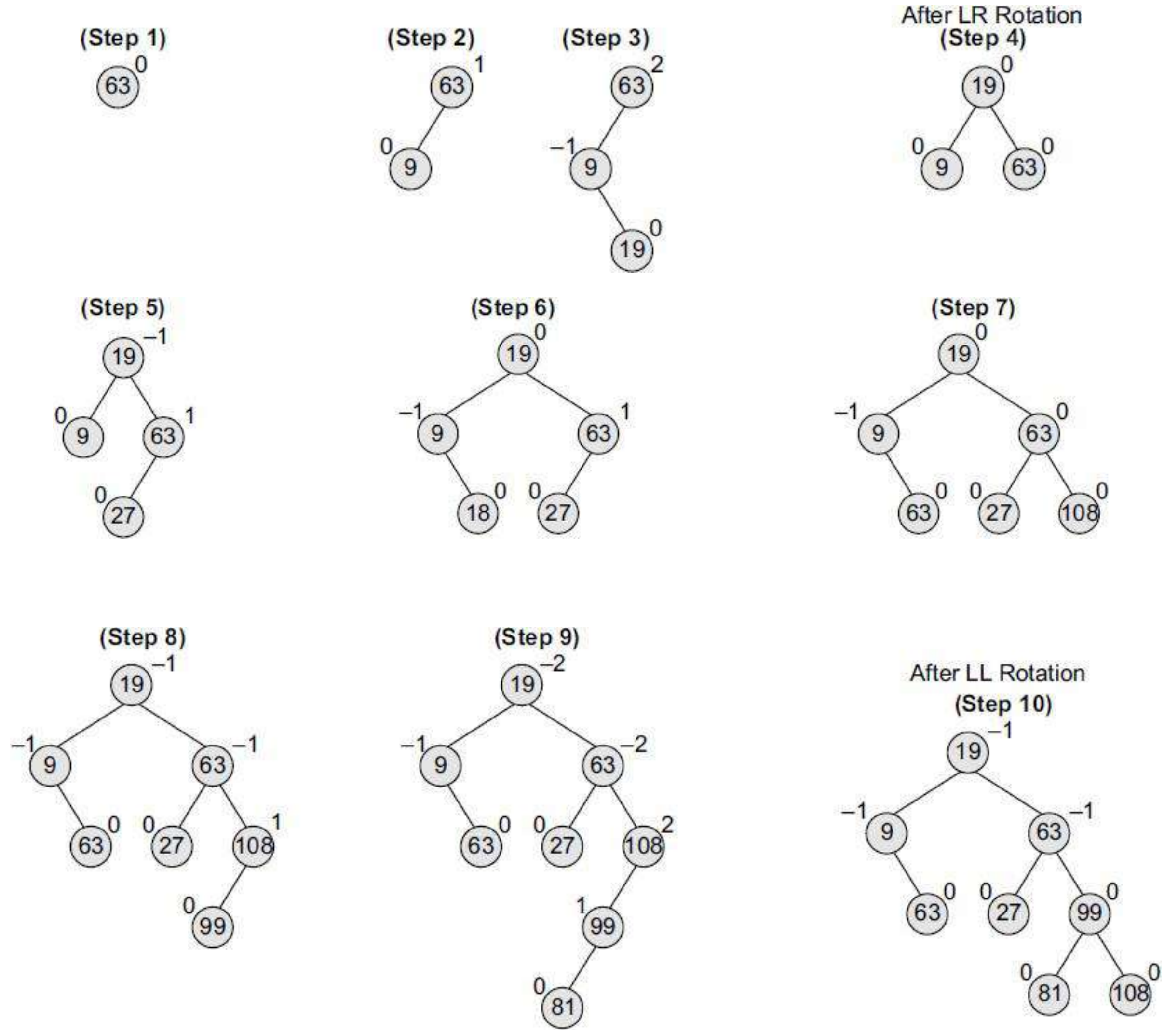


Figure RL rotation in an AVL tree

Example: Construct an AVL tree by inserting the following elements in the given order.
63, 9, 19, 27, 18, 108, 99, 81.



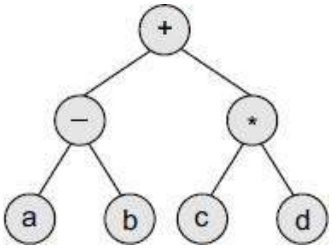
Deleting a Node from an AVL Tree

Deletion of a node in an AVL tree is similar to that of binary search trees. But it goes one step ahead. Deletionmay disturb the AVLness of the tree, so to rebalance the AVL tree, we need to perform rotations.

5.8 APPLICATIONS OF TREES

- Trees are used to store simple as well as complex data. Here simple means an integer value, character value and complex data means a structure or a record.
- Trees are often used for implementing other types of data structures like hash tables, sets, and maps.
- A self-balancing tree, Red-black tree is used in kernel scheduling, to preempt massively multiprocessor computer operating system use.
- Another variation of tree, B-trees are prominently used to store tree structures on disc. They are used to index a large number of records.
- B-trees are also used for secondary indexes in databases, where the index facilitates a select operation to answer some range criteria.
- Trees are an important data structure used for compiler construction.
- Trees are also used in database design.
- Trees are used in file system directories.
- Trees are also widely used for information storage and retrieval in symbol tables.

5.8.1 Expression Trees

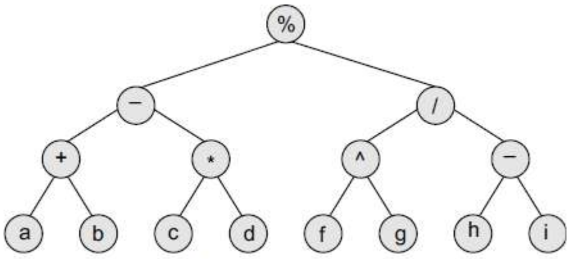


Binary trees are widely used to store algebraic expressions. For example, consider the algebraic expression given as:

Exp = (a – b) + (c * d)

This expression can be represented using a binary tree as shown in Figure.

Given an expression, Exp = ((a + b) – (c * d)) % ((e ^f) / (g – h)), construct the corresponding binary tree.



Expression tree

5.8.2 Heap Sort

Heap: Recall that a heap is a complete binary tree such that the weight of every node is less than the weightsof its children

A heap with n elements can be conveniently represented as the first n elements of an array. Furthermore, the children of a[i] can be found in a[2i] (left child) and a[2i + 1] (right child)

Steps:

1. Consider the values of the elements as priorities and build the heap tree.
2. Start deleteMin operations, storing each deleted element at the end of the heap array.

After performing step 2, the order of the elements will be opposite to the order in the heap tree.
Hence, if we want the elements to be sorted in ascending order, we need to build the heap tree in descending order - the greatest element will have the highest priority.

Note that we use only one array, treating its parts differently:

a. When building the heap tree, part of the array will be considered as the heap, and the rest part - theoriginal array.
b. When sorting, part of the array will be the heap, and the rest part - the sorted array.

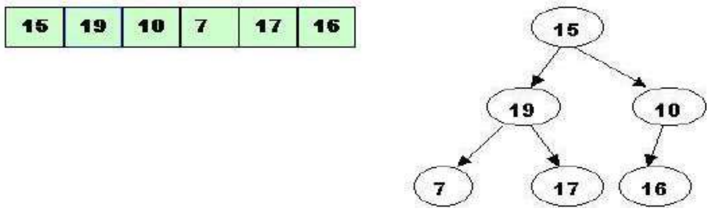
Example:

Given an array of 6 elements: 15, 19, 10, 7, 17, and 16 sort it in ascending order using heap sort

Here is the array: 15, 19, 10, 7, 17, and 6

Building the heap tree:

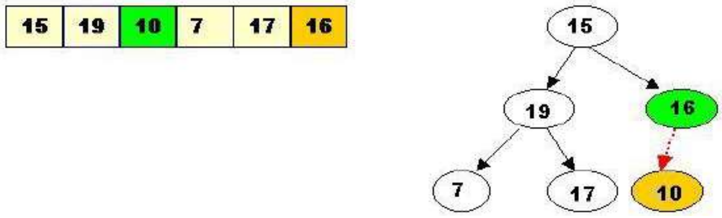
The array represented as a tree, complete but not ordered:



15	19	10	7	17	16

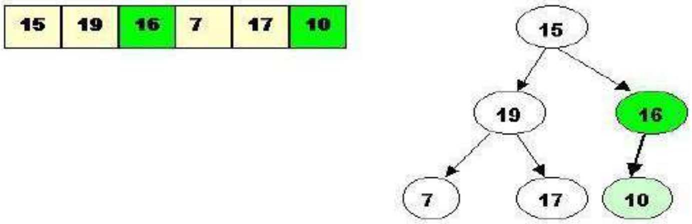
Start with the rightmost node at height 1, the node at position 3 =

Size/2.It has one greater child and has to be percolated down:



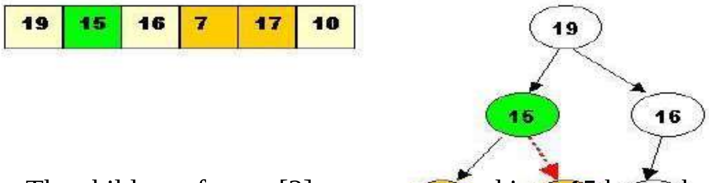
15	19	10	7	17	16

After processing array [3] the situation is:

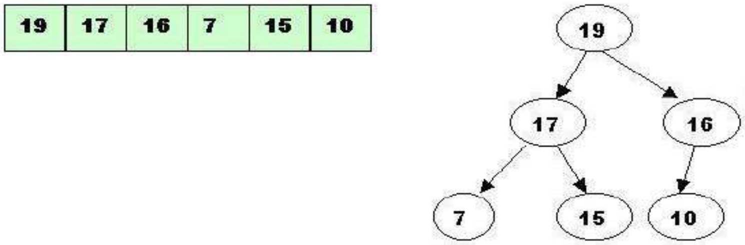


Next come array [2]. Its children are smaller, so no percolation is needed.

The last node to be processed is array [1]. Its left child is the greater of the children.The item at array [1] has to be percolated down to the left, swapped with array [2]. As a result the situation is:

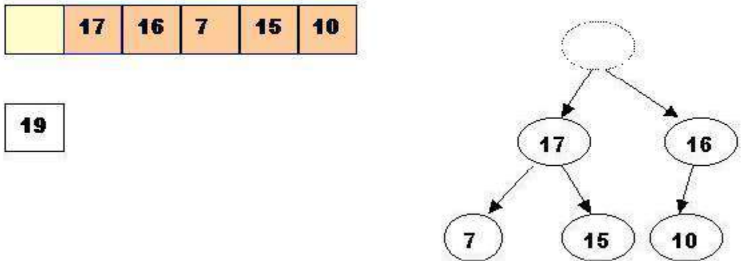


The children of array [2] are greater, and item 15 has to be moved down further, swapped with array [5].



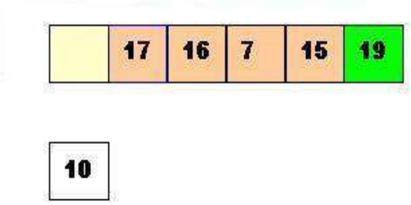
Now the tree is ordered, and the binary heap is built.**Sorting - performing deleteMax operations:** Delete the top element 19.

Store 19 in a temporary place, a hole is created at the top

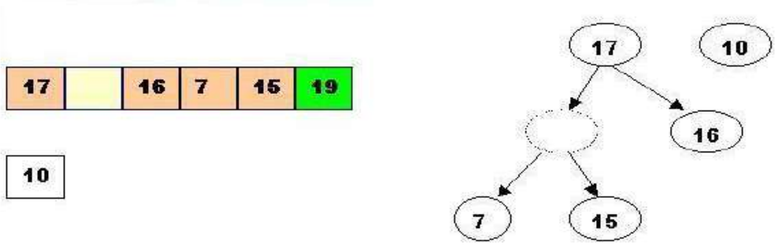


Swap 19 with the last element of the heap.

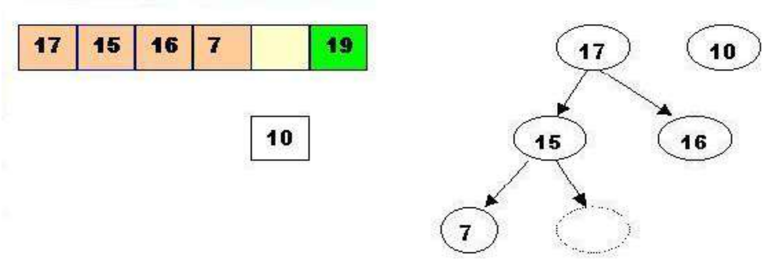
As 10 will be adjusted in the heap, its cell will no longer be a part of the heap.Instead it becomes a cell from the sorted array



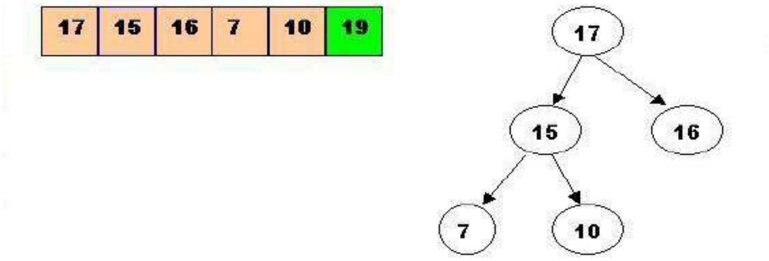
Percolate down the hole



Percolate once more (10 is less that 15, so it cannot be inserted in the previous hole)

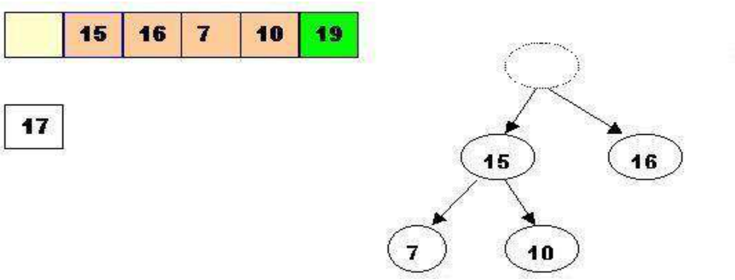


Now 10 can be inserted in the hole



Delete Max the top element 17

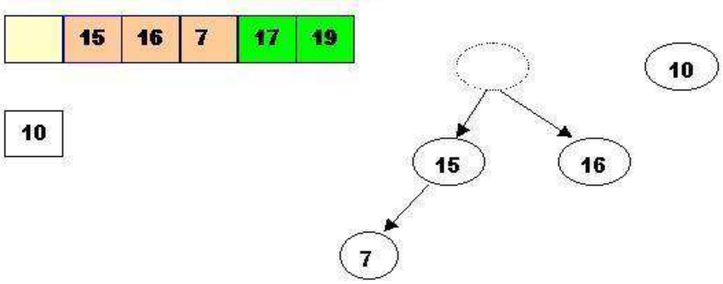
Store 17 in a temporary place, a hole is created at the top



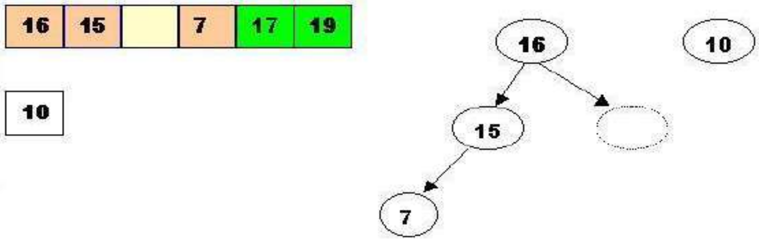
Swap 17 with the last element of the heap.

As 10 will be adjusted in the heap, its cell will no longer be a part

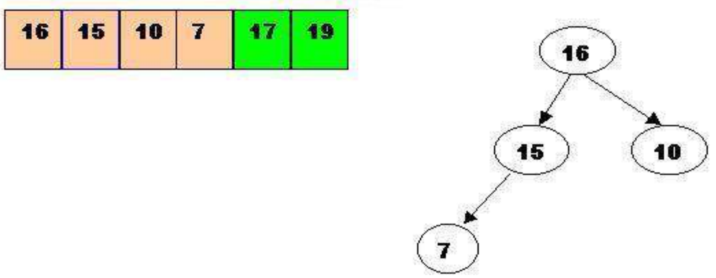
of the heap.Instead it becomes a cell from the sorted array



The element 10 is less than the children of the hole, and we percolate the hole down:

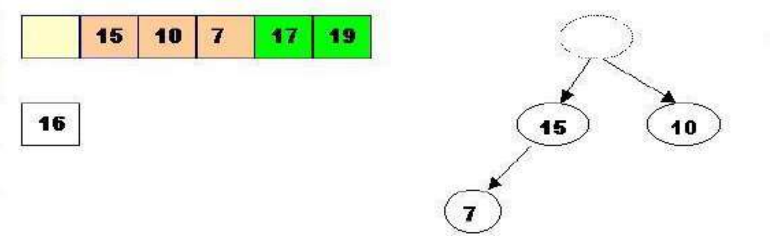


Insert 10 in the hole



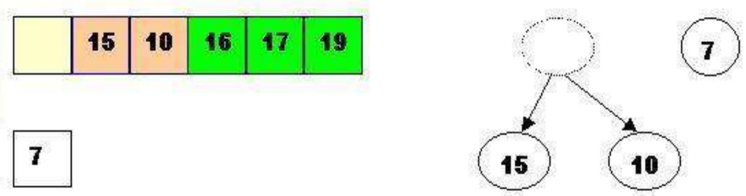
DeleteMax 16

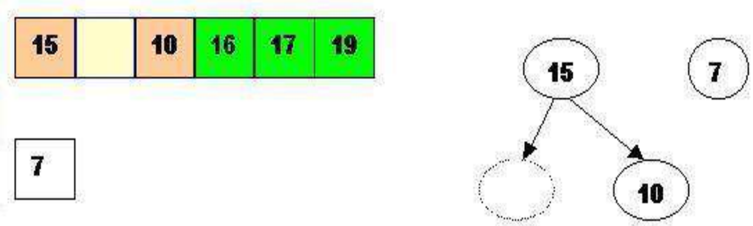
Store 16 in a temporary place, a hole is created at the top



Swap 16 with the last element of the heap.

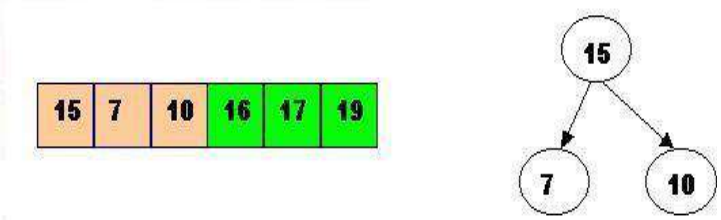
As 7 will be adjusted in the heap, its cell will no longer be a part of the heap.Instead it becomes a cell from the sorted array





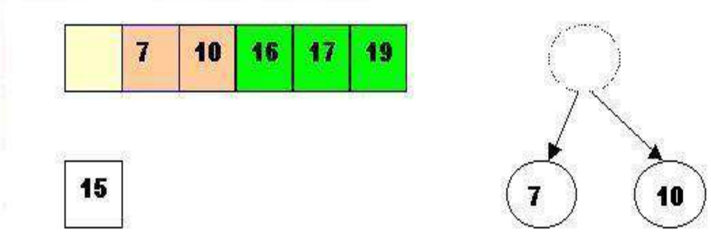
than the children of the hole)

Insert 7 in the hole



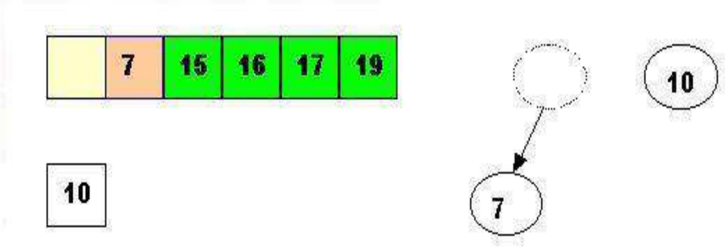
DeleteMax the top element 15

Store 15 in a temporary location, a hole is created.

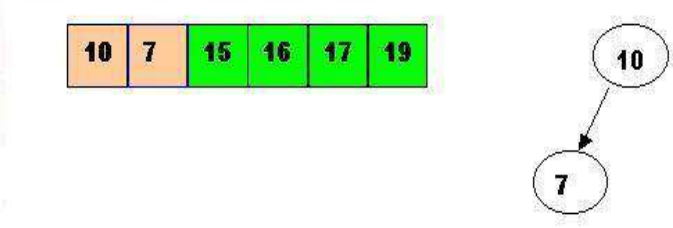


Swap 15 with the last element of the heap.

As 10 will be adjusted in the heap, its cell will no longer be a part of the heap.Instead it becomes a position from the sorted array

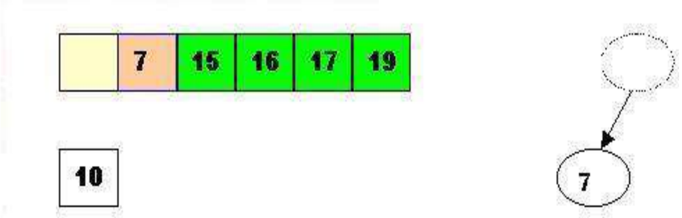


Store 10 in the hole (10 is greater than the children of the hole)



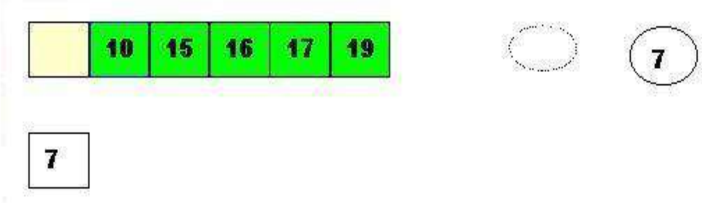
DeleteMax the top element 10

Remove 10 from the heap and store it into a temporary location.

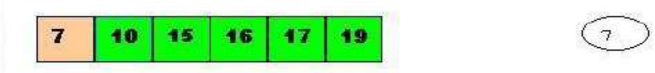


Swap 10 with the last element of the heap.

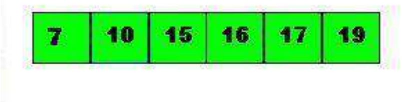
As 7 will be adjusted in the heap, its cell will no longer be a part of the heap. Instead it becomes a cell fromthe sorted array



Store 7 in the hole (as the only remaining element in the heap



7 is the last element from the heap, so now the array is sorted



The HEAPSORT procedure takes time O(*n log n*), since the call to BUILD_HEAP takes time *O*(*n*) and eachof the *n* -1 calls to Heapify takes time O(log *n*).

M.Sc. (Computer Science)

SEMESTER-2

COURSE: DATA STRUCTURE AND ALGORITHMS

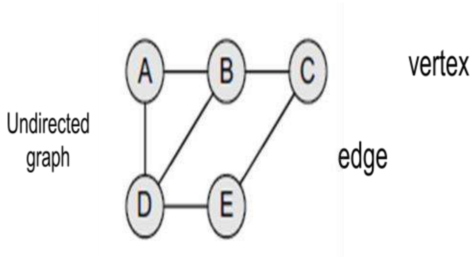
UNIT 6: GRAPHS

6.1 BASIC CONCEPTS

A graph is an abstract data structure that is used to implement the mathematical concept of graphs. It is basically a collection of vertices (also called nodes) and edges that connect these vertices.

Definition

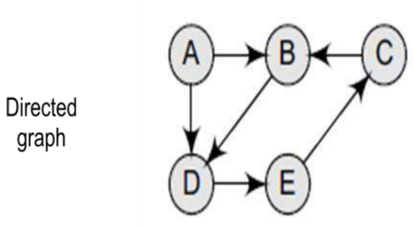
A graph G is defined as an ordered set (V, E), where V(G) represents the set of vertices and E(G) representsthe edges that connect these vertices.



A graph can be directed or undirected. In an undirected graph, edges do not have any direction associated with them.

That is, if an edge is drawn between nodes A and B, then the nodes can be traversed from A to B as well as from B to A.

A graph G with V(G) = {A, B, C, D and E} and
E(G) = {(A, B), (B, C), (A, D), (B, D), (D, E), (C, E)}.



In a directed graph, edges form an ordered pair. If there is an edge from A to B, then there is a path from A to B but not from B to A.

The edge (A, B) is said to initiate from node A (also known as initial node) and terminate at node B (terminal node).

A graph G with V(G) = {A, B, C, D and E} and
E(G) = {(A, B), (C, B), (A, D), (B, D), (D, E), (E, C)}.

6.1.1 Graph Terminology

Adjacent nodes or **neighbours** For every edge, e = (u, v) that connects nodes u and v, the nodes u and v arethe end-points and are said to be the adjacent nodes or neighbours.

Degree of a node Degree of a node u, deg(u), is the total number of edges containing the node u. If deg(u) =0, it means that u does not belong to any edge and such a node is known as an isolated node.

Regular graph It is a graph where each vertex has the same number of neighbours. That is, every node hasthe same degree. A regular graph with vertices of degree k is called a k–regular graph or a regular graph of degree k.

Path A path P written as P = {v0, v1, v2, ..., vn), of length n from a node u to v is defined as a sequence of(n+1) nodes. Here, u = v0, v = vn and vi–1 is adjacent to vi for i = 1, 2, 3,..., n.

Closed path A path P is known as a closed path if the edge has the same end-points. That is, if v0 = vn.

Simple path A path P is known as a simple path if all the nodes in the path are distinct with an exception that v0 may be equal to vn. If v0 = vn, then the path is called a closed simple path.

Cycle A path in which the first and the last vertices are same. A simple cycle has no repeated edges or vertices(except the first and last vertices).

Connected graph A graph is said to be connected if for any two vertices (u, v) in V there is a path from u to

v. That is to say that there are no isolated nodes in a connected graph. A connected graph that does not haveany cycle is called a tree. Therefore, a tree is treated as a special graph.

Complete graph A graph G is said to be complete if all its nodes are fully connected. That is, there is a path from one node to every other node in the graph. A complete graph has n(n–1)/2 edges, where n is the number of nodes in G.

Labelled graph or **weighted graph** A graph is said to be labelled if every edge in the graph is assigned some data. In a weighted graph, the edges of the graph are assigned some weight or length. The weight of an edge denoted by w(e) is a positive value which indicates the cost of traversing the edge.

Multiple edges Distinct edges which connect the same end-points are called multiple edges. That is, e = (u, v)and e' = (u, v) are known as multiple edges of G.

Loop An edge that has identical end-points is called a loop. That is, e =

(u, u). **Multi-graph** A graph with multiple edges and/or loops is

called a multi-graph.**Size of a graph** The size of a graph is the total

number of edges in it.

BI-CONNECTED components

A vertex v of G is called an articulation point, if removing v along with the edges incident on v, results in a graph that has at least two connected components.

A bi-connected graph is defined as a connected graph that has no articulation vertices. That is, a bi-connected graph is connected and non-separable in the sense that even if we remove any vertex from the graph, the resultant graph is still connected.

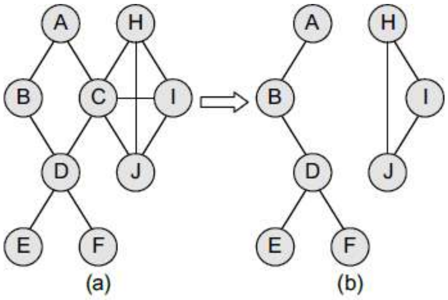
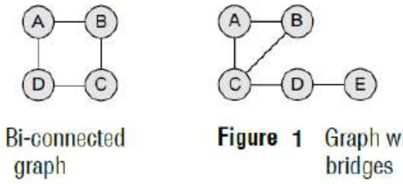


Figure Non bi-connected graph

A bi-connected undirected graph is a connected graph that cannot be broken into disconnected pieces by deleting any single vertex.

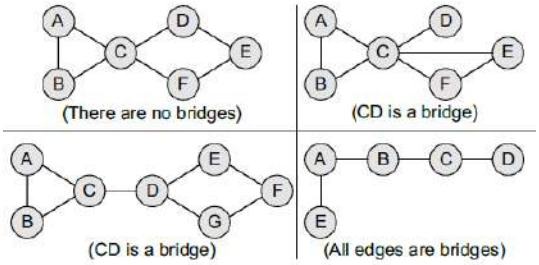
In a bi-connected directed graph, for any two vertices v and w, there are two directed paths from v to w which have no vertices in common other than v and w.

Note that the graph shown in Fig. (a) is not a bi-connected graph, as deleting vertex C from the graph results in two disconnected components of the original graph (Fig. (b)).



Bi-connected graph

Figure 1 Graph with bridges



(There are no bridges)

(CD is a bridge)

(CD is a bridge)

(All edges are bridges)

Figure 2 Graph with bridges

As for vertices, there is a related concept for edges. An edge in a graph is called a bridge if removing that edge results in a disconnected graph.

Also, an edge in a graph that does not lie on a cycle is a bridge. This means that a bridge has at least one articulation point at its end, although it is not necessary that the articulation point is linked to a bridge. Look at the graph shown in Fig.1.

In the graph, CD and DE are bridges. Consider some more examples shown in Fig. 2.

6.2 REPRESENTATION OF GRAPHS

There are three common ways of storing graphs in the computer's memory.They are:

- Sequential representation by using an adjacency matrix.
- Linked representation by using an adjacency list that stores the neighbours of a node using a linkedlist.
- Adjacency multi-list which is an extension of linked representation.

6.2.1 Adjacency Matrix Representation

An adjacency matrix is used to represent which nodes are adjacent to one another. By definition, two nodesare said to be adjacent if there is an edge connecting them.

In a directed graph G, if node v is adjacent to node u, then there is definitely an edge from u to v. That is, if vis adjacent to u, we can get from u to v by traversing one edge. For any graph G having n nodes, the adjacencymatrix will have the dimension of n × n.

In an adjacency matrix, the rows and columns are labelled by graph vertices. An entry a_{ij} in the adjacency matrix will contain 1, if vertices v_i and v_j are adjacent to each other. However, if the nodes are not adjacent, a_{ij} will be set to zero. It is summarized in Figure.

Since an adjacency matrix contains only 0s and 1s, it is called a bit matrix or a Boolean matrix. The entries in the matrix depend on the ordering of the nodes in G. Therefore, a change in the order of nodes will result in a different adjacency matrix.

From the above examples, we can draw the following conclusions:
For a simple graph (that has no loops), the adjacency matrix has 0s on the diagonal.
The adjacency matrix of an undirected graph is symmetric.
The memory use of an adjacency matrix is O(n^2), where n is the number of nodes in the graph.
Number of 1s (or non-zero entries) in an adjacency matrix is equal to the number of edges in the graph.
The adjacency matrix for a weighted graph contains the weights of the edges connecting the nodes.

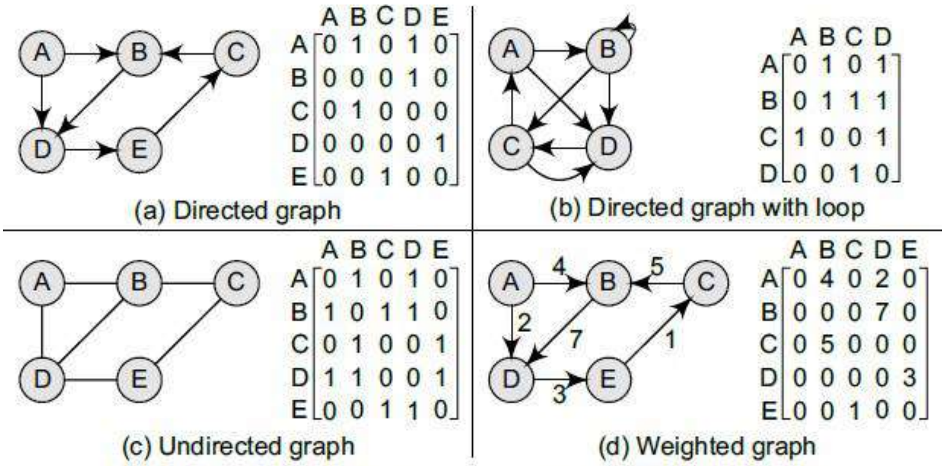


Figure Graphs and their corresponding adjacency matrices

6.2.2 Adjacency List Representation

An adjacency list is another way in which graphs can be represented in the computer's memory.
This structure consists of a list of all nodes in G. Furthermore, every node is in turn linked to its own list thatcontains the names of all other nodes that are adjacent to it.

The key advantages of using an adjacency list are:
6.2.2.1 It is easy to follow and clearly shows the adjacent nodes of a particular node.
6.2.2.2 It is often used for storing graphs that have a small-to-moderate number of edges. That is, an adjacencylist is preferred for representing sparse graphs in the computer's memory; otherwise, an adjacency matrix is a good choice.
6.2.2.3 Adding new nodes in G is easy and straightforward when G is represented using an adjacency list.Adding new nodes in an adjacency matrix is a difficult task, as the size of the matrix needs to be changed and existing nodes may have to be reordered.

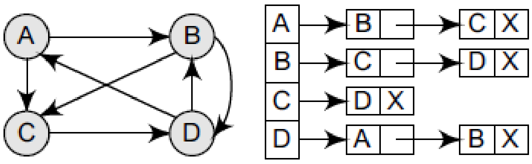
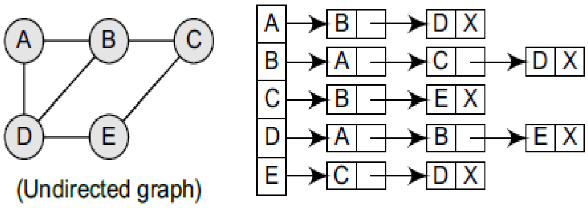
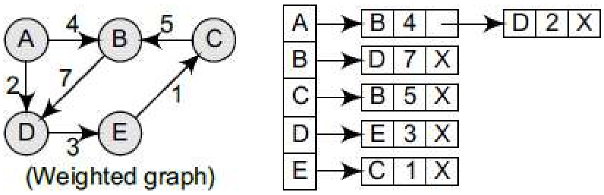


Figure Graph G and its adjacency list



(Undirected graph)



(Weighted graph)

Figure Adjacency list for an undirected graph and a weighted graph

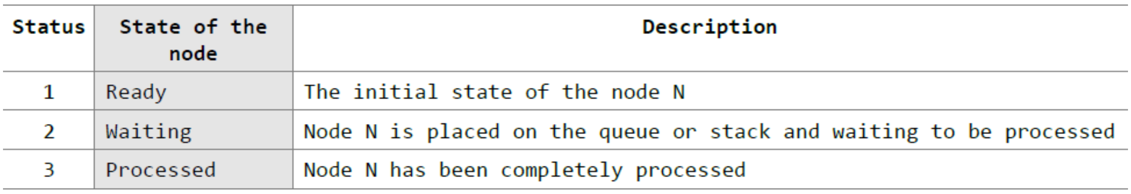
6.3 GRAPH TRAVERSAL ALGORITHMS

In this section, we will discuss how to traverse graphs. By traversing a graph, we mean the method of examining the nodes and edges of the graph. There are two standard methods of graph traversal which we will discuss in this section.

These two methods are:

1. Breadth-first search
2. Depth-first search

While breadth-first search uses a queue as an auxiliary data structure to store nodes for further processing, the depth-first search scheme uses a stack. But both these algorithms make use of a variable STATUS. During the execution of the algorithm, every node in the graph will have the variable STATUS set to 1 or 2, depending onits current state.

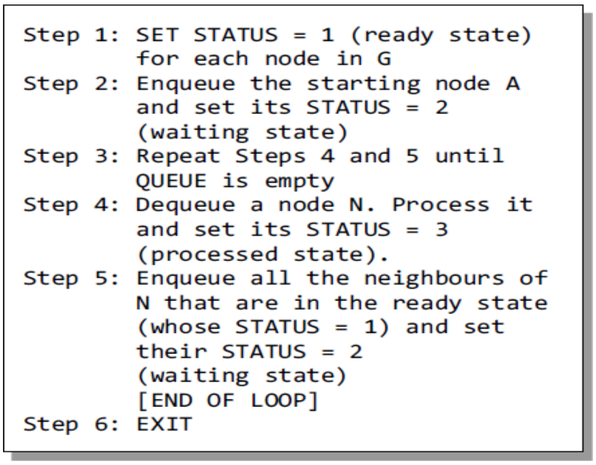


Status	State of the node	Description
1	Ready	The initial state of the node N
2	Waiting	Node N is placed on the queue or stack and waiting to be processed
3	Processed	Node N has been completely processed

6.4 Breadth-first search (BFS):

Breadth-first search (BFS) is a graph search algorithm that begins at the root node and explores all the neighbouring nodes. Then for each of those nearest nodes, the algorithm explores their unexplored neighbour nodes, and so on, until it finds the goal.

That is, we start examining the node A and then all the neighbours of A are examined. In the next step, we examine the neighbours of neighbours of A, so on and so forth. This means that we need to track the neighbours of the node and guarantee that every node in the graph is processed and no node is processed morethan once. This is accomplished by using a queue that will hold the nodes that are waiting for further processing and a variable STATUS to represent the current state of the node.



```
Step 1: SET STATUS = 1 (ready state)
        for each node in G
Step 2: Enqueue the starting node A
        and set its STATUS = 2
        (waiting state)
Step 3: Repeat Steps 4 and 5 until
        QUEUE is empty
Step 4: Dequeue a node N. Process it
        and set its STATUS = 3
        (processed state).
Step 5: Enqueue all the neighbours of
        N that are in the ready state
        (whose STATUS = 1) and set
        their STATUS = 2
        (waiting state)
        [END OF LOOP]
Step 6: EXIT
```

Algorithm for breadth-first search

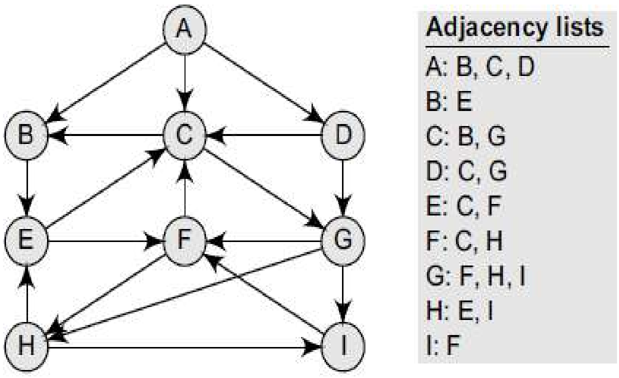


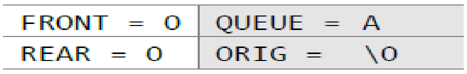
Figure Graph G and its adjacency list

Solution:

- The minimum path P can be found by applying the breadth-first search algorithm that begins at city Aand ends when I is encountered.
- During the execution of the algorithm, we use two arrays: QUEUE and ORIG.
- While QUEUE is used to hold the nodes that have to be processed, ORIG is used to keep track of theorigin of each edge. Initially, FRONT = REAR = –1.
- Initially, FRONT = REAR = –1.

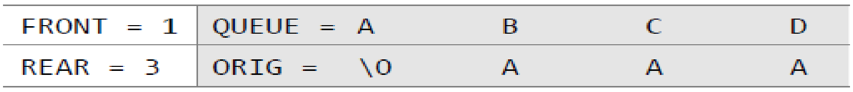
The algorithm for this is as follows:

(a) Add A to QUEUE and add NULL to ORIG.



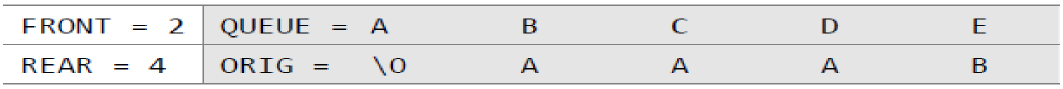
FRONT = 0	QUEUE = A
REAR = 0	ORIG = \0

(b) Dequeue a node by setting FRONT = FRONT + 1 (remove the FRONT element of QUEUE) and enqueuethe neighbours of A. Also, add A as the ORIG of its neighbours.



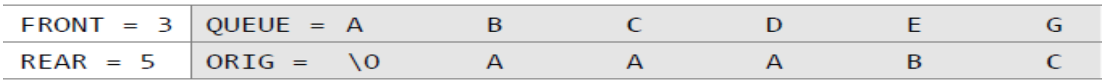
FRONT = 1	QUEUE = A	B	C	D
REAR = 3	ORIG = \0	A	A	A

(c) Dequeue a node by setting FRONT = FRONT + 1 and enqueue the neighbours of B. Also, add B as theORIG of its neighbours.



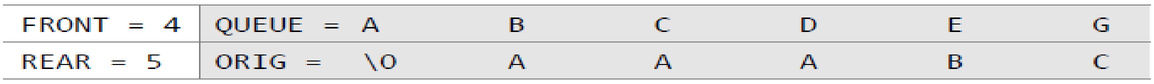
FRONT = 2	QUEUE = A	B	C	D	E
REAR = 4	ORIG = \0	A	A	A	B

(d) Dequeue a node by setting FRONT = FRONT + 1 and enqueue the neighbours of C. Also, add C as the ORIG of its neighbours. Note that C has two neighbours B and G. Since B has already been added to the queue and it is not in the Ready state, we will not add B and only add G.



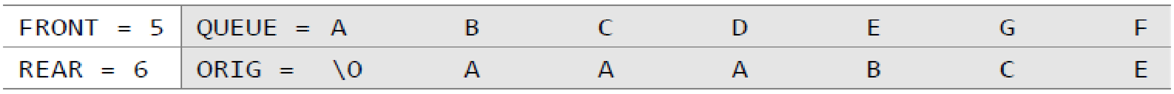
FRONT = 3	QUEUE = A	B	C	D	E	G
REAR = 5	ORIG = \0	A	A	A	B	C

(e) Dequeue a node by setting FRONT = FRONT + 1 and enqueue the neighbours of D. Also, add D as the ORIG of its neighbours. Note that D has two neighbours C and G. Since both of them have already been added to the queue and they are not in the Ready state, we will not add them again.



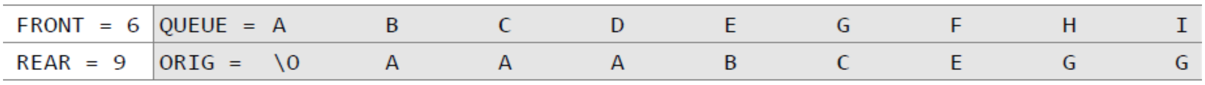
FRONT = 4	QUEUE = A	B	C	D	E	G
REAR = 5	ORIG = \0	A	A	A	B	C

(f) Dequeue a node by setting FRONT = FRONT + 1 and enqueue the neighbours of E. Also, add E as the ORIG of its neighbours. Note that E has two neighbours C and F. Since C has already been added to the queueand it is not in the Ready state, we will not add C and add only F.



FRONT = 5	QUEUE = A	B	C	D	E	G	F
REAR = 6	ORIG = \0	A	A	A	B	C	E

(g) Dequeue a node by setting FRONT = FRONT + 1 and enqueue the neighbours of G. Also, add G as theORIG of its neighbours. Note that G has three neighbours F, H, and I.



FRONT = 6	QUEUE = A	B	C	D	E	G	F	H	I
REAR = 9	ORIG = \0	A	A	A	B	C	E	G	G

Since F has already been added to the queue, we will only add H and I. As I is our final destination, we stop the execution of this algorithm as soon as it is encountered and added to the QUEUE. Now, backtrack from I using ORIG to find the minimum path P. Thus, we have Path P as A -> C -> G -> I.

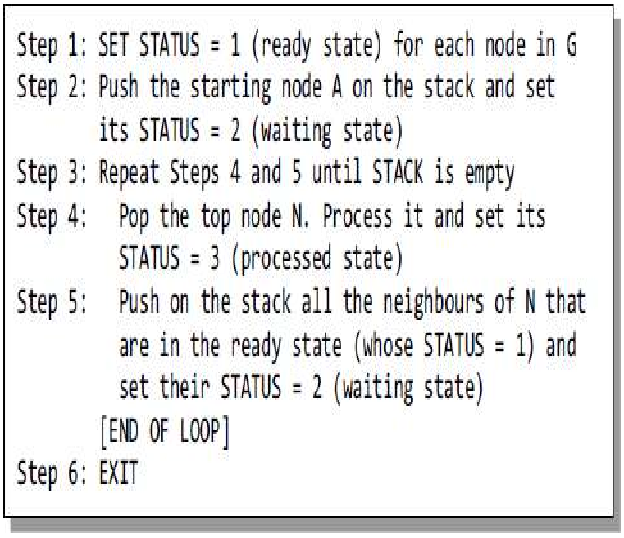
The time complexity can also be expressed as O(| E | + | V |)

6.5 DEPTH-FIRST SEARCH (DFS)

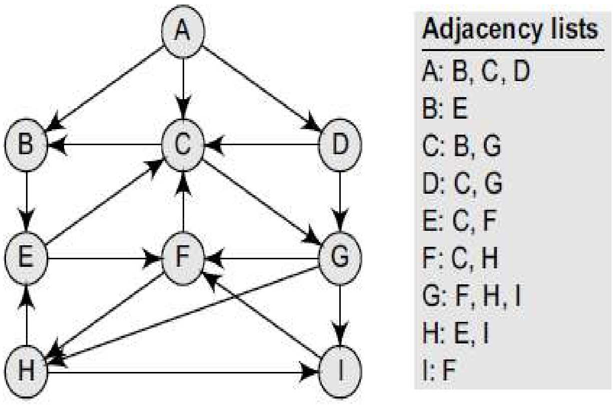
The depth-first search algorithm progresses by expanding the starting node of G and then going deeper and deeper until the goal node is found, or until a node that has no children is encountered. When a dead-end is reached, the algorithm backtracks, returning to the most recent node that has not been completely explored.

In other words, depth-first search begins at a starting node A which becomes the current node.

Then, it examines each node N along a path P which begins at A. That is, we process a neighbor of A, then a neighbour of neighbour of A, and so on. During the execution of the algorithm, if we reach a path that has a node N that has already been processed, then we backtrack to the current node. Otherwise, the unvisited (unprocessed) node becomes the current node.



```
Step 1: SET STATUS = 1 (ready state) for each node in G
Step 2: Push the starting node A on the stack and set
        its STATUS = 2 (waiting state)
Step 3: Repeat Steps 4 and 5 until STACK is empty
Step 4:    Pop the top node N. Process it and set its
           STATUS = 3 (processed state)
Step 5:    Push on the stack all the neighbours of N that
           are in the ready state (whose STATUS = 1) and
           set their STATUS = 2 (waiting state)
        [END OF LOOP]
Step 6: EXIT
```



Adjacency lists

A: B, C, D
B: E
C: B, G
D: C, G
E: C, F
F: C, H
G: F, H, I
H: E, I
I: F

Figure Graph G and its adjacency list

Consider the graph G given in Figure. The adjacency list of G is also given. Suppose we want to print all the nodes that can be reached from the node H (including H itself). One alternative is to use a depth-first search ofG starting at node H. The procedure can be explained here.

(a) Push H onto the stack.

STACK: H

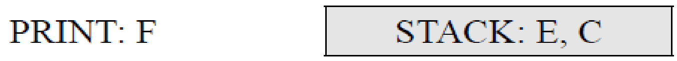
(b) Pop and print the top element of the STACK, that is, H. Push all the neighbours of H onto the stackthat are in the ready state. The STACK now becomes

PRINT: H STACK: E, I

(c) Pop and print the top element of the STACK, that is, I. Push all the neighbours of I onto the stack thatare in the ready state. The STACK now becomes

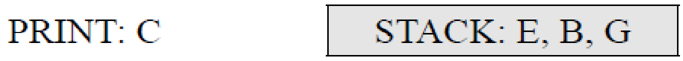
PRINT: I STACK: E, F

(d) Pop and print the top element of the STACK, that is, F. Push all the neighbours of F onto the stack that arein the ready state. (Note F has two neighbours, C and H. But only C will be added, as H is not in the ready state.) The STACK now becomes



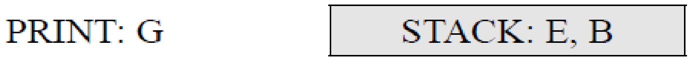
PRINT: F STACK: E, C

(e) Pop and print the top element of the STACK, that is, C. Push all the neighbours of C onto the stack that arein the ready state. The STACK now becomes



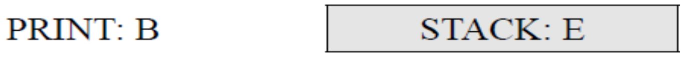
PRINT: C STACK: E, B, G

(f) Pop and print the top element of the STACK, that is, G. Push all the neighbours of G onto the stack that arein the ready state. Since there are no neighbours of G that are in the ready state, no push operation is performed. The STACK now becomes



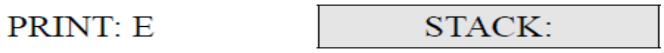
PRINT: G STACK: E, B

(g) Pop and print the top element of the STACK, that is, B. Push all the neighbours of B onto the stack that arein the ready state. Since there are no neighbours of B that are in the ready state, no push operation is performed. The STACK now becomes



PRINT: B STACK: E

(h) Pop and print the top element of the STACK, that is, E. Push all the neighbours of E onto the stack that arein the ready state. Since there are no neighbours of E that are in the ready state, no push operation is performed. The STACK now becomes empty.



PRINT: E STACK:

Since the STACK is now empty, the depth-first search of G starting at node H is complete and the nodeswhich were printed are: H, I, F, C, G, B, E.

These are the nodes which are reachable

from the node H.The time complexity can

be given as (O(|V| + |E|)).

M.Sc. (Computer Science)

SEMESTER-2

COURSE: DATA STRUCTURE AND ALGORITHMS

UNIT 7: SEARCHING

7.1 LINEAR SEARCH:

Definition:

- It starts at the beginning of the list and checks every element of the list.
- i.e. It sequentially checks each element of the list until a match is found or the whole list has been searched.So it is also called sequential search.

Example:

- Let the elements are: 10,6,3,8,9,12,14
- The search element is : 12
- Now it compare 12 with each and every element.
- The 12 is available in 6th place.
- So the searching process is success and element is found

Algorithm:

```
Step 1: Read
  elements in array
  Step 2: Read the
  element to search
Step 3: Compare the element to sear and each element in array
  sequentiallyStep 4: If match is found then the search success
Step 5: If match is not found upto the end then the search un success
```

Program:

```
#include <stdio.h>int main()
{
int a[100],n,i,s;
printf("Enter Number of Elements in
  Array:\n");scanf("%d", &n);
printf("Enter
  numbers:\n")
  ;for(i = 0; i <
  n; i++)
  scanf("%d",
  &a[i]);
printf("Enter a number to search in
  Array:\n");scanf("%d", &s);
for(i = 0; i < n; i++)
{
if(s==a[i])
{
printf("Number found\n”);
  break;
}
}
if(i== n)
printf("Number
  not found\n”);
  return 0;
}
```

7.2 BINARY SEARCH:

Definition:

- Binary search is the most popular Search algorithm. It is efficient and also one of the mostcommonly used techniques that are used to solve problems.
- Binary search sorts the records either in ascending or descending order to gain much better performancethan linear search.
- Now suppose we have an ascending order record. At the time of search it takes the middle record/element, ifthe searching element is greater than middle element then the element mush be located in the second part

else it is in the first half. In this way this search algorithm divides the records in the two parts in eachiteration and thus called binary search.

Example:

- Let the elements in ascending order are2 4 6 8 10 12 15
- Let the element to search 12
- For searching it compare first middle element.2 4 6 8 10 12 15
- The middle element is 8 and is not equal to 12. Since 12 is greater than 8 search on right side part of 8. 12 isequal to right side part middle. So element is found.

Algorithm:

Step 1: Read sorted elements in array

Step 2: Read the element to search

Step 3: Compare the element to sear and middle element in array. If match is found the search success.

Step 4: If match is not found check the search element with middle element. If search element is greater thanthe middle element then search on right side of middle element otherwise search on left.

Step 5: This process is repeated for all elements in array. If no match is found upto the end then the search isnot success.

Program:
```
#include <stdio.h>int main()
{
int i, first, last, middle, n, s, a[100]; printf("Enter number of elements:\n");scanf("%d",&n);
printf("Enter elements in ascending order:\n");for (i = 0; i < n; i++)
scanf("%d",&a[i]);
printf("Enter an element to search:\n");scanf("%d", &s);
first = 0; last = n - 1;
middle = (first+last)/2;while (first <= last)
{
if(s==a[middle])
{
printf("Element is found at index: %d",middle);break;
}
else if(s>a[middle])first = middle + 1; else if(s<a[middle])last = middle - 1;
middle = (first + last)/2;
}
if (first > last)
printf("Element is not found");return 0;
}
```

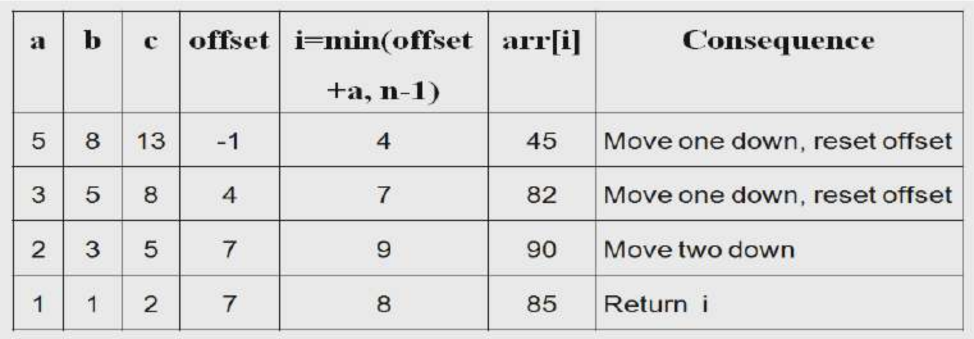
7.3 FIBONACCI SEARCH:

- Fibonacci Search uses Fibonacci numbers to search an element in a sorted array.
- Fibonacci numbers are: 0,1,1,2,3,5,8…
- Fibonacci series generates the subsequent number by adding two previous numbers

Example: Let the elements are given by

10, 22, 35, 40, 45, 50, 80, 82, 85, 90,100 arr[] = {10, 22, 35, 40, 45, 50, 80, 82, 85, 90,100}

- Fibonacci No’s are: 0,1,1,2,3,5,8,13,21,… Target element x is 85. Length of array n = 11
- Find the smallest Fibonacci number greater than or equal to 11 is 13.
- As per our step, a= 5, b = 8, and c = a+b=13.
- Let offset=-1
- First Compute i= min(offset+a, n-1)
- If x is greater than the element, move the three Fibonacci variables one Fibonacci down. Reset offset to index. Else If x is less than the element, move the three Fibonacci variables two Fibonacci down. Else (orwhen b=1) return i – This is the case Element Found
- If element not found return -1
- These steps are shown in the following table



a	b	c	offset	i=min(offset +a, n-1)	arr[i]	Consequence
5	8	13	-1	4	45	Move one down, reset offset
3	5	8	4	7	82	Move one down, reset offset
2	3	5	7	9	90	Move two down
1	1	2	7	8	85	Return i

Algorithm:

Let arr[0..n-1] be the input array and element to be searched be x.

Step 1: Find the smallest Fibonacci Number greater than or equal to n. Let this number be c. Let the twoFibonacci numbers preceding it be a,b.

Step 2: While the array has elements to be inspected such as:

Step-2.1: Compute i= min(offset+a, n-1)

Step-2.2: If x is greater than the element, move the three Fibonacci variables one Fibonacci down. Reset offsetto index.

Step-2.3: Else If x is less than the element, move the three Fibonacci variables two Fibonacci downStep-2.4: Else (or when b=1) return i – This is the case Element Found

Step-3: If element not found return -1

Program: #include <stdio.h>int min(int x, int y)

```
{
return (x<=y)? x : y;
}
int search(int arr[], int x, int n)
{
int a = 0;int b = 1;
int c = a + b;while (c < n)
{
a = b;b = c;
c  = a + b;
}
int offset = -1;while (c > 1)
{
int i = min(offset+a, n-1);


if (x>arr[i])
{
c = b;b = a;
a = c - b; offset = i;
}
else if (x<arr[i])
{
```

```
c  = a;
b = b - a;
a = c - b;
}
else return i;
}
return -1;
}
int main()
{
int arr[] = {10, 22, 35, 40, 45, 50, 80, 82, 85, 90, 100};
int n = sizeof(arr)/sizeof(arr[0]);int x,s;
printf("Enter an element to search:\n");scanf("%d",&x);
s=search(arr, x, n);if(s==-1)
printf("Element is not found");else
printf("Element is Found at index: %d", s);return 0;
}
```

7.4 SORTING

➢ Sorting is a process of placing a list of elements from the collection of data in some order.

➢ It is nothing but storage of data in sorted order. Sorting can be done in ascending and descending order. Itarranges the data in a sequence which makes searching easier

7.5 INSERTION SORT:

❖ In this sorting technique first elements are stored in an array.

❖ The process of sorting starts with second element.

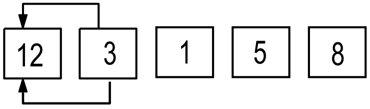
❖ First the second element is picked and is placed in specified order Next third element is picked and is placedin specified order. Similarly the fourth, fifth, …nth element .is placed in specified order.

❖ Finally we get the sorting elements.

Example:

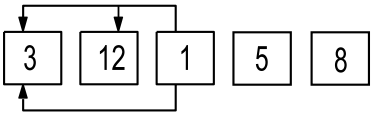
 Let us consider the elements: 12, 3, 1, 5, 8

7.6 Checking second element of array with element before it and inserting it in proper position. In this case 3is inserted in position of 12



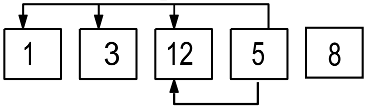
7.7 Checking third element of array with elements before it and inserting it in proper position. In this case 1 is

inserted in position of 3



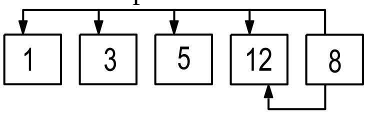
7.8 Checking fourth element of array with elements before it and inserting it in proper position. In this case 5

is inserted in position of 12



7.9 Checking fifth element of array with elements before it and inserting it in proper position. In this case 8 is

inserted in position of 12



7.10 Sorted array in ascending order

1	3	5	8	12

Algorithm:

Step 1: Check second element of array with element before it and insert it in proper position. Step 2: Checking third element of array with element before it and inserting it in proper position.Step 3: Repeat this till all elements are checked.

Step 4: Stop

Program:

```
#include<stdio.h>int main()
{
int n,a[30],key,i,j;
printf("Enter total
  elements:\n");
  scanf("%d",&n);
printf("Enter
  elements:\n"
  );
  for(i=0;i<n;i
  ++)
  scanf("%d",
  &a[i]);
  for(i=1;i<n;i
  ++)
{
j=i;
while(j>0 && a[j]<a[j-1])
{
temp=a[j]
  ;
  a[j]=a[j-
  1]; a[j-
  1]=temp
  ;
j--;
}
}
printf("After
  sorting
  is:\n");
  for(i=0;i<n;i
  ++)
print
  f("
  %d
  ",a
  [i])
  ;
  ret
  urn
  0;
}
```

7.6 SELECTION SORT:

- Selection sort is an algorithm that selects the smallest element from an unsorted list in each iteration andplaces that element at the beginning of the unsorted list.

Example:

- The following figure shows the first pass of a selection sort.

First pass

54	**26**	93	17	77	31		10	5	assign 54 min
26	54	**93**	17	77	31	44	10	5	assign 26 min
26	54	93	**17**	77	31	44	10	5	assign 26 min
17	54	93	26	**77**	31	44	10	5	assign 17 min
17	54	93	26	77	**31**	44	10	5	assign 17 min
17	54	93	26	77	31	**44**	10	5	assign 17 min
17	54	93	26	77	31	44	**10**	5	assign 17 min
10	54	93	26	77	31	44	17	**5**	assign 10 min
5	54	93	26	77	31	44	17	10	

Exchange 10 and 5after first pass

- In first pass the first element is compared with all remaining elements and exchange element if first one isgreater than second so that the smallest value is in first place. Leave this element.
- In second pass compare second element to all elements and put the next smallest value, in second place.Leave this element. This process is repeated till all the elements are placed.
- Now we get the sorted elements.

Algorithm:

Step 1 − Set min to the first location.

Step 2 − Search the minimum element in the array.

Step 3 – swap the first location with the minimum value in the array.Step 4 – assign the second element as min.

Step 5 − Repeat the process until we get a **sorted** array.

Program: #include<stdio.h>int main()

```
{
int
  n,i,j,temp,a[20],m
  in; printf("Enter
  total
  elements:\n");
  scanf("%d",&n);
printf("Enter
  elements:\n")
  ;
  for(i=0;i<n;i
  ++)
  scanf("%d",
  &a[i]);
  for(i=0;i<n;i
  ++)
{
min=i;
  for(j=i+1
  ;j<n;j++)
{
if(a[j] <
  a[min])
  min=j;
}
temp=
  a[i];
  a[i]=
  a[mi
  n];
  a[mi
  n]=t
  emp;
}
printf("After
  sorting
  is:\n");
  for(i=0;i<n;i
  ++)
print
  f("
  %d
  ",a
  [i])
  ;
  ret
  urn
  0;
}
```

7.7 EXCHANGE SORT:

- The exchange sort is almost similar as the bubble sort. The exchange sort compares each element of an arrayand swap those elements that are not in their proper position,

just like a bubble sort does. The only difference between the two sorting algorithms is the manner in whichthey compare the elements..

7.8 BUBBLE SORT:

- Bubble Sort is based on the idea of repeatedly comparing pairs of adjacent elements and then swapping theirpositions if they exist in the wrong order.

Example:

- The following figure shows the first pass of a bubble sort. In first pass the first element is compared withsecond and exchange element if first one is greater than second.
- Similarly second element is compared with third and exchange element if second one is greater than third.
- Repeat this so that at the end of first pass the largest value is in last place. Leave this element.

First pass

54	**26**	93	17	77	31	44	55	20
26	**54**	**93**	17	77	31	44	55	20

26	54	**93**	**17**	77	31	44	55	20
26	54	17	**93**	**77**	31	44	55	20
26	54	17	77	31	44	55	20	93

26	54	17	77	**93**	**31**	44	55	20

26	54	17	77	31	**93**	**44**	55	20
26	54	17	77	31	44	**93**	**55**	20

26	54	17	77	31	44	55	**93**	**20**

Algorithm:

Step 1: The first element is compared with second and exchange element if first one is greater than second

Step 2: Similarly second element is compared with third and exchange element if second one is greater than thirdStep 3: Repeat this so that at the end the largest value is in last place

Step 4: Likewise sorting is repeated for all elements.

Program:

```
#include<stdio.h>int main()
{
int n,temp,i,j,a[20];
printf("Enter total numbers of
  elements:\n");scanf("%d",&n);
printf("Enter
  elements:\n");
  for(i=0;i<n;i++)
  scanf("%d",&a[i]
  );
  for(i=0;i<n;i++)
{
for(j=0;j<n-1;j++)
{
if(a[j]>a[j+1])
{
temp=a[j];
  a[j]=a[j+1];
  a[j+1]=temp;
}
}
}
printf("After sorting elements
  are:\n");for(i=0;i<n;i++)
printf("
  %d",a[i
  ]);
  return
  0;
}
```

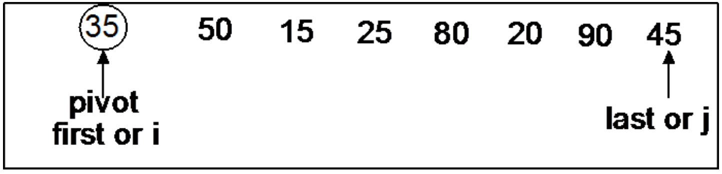
7.9 QUICK SORT:

- Quick Sort is also one of the exchange sort.
- In a quick sort we take pivot element, then we place all the smaller elements are on one side of pivot, andgreater elements are on other side of pivot.
- After partitioning we have pivot in the final position. After repeatedly partitioning, we get the sortedelements.

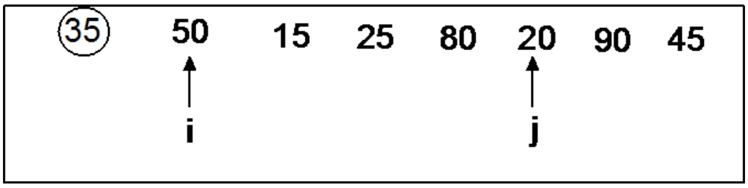
Example:

- Let us consider the elements: 35,50,15,25,80,20,90,45

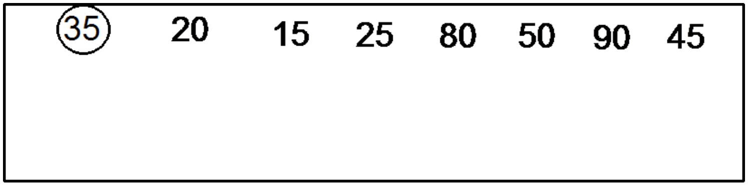
- Let us consider the first element 35 as pivot or i. The last element 45 as j



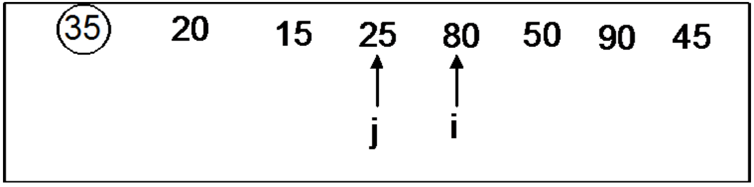
- 50 which is greater than pivot taken as i and the 20 smaller than pivot taken as j



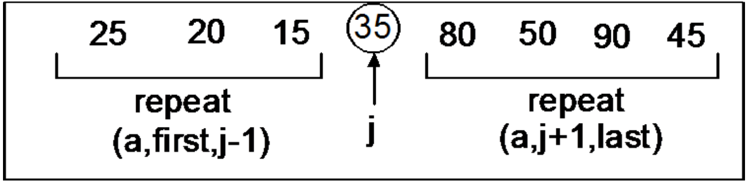
- Now i is less than j so swap the elements in i and j.



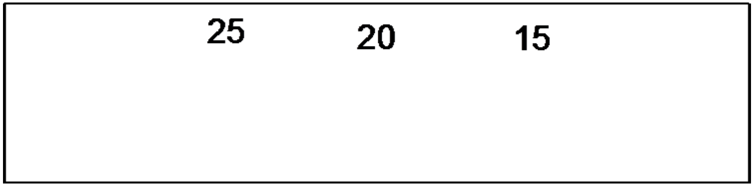
- Find greater than 35 (80)is i and less than 35(25) is j



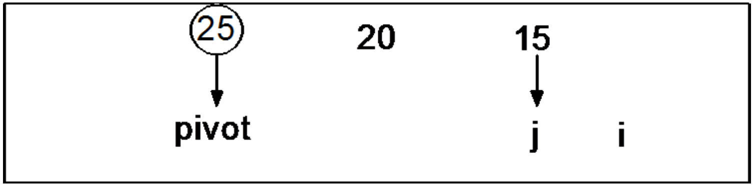
- Now i is not less than j. swap 35 and j so 35 becomes at j place.



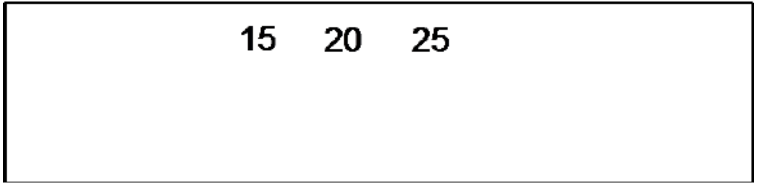
- Now 35 is in correct position.
- On left side and right side of 35 repeat the process. Consider on left side of 35



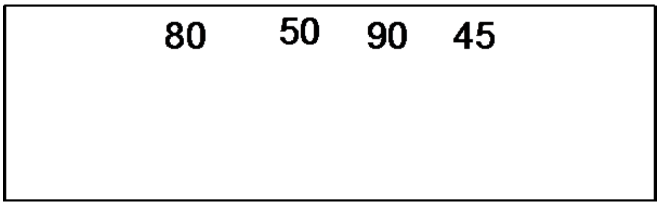
- Let 25 as pivot. The lesser of 25 that is 15 as j and there is no greater. So bring i after j i > j so swap pivotand j



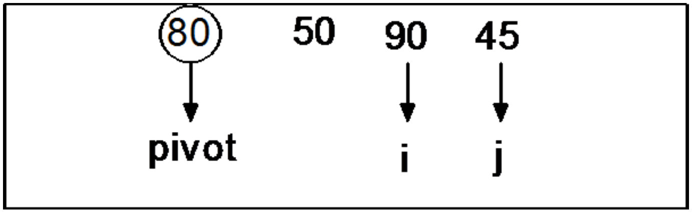
- After swapping



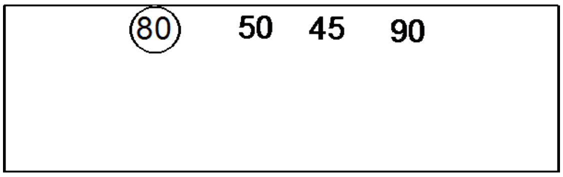
- Now the left part is sorted. Consider right part



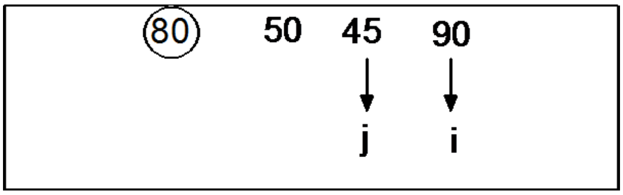
- Here 80 as pivot, Greater than to 80 is i and less than to 80 is j



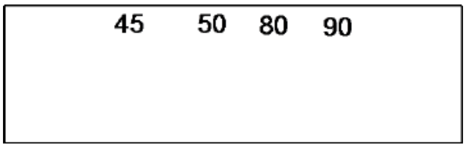
- Here i is less than j so swap i and j elements.



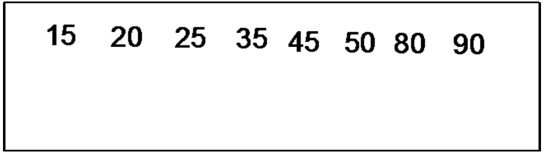
- first find greater to 80 is i and lesser to 80 is j. i > j so swap 80 and j.



- After swapping. The sorting elements are given by



- Now join all left part j and right part j to get the sorted elements



Algorithm:

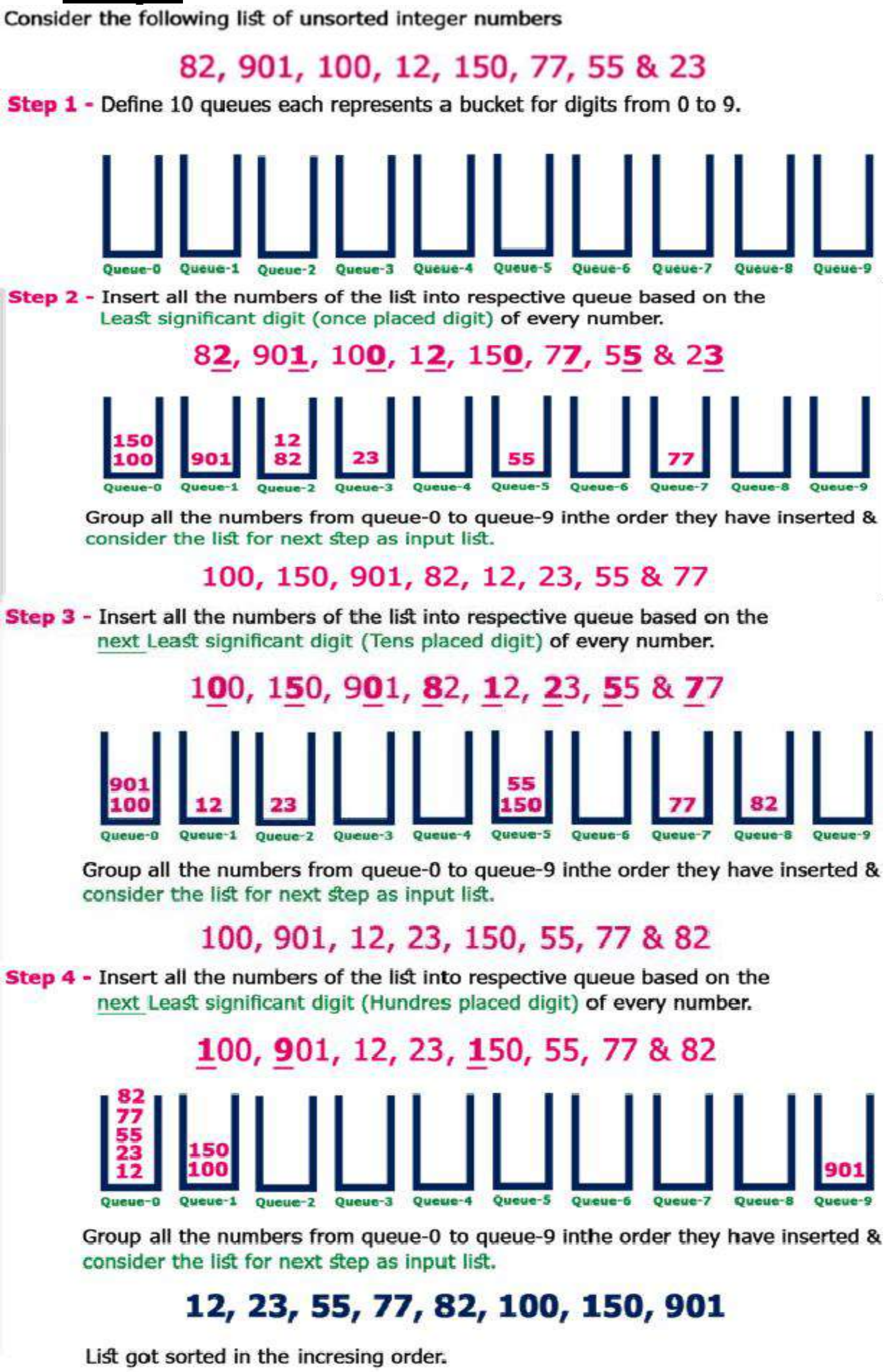
Step 1: Let the first element taken as pivot
Step 2: Find lesser of pivot say i and greater of pivot say j.
Step 3: If i is less than j then i and j elements are swapped. Repeat step 2
Step 4: Repeat
step 3 until i >
jNow swap j
and pivot
Step 5: Now the pivot element is final position.
Repeat the above procedure for left and right side of pivot elements until all elements are sorted
Step 6: Stop

Program: #include<stdio.>

```
void quicksort(int a[25],int first,int last)
{
int i, j,
  pivot,
  temp;
  if(firs
  t<last
  )
{
pivot=
  first;
  i=fir
  st;
  j=las
  t;
  whil
  e(i<j
  )
{
while(a[i]<a[pivot]&&i<=last)i++;
while(a[j]>a[pivot])j--;
  if(i<j)
                    {
temp=a[i];a[i]=a[j];
  a[j]=temp;
}
}
temp=a[pivot];
  a[pivot]=a[j];
  a[j]=temp;
  quicksort(a,first
  ,j-1);
  quicksort(a,j+1,
  last);
}
}
int main()
{
int i, n, a[25];
printf("Enter total a of
  elements:\n ");
  scanf("%d",&n);
printf("Enter
  elements:\n");
  for(i=0;i<n;i+
  +)
  scanf("%d",&
  a[i]);
quicksort(a,0,n-1);
printf("The Sorted
  elements are:\n ");
  for(i=0;i<n;i++)
printf(
  "
  %d",
  a[i]);
  retur
  n 0;
}
```

(iv) Distribution Sort or Radix Sort:

- Radix sort is one of the sorting algorithms used to sort a list of integer numbers in ascending or descendingorder.
- In radix sort algorithm, a list of integer numbers will be sorted based on the digits of individual numbers.Sorting is performed from least significant digit to the most significant digit
- Radix sort algorithm requires the number of passes which are equal to the number of digits present in thelargest number among the list of numbers.
- For example, if the largest number is a 3 digit number then that list is sorted with 3 passes.
- **Example:**



Consider the following list of unsorted integer numbers

82, 901, 100, 12, 150, 77, 55 & 23

Step 1 - Define 10 queues each represents a bucket for digits from 0 to 9.

Step 2 - Insert all the numbers of the list into respective queue based on the Least significant digit (once placed digit) of every number.

82, 901, 100, 12, 150, 77, 55 & 23

Group all the numbers from queue-0 to queue-9 inthe order they have inserted & consider the list for next step as input list.

100, 150, 901, 82, 12, 23, 55 & 77

Step 3 - Insert all the numbers of the list into respective queue based on the next Least significant digit (Tens placed digit) of every number.

100, 150, 901, 82, 12, 23, 55 & 77

Group all the numbers from queue-0 to queue-9 inthe order they have inserted & consider the list for next step as input list.

100, 901, 12, 23, 150, 55, 77 & 82

Step 4 - Insert all the numbers of the list into respective queue based on the next Least significant digit (Hundres placed digit) of every number.

100, 901, 12, 23, 150, 55, 77 & 82

Group all the numbers from queue-0 to queue-9 inthe order they have inserted & consider the list for next step as input list.

12, 23, 55, 77, 82, 100, 150, 901

List got sorted in the incresing order.

Algorithm

Step 1 - Define 10 queues each representing a bucket for each digit from 0 to 9.

Step 2 - Consider the least significant digit of each number in the list which is to be sorted.Step 3 - Insert each number into their respective queue based on the least significant digit.

Step 4 - Group all the numbers from queue 0 to queue 9 in the order they have inserted into their respectivequeues.

Step 5 - Repeat from step 3 based on the next least significant digit.

Step 6 - Repeat from step 2 until all the numbers are grouped based on the most significant digit.

Program:

```
#include<stdio.h>
int getMax(int arr[], int n)
{
int
  max
  =
  arr[0]
  ;int i;
for (i = 1; i < n; i++)
{
if (arr[i] >
  max)max =
  arr[i];
}
return max;
}
void countSort(int arr[], int n, int exp)
{
int output[n]; //
  output arrayint i,
  count[10] = { 0
  };
// Store count of occurrences
  in count[]for (i = 0; i < n;
  i++)
count[(arr[i] / exp)
  % 10]++;for (i =
  1; i < 10; i++)
  count[i] +=
  count[i - 1];
// Build the
  output array
  for (i = n - 1;
  i >= 0; i--)
{
output[count[(arr[i] / exp) % 10] - 1] =
  arr[i];count[(arr[i] / exp) % 10]--;
}
for (i = 0; i <
  n; i++)
  arr[i] =
  output[i];
}
// The main function to that sorts arr[] of size n
  using Radix Sortvoid radixsort(int a[], int n)
{
```

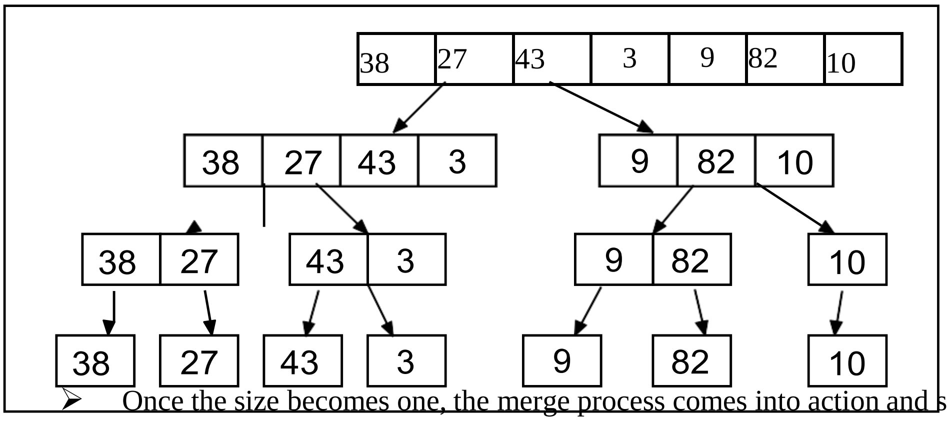
```
int max =
  getMax(a,
  n);int i;
for (i = 1; max / i >
  0; i *= 10)
  countSort(a, n, i);
}
int main()
{
int a[] = { 170, 45, 75, 90, 802, 24, 2, 66 };
int i;
int n = sizeof(a) /
  sizeof(a[0]);
  radixsort(a, n);
for (i = 0;
  i < n;
  i++)
  printf("
  %d ",
  a[i]);
  return 0;                    }
```

7.11 Merging or Merge Sort:

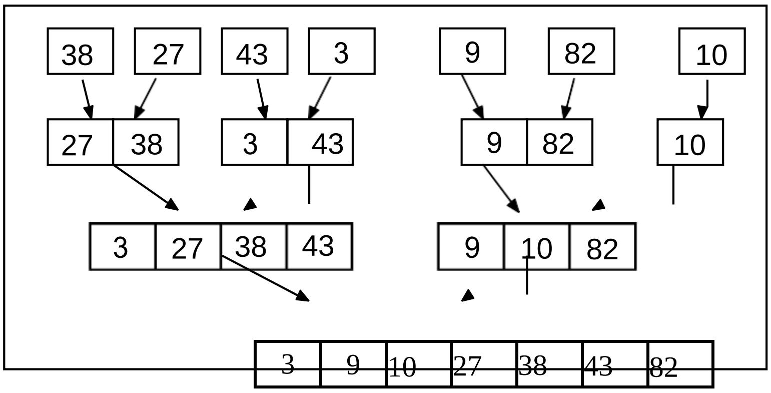
- ➢ It divides input array into two halves, calls itself for the two halves and then sorted and merged that twohalves.

Example:

- ➢ For example consider the array of elements: 38, 27, 43, 3, 9, 82, 10
- ➢ Now the array is recursively divided into two halves till the size becomes one which is shown in thefollowing figure.



- ➢ Once the size becomes one, the merge process comes into action and starts merging with sorted array till thecomplete array is merged



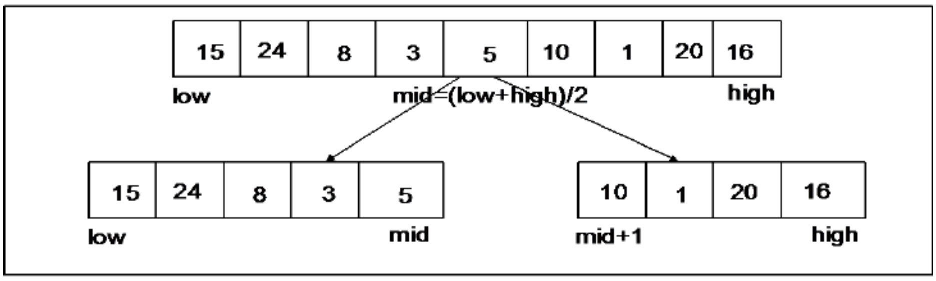
Algorithm:

Step 1 − If it is only one element in the list then it is already sorted.

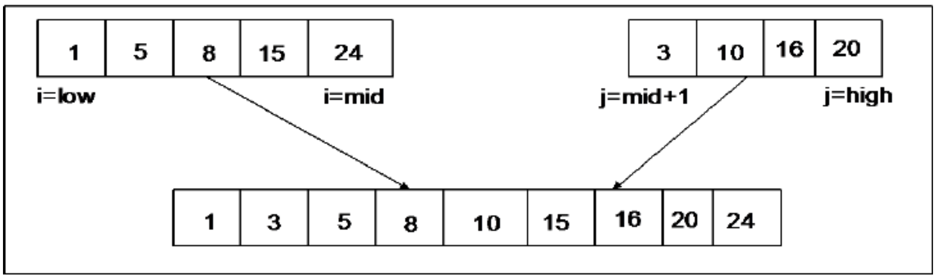
Step 2 − Divide the list recursively into two halves till the size becomes one.

Step 3 − Once the size becomes 1, the merge process comes into action and starts merging with sorted array till thecomplete array is merged

mergesort[splitting]



merge[sorting and merging]



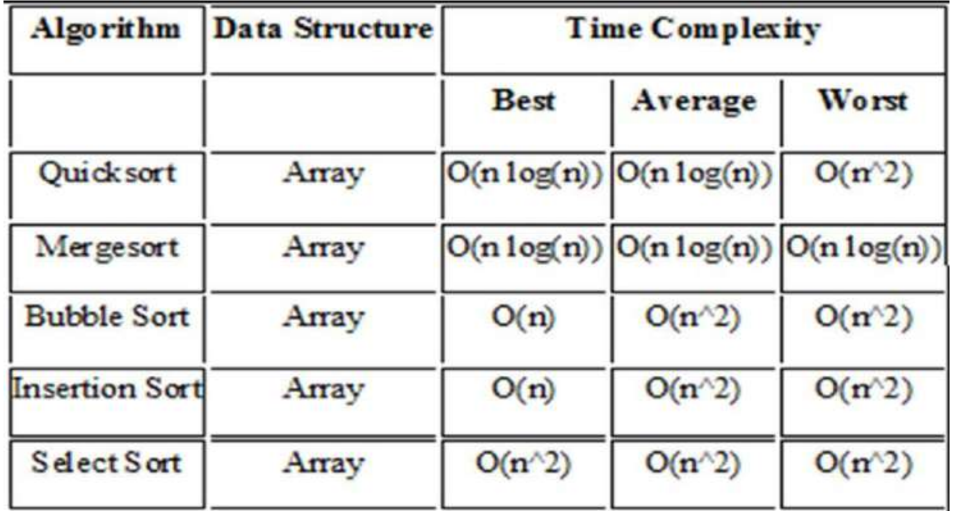
Program:

```
#include<stdio.h>
int n,a[30],i,j,k,temp[30];
void merge(int low,int mid,int high)
{
i=low; j=mid+1;k=low;
while((i<=mid) && (j<=high))
{
if(a[i]>=a[j])
  temp[k++]=a[j++
  ];else
  temp[k++]=a[i++
  ];
}
while(i<=mid)
  temp[k++]=a[i++];
  while(j<=high)
  temp[k++]=a[j++];
  for(i=low;i<=high;i++)
  a[i]=temp[i];
}
void mergesort(int low,int high)
{
int
  mid;
  if(lo
  w!=h
  igh)
{
mid=((low+high)/2);
  mergesort(low,mid);
  mergesort(mid+1,hig
  h);
  merge(low,mid,high
  );
}
}
int main()
{
printf("Enter total
  elements:\n");
  scanf("%d",&n);
printf("Enter
  elements:\n");
  for(i=0;i<n;i++)
  scanf("%d",&a[i]);
  mergesort(0,n-1);
  printf("After
  sorting is:\n");
  for(i=0;i<n;i++)
printf("
  %d",a[i]);
  return 0;
}
```

Time Complexity:

- **O(n^2)** means that for every insert, it takes n*n operations. i.e. 1 operation for 1 item, 4 operations for 2items, 9 operations for 3 items.

Comparison of Sorting Algorithms



Algorithm	Data Structure	Time Complexity		
		Best	Average	Worst
Quicksort	Array	O(n log(n))	O(n log(n))	O(n^2)
Mergesort	Array	O(n log(n))	O(n log(n))	O(n log(n))
Bubble Sort	Array	O(n)	O(n^2)	O(n^2)
Insertion Sort	Array	O(n)	O(n^2)	O(n^2)
Select Sort	Array	O(n^2)	O(n^2)	O(n^2)

Complexity of Radix Sort:

- Radix sort is a non-comparative algorithm, it has advantages over comparative sorting algorithms.
- For the radix sort that uses counting sort as an intermediate stable sort, the time complexity is O(d(n+k)).
- Here, d is the number cycle and O(n+k) is the time complexity of counting sort
- Thus, radix sort has linear time
- complexity which is better than
- O(nlog n) of comparative sorting algorithms.
- If we take very large digit numbers or the number of other bases like 32-bit and
- 64-bit numbers then it can perform in linear time however the intermediate sort takes large space.
- This makes radix sort space inefficient.
- This is the reason why this sort is not used in software libraries.

Best case, Worst case and Average Case of Radix sort -

- Radix sort complexity is O(kn) for n keys which are integers of word size k.
- For all there cases time i.e best , worst and average time complexity is O(kn)

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University
(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-II
Course: INTRODUCTION TO CYBER SECURITY(MSCS-2-03T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

M.Sc. (Computer Science) Programme Coordinator:
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:
Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Business Management & Commerce:
Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Social Sciences and Liberal Arts:
Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab

PROGRAMME COORDINATOR

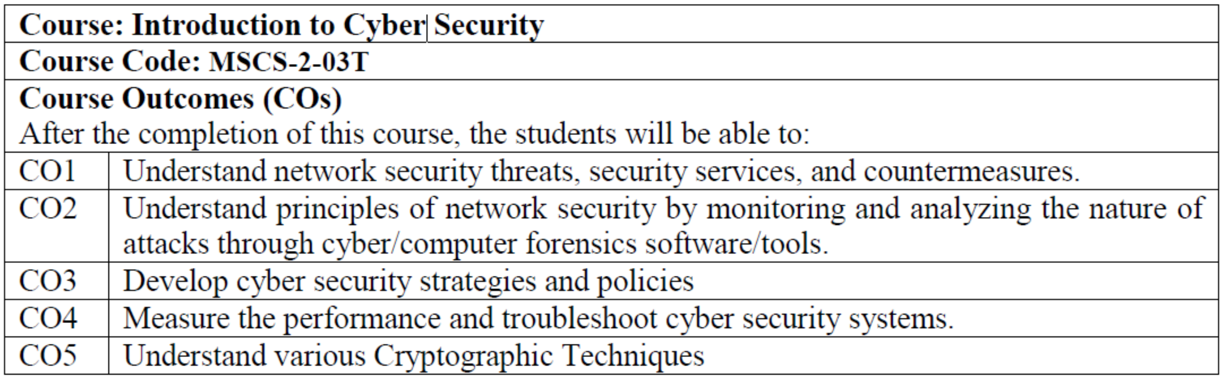
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

COURSE COORDINATOR AND EDITOR:

Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE OUTCOMES (COs)

After Completion of the course, learners will be able to



Course: Introduction to Cyber Security	
Course Code: MSCS-2-03T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Understand network security threats, security services, and countermeasures.
CO2	Understand principles of network security by monitoring and analyzing the nature of attacks through cyber/computer forensics software/tools.
CO3	Develop cyber security strategies and policies
CO4	Measure the performance and troubleshoot cyber security systems.
CO5	Understand various Cryptographic Techniques

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY PATIALA
(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

M.Sc. (Computer Science)
Semester-2
MSCS-2-03T: Introduction to Cyber Security

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

INSTRUCTIONS FOR THE PAPER SETTER/EXAMINER

1. The syllabus prescribed should be strictly adhered to.
2. The question paper will consist of three sections: A, B, and C. Sections A and B will have four questions from the respective sections of the syllabus and will carry 10 marks each. The candidates will attempt two questions from each section.
3. Section C will have fifteen short answer questions covering the entire syllabus. Each question will carry 3 marks. Candidates will attempt any ten questions from this section.
4. The examiner shall give a clear instruction to the candidates to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.
5. The duration of each paper will be three hours.

INSTRUCTIONS FOR THE CANDIDATES

Candidates are required to attempt any two questions each from the sections A and B of the question paper and any ten short q questions from Section C. They have to attempt questions only at one place and only once. Second or subsequent attempts, unless the earlier ones have been crossed out, shall not be evaluated.

SECTION-A

Unit I: Cyber Attacks: Introduction, Types. Assets: Identification, Accountability. Vulnerability and Threats, Risk Management, Qualitative Risk Assessment, Information Security Framework: Introduction, Policies, Standards, Baselines, Guidelines and Procedures.

Unit II: Security: Basics, User Access Controls, Authentication, Access Control: Framework, Techniques and Technologies, Training and Awareness and Its types, Technical Security Controls: Preventive, Detective, Corrective. Protection form malicious attacks.

Unit III: Networks and Communication: Data Communication, Characteristics and components, Data flow. Computer Network, Categories, Protocol, External Services, Cloud Computing: Introduction, Models, Benefits, Challenges, Private, Public Clouds.

Unit IV: Software Engineering Life Cycle: Stages, Models: Waterfall, Iterative, Spiral, V Model, Big Bang, Agile, RAD, Prototype.

SECTION-B

Unit V: Authentication: Authentication Vs Authorization, Methods and Protocols: Kerberos, SSL, Protocol, Password Authentication, Challenge-Handshake Authentication (CHAP), MSCHAP, Extensible Authentication, Remote Authentication.

Unit VI: Service Set Identification (SSID), Encryption Methods: Wire Equivalent Privacy, WPA, WPA2, MAC Filtering, Wireless Routers, Creating Wireless Network, WLAN.

Unit VII: Investigation Techniques and Cyber Forensics: Types of Investigation, Evidence and Analysis, Steps for Forensics Investigation, Forensics Tools, Investigation, Common Types of Email Abuse, Tracking Location of Email Sender, Scam or Hoax Emails and Websites, Fake Social Media Profile.

Unit VIII: Cryptography: Objectives, Type, OS Encryption, Public key Cryptography.

Reference Books:

- Mayank Bhushan, Rajkumar Singh Rathore, Aatif Jamshed, “Fundamentals of CyberSecurity”, BPB Publications.
- Nina Godbole, Sunit Belapure, “Cyber Security”, Wiley.
- Sanil Nadkarni, Fundamentals of Information Security”, pbp.
- Mike Chapple, James Michael Stewart, Darril Gibson, “CISSP Certified InformationSystems Security Professional Official Study Guide” 9th Edition, SYBEX, A Wiley Brand.
- William Chuck Eastton, “Computer Security Fundamentals”, 4th Edition, Pearson

UNIT I: CYBER ATTACKS, TYPES OF ATTACKS MOTIVATION

STRUCTURE

1.0 LEARNING OBJECTIVES

This unit purports at making you understand:

- What constitutes a cyber-attack,
- Types of cyber-attacks, and
- What motivates attacker(s) to do carry out attack(s).

1.1 INTRODUCTION

Everyone among us has one time or another has come across some form of attack. It could be physical or emotional or of some other kind. The intent is to cause some sort of harm – though sometimes it turn into a blessing in disguise. However, cyber attacks always aim at causing harm. They can be varied in their nature of approach and type of harm they inflict, depending on the motive, but the purpose is certainly malicious.

All of you must have encountered a situation when some unwanted changes, like installing some software or change your search engine, are made to your system or seen unwanted advertisements popping up while surfing Internet. These are examples of cyber attacks.

These can range from being minor nuisance, like occasional popups, to creating havoc, like formatting hard disk.

1.2 CYBER ATTACK

Farhat et al[1] on 'What is a cyber attack' state as below:

A cyber attack is an attack initiated from a computer against a website, computer system or individual computer (collectively, a computer) that compromises the confidentiality, integrity or availability of the computer or information stored on it.

According to Anonymous[2], "Cyber-attack is any type of offensive maneuver employed by individuals or whole organizations that targets computer information systems, infrastructures, computer networks, and/or personal computer devices by various means of malicious acts usually originating from an anonymous source that either steals, alters, or destroys a specified target by hacking into a susceptible system. These can be labeled as either a Cyber campaign, cyberwarfare or cyberterrorism in different context. Cyber-attacks can range from installing spyware on a PC to attempts to destroy the infrastructure of entire nations."

n a nutshell, use of a device/system against another system/device with a malicious intent constitutes a cyber attack.

1.3 TYPES OF CYBER ATTACK OR THREATS

Anonymous[3] gives a comprehensive list of cyber attacks/threats which is reproduced below:

1. Backdoors – Backdoors[4] is bypassing normal authentication. Backdoor is a type of cyber threat in which the attacker uses a back door to install a keylogging software, thereby allowing an illegal access to your system. This threat can turn out to be potentially serious as it allows for modification of the files, stealing information, installing unwanted software or even taking control of the entire computer.

   Default passwords can function as backdoors if they are not changed by the user. Some debugging features can also act as backdoors if they are not removed in the release version.

   Many computer worms, such as Sobig and Mydoom, install a backdoor on the affected computer (generally a PC on broadband running Microsoft Windows and Microsoft Outlook). Such backdoors appear to be installed so that spammers can send junk e-mail from the infected machines. Others, such as the Sony/BMG rootkit distributed silently on millions of music CDs through late 2005, are intended as DRM measures—and, in that case, as data gathering agents, since both surreptitious programs they installed routinely contacted central servers.

   A sophisticated attempt to plant a backdoor in the Linux kernel, exposed in November 2003, added a small and subtle code change by subverting the revision control system. In this case, a two-line change appeared to check root access

permissions of a caller to the sys_wait4 function, but because it used assignment = instead of equality checking
==, it actually granted permissions to the system. This difference is easily overlooked, and could even be interpreted as an accidental typographical error, rather than an intentional attack.

In January 2014, a backdoor was discovered in certain Samsung Android products, like the Galaxy devices. The Samsung proprietary Android versions are fitted with a backdoor that provides remote access to the data stored on the device. In particular, the Samsung Android software that is in charge of handling the communications with the modem, using the Samsung IPC protocol, implements a class of requests known as remote file server (RFS) commands, that allows the backdoor operator to perform via modem remote I/O operations on the device hard disk or other storage. As the modem is running Samsung proprietary Android software, it is likely that it offers over-the-air remote control that could then be used to issue the RFS commands and thus to access the file system on the device.

2. Denial-of-Service Attack – A denial-of-service (DoS) attack is attacking the network to bring it down completely with useless traffic by affecting the host device which isconnected to the Internet. DoS attack targets websites or services which are hosted on the servers. This type of attack can aim bank servers and credit card payment gateways.
3. Direct-access Attack – A direct-access attack simply means gaining physical access to the computer or its part and performing various functions or installing various types of devices to compromise security. The attacker can install software loaded with worms or download important data, using portable devices.
4. Eavesdropping – As the name suggests, eavesdropping means secretly listening to a conversation between the hosts on a network. There are various programs such as Carnivore and NarusInsight that can be used to eavesdrop.
5. Spoofing – Spoofing is a cyber attack where a person or a program impersonate another by creating false data in order to gain illegal access to a system. Such threats are commonly found in emails where the sender’s address is spoofed.
6. Tampering – Tampering is a web based attack where certain parameters in the URL are changed without the customer’s knowledge; and when the customer keys in that URL, it looks and appears exactly the same. Tampering is basically done by hackers and criminals to steal the identity and obtain illegal access to information.
7. Repudiation Attack – A repudiation attack occurs when the user denies the fact that he or she has performed a certain action or has initiated a transaction. A user can simply deny having knowledge of the transaction or communication and later claim that such transaction or communication never took place.
8. Information Disclosure – Information disclosure breach means that the information which is thought to be secured is released to unscrupulous elements who are not trustworthy.
9. Privilege Escalation Attack – A privilege escalation attack is a type of network

intrusion which allows the user to have an elevated access to the network which was primarily not allowed. The attacker takes the advantage of the programming errors and permits an elevated access to the network.

10. Exploits – An exploit attack is basically a software designed to take advantage of a flaw in the system. The attacker plans to gain easy access to a computer system and gain control, allows privilege escalation or creates a DOS attack.

11. Social Engineering – An attack by a known or a malicious person is known as social engineering. They have knowledge about the programs used and the firewall security and thus it becomes easier to take advantage of trusted people and deceive them to gain passwords or other necessary information for a large social engineering attack.

12. Indirect Attack – Indirect attack means an attack launched from a third party computer as it becomes more difficult to track the origin of theattack.

13. Computer Crime – A crime undertaken with the use of a computer and a network is called as a computer crime.

14. Malware – Malware refers to malicious software that are being designed to damage or perform unwanted actions into the system. Malware is of many types like viruses, worms, Trojan horses, etc., which can cause havoc on a computer's hard drive. They

can either delete some files or a directory or simply gather data without the actual knowledge of the user.

15. Adware – Adware is a software that supports advertisements which renders ads to its author. It has advertisements embedded in the application. So when the program is running, it shows the advertisement. Basically, adware is similar to malware as it uses ads to inflict computers with deadly viruses.

16. Bots – Bots is a software application that runs automated tasks which are simple and repetitive in nature. Bots may or may not be malicious, but they are usually found to initiate a DoS attack or a click fraud while using the internet.

17. Ransomware – Ransomware is a type of cyber security threat which will restrict access to your computer system at first and will ask for a ransom in order for the restriction to be removed. This ransom is to be paid through online payment methods only which the user can be granted an access to their system.

18. Rootkits – A rootkit is a malicious software designed in such a way that hides certain process or programs from normal anti virus scan detection and continues to enjoy a privilege access to your system. It is that software which runs and gets activated each time you boot your system and are difficult to detect and can install various files and processes in the system.

19. Spyware – Spyware, as the name suggests, is a software which typically spies and gathers information from the system through a user's internet connection without the user's knowledge. A spyware software is majorly a hidden component of a freeware program which can be downloaded from the internet.

20. Scareware – Scareware is a type of threat which acts as a genuine system message and guides you to download and purchase useless and potentially dangerous software. Such scareware pop-ups seem to be similar to any system messages, but actually aren't. The main purpose of the scareware is to create anxiety among the users and use that anxiety to coax them to download irrelevant softwares.

21. Trojan Horses – Trojan Horses are a form of threat that are malicious or harmful codes hidden behind genuine programs or data which can allow complete access to the system and can cause damage to the system or data corruption or loss/theft of data. It acts as a backdoor and hence it is not easily detectable.

22. Virus – A computer virus is a self replicating program which, when executed, replicates or even modifies by inserting copies of itself into another computer file and infects the affected areas once the virus succeeds in replicating. This virus can be harmful as it spreads like wildfire and can infect majority of the system in no time.

23. Worm – Just like a virus, worm is a self replicating program which relies on computer network and performs malicious actions and spreads itself onto other computer networks. Worms primarily rely on security failures to access the infected system.

24. Phishing – Phishing is a cyber threat which makes an attempt to gain sensitive information like passwords, usernames and other details for malicious reasons. It is basically an email fraud where the perpetrator sends a legitimate looking email and attempts to gain personal information.

25. Identity Theft – Identity theft is a crime wherein your personal details are stolen and these details are used to commit a fraud. An identity theft is committed when a criminal impersonates individuals and use the information for some financial gain.

26. Intellectual Property Theft – Intellectual Property theft is a theft of copyrighted material where it violates the copyrights and the patents. It is a cybercrime to get hands onto some trade secrets and patented documents and research. It is basically a theft of an idea, plan and the methodology being used.

27. Password Attacks – Password attack is a form of a threat to your system security where attackers usually try ways to gain access to your system password. They either simply guess the password or use an automated program to find the correct password and gain an entry into the system.

28. Bluesnarfing – Bluesnarfing is a threat of information through unauthorized means. The hackers can gain access to the information and data on a Bluetooth enabled phone using the wireless technology of the Bluetooth without alerting the user of the phone.

29. Bluejacking – Bluejacking is simply sending of texts, images or sounds, to another Bluetooth enabled device and is a harmless way of marketing. However, there is a thin line between bluejacking and bluesnarfing and if crossed it results into an act of threat.

30. DDoS – DDoS basically means a Distributed Denial of Service. It is an attempt to make any online service temporarily unavailable by generating overwhelming traffic from multiple sources or suspend services of a host connected to the internet.
31. Keylogger – A keylogger is a spyware that has the capability to spy on the happenings on the computer system. It has the capability to record every stroke on the keyboard, web sites visited and every information available on the system. This recorded log is then sent to a specified receiver.

1.4 MOTIVATION

Depending on the motivation, according to Ray[5], Verisign iDefense Security Intelligence Services classifies cyber-attacks into three categories: hacktivism, cyber crime and cyber-espionage.

Hacktivism is the act of hacking, or breaking into a computer system, for a politically or socially or ideologically motivated purpose. It is basically used as a means to promote an agenda. Hacktivists are responsible for denial-of-service (DoS), distributed denial of service (DDoS), information theft, data breaches, web site defacement, typosquatting(URL hijacking relying on typographical errors in URL spelling) and many other acts of digital sabotage.

Cyber crime, though, in a broad sense, covers any illegal activity that is committed through a digital means, here it refers to an activity with the monetary gain in mind. Such an activity can be a direct one, e.g., fraudulent bank transaction, or an indirect one, e.g., selling stolen information in black market. Frequently used cyber crime tools are ATM and point-of-sale (PoS) skimming, RAM scrapping, code injection, key logging and phishing to extract confidential personal information.

Cyber espionage is unauthorized spying by computer[6]. However, a more comprehensive definition, and the associated tools, is given by Anonymous[7] which is as below:

Cyber spying, or cyber espionage, is the act or practice of obtaining secrets without the permission of the holder of the information (personal, sensitive, proprietary or of classified nature), from individuals, competitors, rivals, groups, governments and enemies for personal, economic, political or military advantage using methods on the Internet, networks or individual computers through the use of cracking techniques and malicious software including Trojan horses and spyware. It may wholly be perpetrated online from computer desks of professionals on bases in far away countries or may involve infiltration at home by computer trained conventional spies and moles or in other cases may be the criminal handiwork of amateur malicious hackers and software programmers.

John Arquilla (a US expert on national security affairs and defense analysis) added to new dimension to motivation behind cyber attacks by coining the term cyber warfare or cyber war. Cyberwarfare has been defined as "actions by a nation-state to penetrate another nation's computers or networks for the purposes of causing damage or disruption," but other definitions also include non-state actors, such as terrorist groups, companies, political or ideological extremist groups, hacktivists, and transnational criminal organizations[8].

The above definition of cyber espionage is very likely to raise some confusion as to whether it does not cover cyber war. It does not, which has been made clear by Anonymous [8] as below:

‘Cyber "war" is simply the act of fighting on an electronic battlefield with digital weapons. To attack an adversary’s capabilities in an effort to disable or destroy their ability to get things done. This may be completely digital in nature (such as communication and information systems) or the electronics that monitor and manage physical infrastructure, like power and water systems. Hostile code like StuxNet is an example of such weapons for cyber warfare.

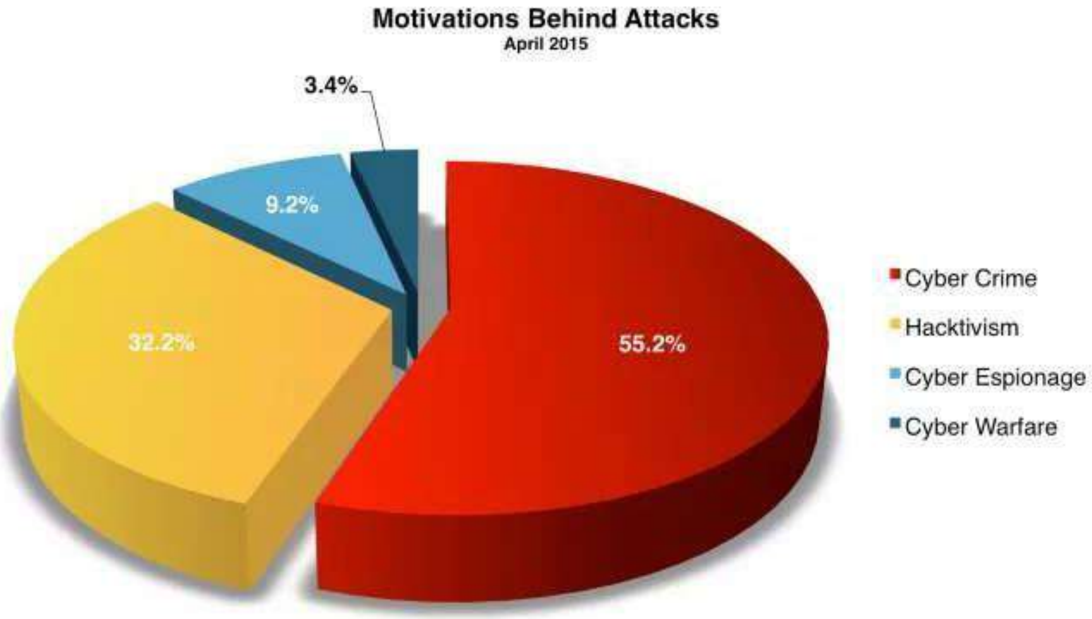
Cyber "espionage" on the other hand is the act of obtaining information that is held in secrecy by the adversary. This in itself is not the end game - this information is then used for some sort of gain or strategic advantage. It must have an intrinsic value to the adversary, or its useless. In many cases, this may be to gain financial / competitive advantage in the business world, or strategic advantage over political communities of conflict.

Now here is where it gets complicated and is the source of much of the confusion. Cyber espionage is routinely used as a precursor to a cyber warfare strike. This allows an adversaryto do reconnaissance in aid of an attack. In the movies, this would be sending in the recon patrol in the military to disable an enemy’s capabilities before a major attack, or sending a spy into the enemy territory to gather intel before the strike. And this happens in the real world too.

Typically though cyber espionage is a covert operation that takes months or years to commit. It usually comes with signs of exfiltration and with the right tools can be tracked back to the source, with some level of certainty. Cyber warfare is different. The attack is usually pretty fast, striking in seconds and causing damage for use with other objectives.’

It must be noted that a perpetrator may belong to more than category of attack. For example, politically motivated cyber attacks may be carried out by members of extremist groups who use cyberspace to spread propaganda, attack websites, and steal money to fund their activities or to plan and coordinate physical-world crime[9].

The figure below shows worldwide motivation statistics, typically for April 2015. It clearly shows that most attacks (> 50%) fall under category ‘cyber crime’ whereas about one third belong to hacktivism. This is obvious from the fact that these two categories consist of mainly individuals and groups and require less resources whereas ‘cyber espionage’ and ‘cyber warfare’ usually require greater resources and, in many cases, government backing.


Motivations Behind Attacks
April 2015
3.4%
9.2%
32.2%
55.2%
🟥 Cyber Crime
🟨 Hacktivism
🟦 Cyber Espionage
🟦 Cyber Warfare

1.5 ASSET, THREAT AND RISK MANAGMENT

1.5.1 INTRODUCTION TO ASSET

Information security core objective is to secure the information asset of the organization. Loss of information can have severe impact over the economic condition of the organization along with the reputation loss in the market. It is a well-known fact that you cannot secure what you do not know exist in your environment. Asset management is all about discovery, ownership, value, acceptable use, protection, disposal of information related assets. Assets can be tangible and intangible. Examples of tangible assets are software and data while server is an example of tangible asset

The task of identifying assets that need to be secure is a less glamorous aspect of information security. But unless we know these assets, their locations and value, how are we going to choose the amount of time, effort or money that we should spend on safeguarding the assets? The major steps required for asset classification and controls are:

- Identification of the assets
- Accountability of assets
- Preparing a schema for information classification
- Implementing the classification schema

Identification of assets

What are the critical assets? Suppose your corporate office was devastated in a major fire. Surviving with this level of adversity will depend on what critical information you previously backed up at a remote location. Another terrifying scene is that a hacker hacked into your network and copied your entire customer database. What impact will this have on your business?

Identifying the critical assets is important for many reasons. You will come to know what is critical and crucial for the business. You will be able to take suitable decisions regarding the level of security that should be provided to safeguard the assets. You will also be able to decide about the level of redundancy that is necessary by keeping an extra copy of the data or an extra server that you should procure and keep as a hot standby.

We should now focus on what is “Information Asset”? Is it hardware, software, program or database? We can broadly classify assets in the following categories:

Information assets

Every piece of information about your organization falls in this category. This information has been collected, classified, organized and stored in various forms.

i. Databases: Information about customer, production, finances and other different areas which are critical to the business. Confidentiality, Integrity and availability depends upon the classification by the data owner. Operational and support procedures: These have been developed over the years and provide detailed instructions on how to perform various activities.

ii. Archived information: Information of previous months or business cycles to maintain because of the law.
iii. Continuity plans, fall-back arrangements: These plans are created to overcome any incident which can impact the business. Absence of these could result into the discontinuity of the business for a shorter or longer period depends upon the severity of the incident.

Software assets
These can be divided into two categories:

i. Application software: Application software implements business rules of the organization. Creation of application software is a time consuming task. Integrity of application software is very important. Any flaw in the application software could impact the business adversely.
ii. System software: An organization would invest in various packaged software programs like operating systems, DBMS, development tools and utilities, software packages, office productivity suites etc.

Most of the software under this category would be available off the shelf, unless the software is obsolete or non-standard.

Physical assets
These are the visible and tangible equipment and could comprise of:

i. Computer equipment: Mainframe computers, servers, desktops and notebook computers.
ii. Communication equipment: Modems, routers, EPABXs and fax machines.
iii. Storage media: Magnetic tapes, disks, CDs and DATs.
iv. Technical equipment: Power supplies, air conditioners.
v. Furniture and fixtures

Services
Services that organization has outsourced to third party.

1.5.2 Accountability of assets

The next step is to create accountability of assets. This can be done easily for the tangible asset. A more difficult task is creating ownership for the information assets. There will be a number of users for these assets. But the prime responsibility for accuracy will lie with the asset owner. Any addition or modification to the information asset will only be done with the consent of the asset owner. For example, any changes to customer information will be done with the knowledge and consent of the marketing head. Information technology staff will probably make the changes, physically. But ownership clearly lies with the business head who has the prime responsibility for the content in the customer database.

Using these criteria, we have to identify the actual owners of each of the information assets. This is also an important step for one more reason. Only an owner of the asset will be able to decide the business value of the asset. Unless the correct business value of the asset is known, we cannot identify the security requirement of the asset.

The next step is identifying owners of the application software. Application software implements the business rules. As such the business process owner should be the owner of application software. But the responsibility of maintaining application software to accurately reflect business rules will be vested with the application developers. As such, the accountability for application software should be with the application development manager.

System software ownership could be with the appropriate persons within the IT team. The owner of these assets will be responsible for maintaining all the system software including protecting the organization against software piracy.

Assets valuation

Another important task is to identify the value of the asset. Asset owner is the right person to verify the value of the asset. But the valuation of the information is a tedious task and depends on many factors which needs to be consider while evaluating them. We need to also consider the fact if in case information is not available how much it will going to impact our business. Also in case this information is leaked in market how it will going to impact the organization reputation in the market.

1.5.3 Preparing a schema for classification

The next important task is to create classification levels. The criteria for the classification of assets could be:

1. Confidentiality: Information comes under this criteria is highly important to the organization and only privileged employees should have access to it. Proper control should be put in place to control the access to this information.
2. Value: What is the asset value? Is it a high value item, costly to replace or a low value item?
3. Time: Is the information time sensitive? Will its confidentiality status change after some time?
4. Access rights: Who will have access to the asset?
5. Destruction: How long the information will be stored? How can it be destroyed, if necessary?

Each asset needs to be evaluated against the above criteria and classified for easy identification. Let us look at each category for classification.

Confidentiality could be defined in terms of:

a. **Confidential**: Where the access is restricted to a specific list of people. These could be company plans, secret manufacturing processes, formulas, etc.
b. **Company only**: Where the access is restricted to internal employees only. These could be customer databases, manufacturing procedures, etc.
c. **Shared**: Where the resources are shared within groups or with people outside of the organization. This could be operational information and contact information like the internal telephone book to be shared with business partners and agents.
d. **Unclassified**: Where the resources are publicly accessible. For example, the company sales brochure and other publicity material.

Classification based on values could be high, medium or low value. Business justifications should be needed to support this classification. Criticality of the assets depends upon the impact it will create on the business. For example, a server who might not be very expensive but it can have the data which is very critical to the organization.

Access rights need to be defined for individuals by the owners. It depends on who is allowed to access the confidential information in the organization. Also who will approve to access those data in the organization?

Destruction of the information is a controlled activity. The information that is not required by the company any longer should be used by the competitor in the same business, that information should be destroyed by the pre-decided schedule and method depends on the confidentiality classification.

Classification schema should lead to an implementable structure. It should be simple to understand and identify.

1.5.4 Implementation of the classification schema

The real test of classification schema is when it is implemented. Information is a fluid resource. It keeps changing its form. The implementation should lead to a uniform way of identifying the information so that a uniform protection could be provided.

Let us take an example. A company's business plan is a confidential document. Let us trace its journey in the corporate world. The plan will be discussed behind closed doors, known to only a few senior members. In the next step the final plan will be prepared and stored on the MD's computer or that of his secretary. A soft copy of this plan would be sent by email to all executives who need to refer to it. The hard disk of every computer where the plan is stored will also have a backup copy on floppy or other media. Each member will no doubt print it and keep a hard copy folder for reference. An extra copy will also be prepared using the copying machine. If the email is not available, the plan would be sent by fax, post or courier.

So the 'confidential' plan is now distributed across the organization, available on the hard disks of computers belonging to each secretary and each senior executive. You get the general idea. If this can happen to confidential information, imagine how easy it is to get hold of other types of information. The information explosion has given rise to proliferation of information in every nook and corner of the organization.

A practical implementation of classification schema thus becomes very important. The classification label should not give an easy way of identification, which could be misused. It should provide the right amount of protection. In the example given above, each and every asset where the confidential information is residing or transiting through will have to be given the same classification level as that of the information itself. It may be desirable to altogether avoid transmission of confidential documents in soft copy format, for example as an attachment to email. Only a restricted number of hard copies should be circulated. If it is necessary to carry the soft copies, everyone should be instructed to encrypt information for transmission and storage, and to memorize their passwords and keep them secret.

These frame works are used as plans or blueprints to design the security of an information security program to mitigate risk and bring down the impact of the risks under the acceptance criteria. Frameworks are often customized as per the requirement of the organizations. Framework assists enterprise to achieve their objectives and deliver values through effective governance and management.

1.6 VULNERABILITY AND THREATS

Information security vulnerabilities are weaknesses that expose an organization to risk. Vulnerability is a weakness in a system that could allow an attacker to compromise the security of the organization.

Vulnerability is a cyber-security term that refers to a flaw in a system that can leave it open to attack. Vulnerability may also refer to any type of weakness in a computer system itself, in a set of procedures, or in anything that leaves information security exposed to a threat.

Threats can exploit the vulnerabilities to impact the performance of the systems. A threat, in the context of information security, refers to anything that has the potential to cause serious harm to a system. Threats can include everything from viruses, Trojans, and back doors to outright attacks from hackers. A threat is something that may or may not happen, but has the potential to cause serious damage. Threats can lead to attacks on computer systems, networks and more.

The lack of access control in an office can be an example of vulnerability but unauthorized person who intentionally or unintentionally want to access the office premises will act as a threat which can exploit the absence of access control in the premises.

Types of threat

- **Physical damage**
    - fire
    - water
    - pollution
- **Natural events**
    - climatic
    - seismic
    - volcanic
- **Loss of essential services**
    - electrical power
    - air conditioning
    - telecommunication
- **Compromise of information**
    - eavesdropping,
    - theft of media
    - retrieval of discarded materials
- **Technical failures**
    - equipment
    - software
    - capacity saturation

- **Compromise of functions**
    - error in use
    - abuse of rights
    - denial of actions

- **Accidental**
    - equipment failure
    - software failure
- **Environmental**
    - natural event
    - loss of power supply

1.7 RISK MANAGEMENT

Risk management is an activity to manage the assessment, mitigation and monitoring of the risk in an organization. Information Security Risk Management is subset of the enterprise risk management. Information Security risk management access the risk which can impact the ‘Confidentiality’, ‘Integrity’ and ‘Availability’ of the organizational information. It also helps to identify the appropriate management actions and defined the priorities for implementing controls to protect those risks.

The risk management process help to create the organizational priorities and help organization to identify risk appetite for them. Top level management is authorized to make decisions about risk acceptance criteria.

Information security decisions should be managed by the top management. Only leadership of the organization should be able to decide the risk acceptance criteria because they are the stakeholders.

This process can be broadly divided into two components:

- Risk assessment
- Risk Mitigation

Risk assessment identifies, quantifies, and prioritizes risks against both criteria for risk acceptance and intents related to the organization. The assessment will result into the proper priorities of security risks and implementation of controls for securing those risks. The assessment result into determining of appropriate management actions and priorities for managing information security risks and for implementation of controls against them. The assessment helps to identify the impact of the risk. It also helps to identify the amount of resources needs to protect the assets. The scope of a risk assessment can be either whole organization, parts of the organization, and individual information system, or even specific system components or services. Performing risk assessment in a company infrastructure includes vulnerability assessment to help quantify risks. This process of assessing risks and helps to quantify them. This will also ensure that constantly evolving changes in security requirements and/or significant changes are assessed. For example, IT will be implementing new products or service each year and new additional risk may be introduced due to vulnerabilities that can be exploited.

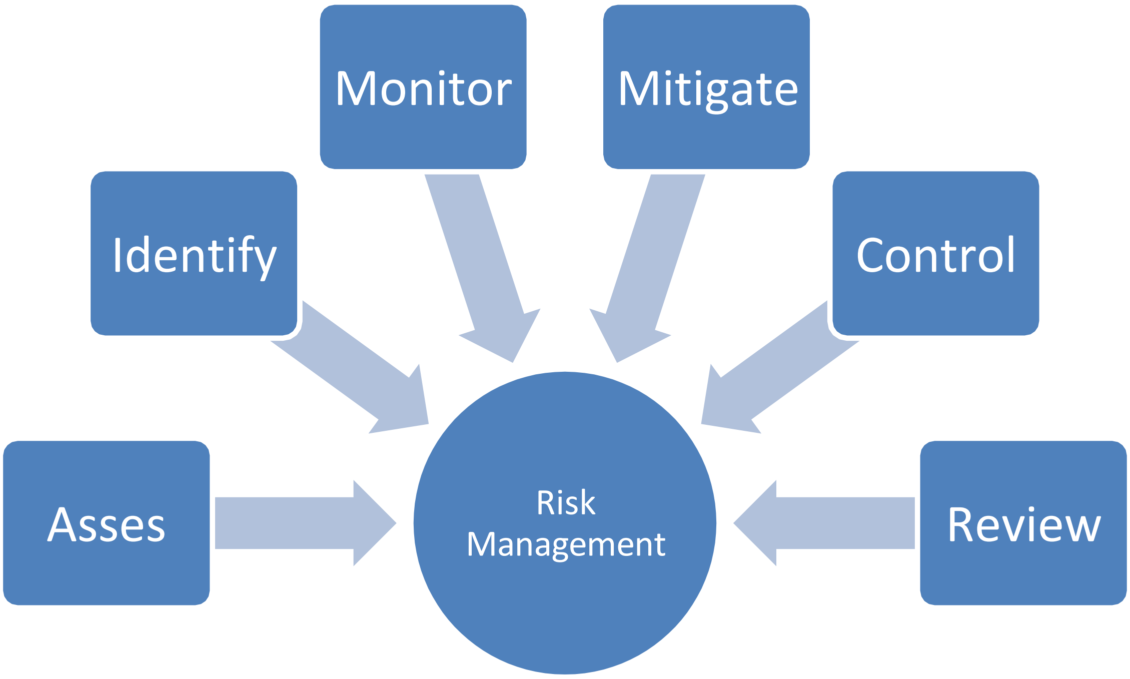


Figure 2: Risk management

Once a risk assessment is finished, risk treatment/risk mitigation is the next step in the process. For each of the risks identified during an assessment there should be a risk mitigation needs to be made. Risk mitigation is a systematic methodology used by senior management to reduce impact of the risk.

Risk mitigation can be completed through any of the following risk mitigation options:

- Risk Assumption: To accept the potential risk and continue operating the IT system or to implement controls to lower the risk to an acceptable level.
- Risk Avoidance: To avoid the risk by eliminating the cause or root cause of the system.
- Risk Limitation: To avoid the risk by eliminating the risk cause and/or consequence (e.g., for certain functions of the system or shut down the system when risks are identified)

Once a risk assessment is finished, **risk treatment/risk mitigation** is the next step in the process. For each of the risks identified during an assessment there should be a risk mitigation needs to be made. Risk mitigation is a systematic methodology used by senior management to reduce impact of the risk. Risk mitigation can be completed through any of the following risk mitigation options:

- Risk Assumption: To accept the potential risk and continue operating the IT system or to implement controls to lower the risk to an acceptable level.
- Risk Avoidance: To avoid the risk by eliminating the cause or root cause of the system.
- Risk Limitation: To avoid the risk by eliminating the risk cause and/or consequence (e.g., for certain functions of the system or shut down the system when risks are identified)
- Risk Planning: To manage risk by developing risk mitigation plan that prioritizes, implements, and maintains controls.
- Risk Transfer: To transfer the risk by using other options to compensate for the loss, such as purchasing insurance.

There are a variety of risk assessment tools and methodologies that can be used, but all are basically divided into **quantitative** and **qualitative** risk assessments.

1.7.1 Quantitative Risk Assessment

Quantitative risk assessments attempt to allocate a monetary value to the assets being measured, a monetary cost to the influence of an adverse event, and percentages to the frequency of threats and the likelihood of events. The monetary values and costs mentioned above are used to determine three elements needed to complete a quantitative risk assessment:

1. **Single Loss Expectancy (SLE):** What is the predictable loss from a single event? Consider physical destruction or theft of assets, loss of data, stopped or delayed processing, and interruption of business processes. Single-loss expectancy (SLE) is the monetary value predictable from the occurrence of a risk on an asset.

   SLE = Asset Value x Impact (percent of asset loss incurred after an event)

2. **Annualized Rate of Occurrence (ARO):** How many times is an event expected to happen in a year?

For example, if insurance data suggests that a serious fire is likely to occur once in 25 years, then the annualized rate of occurrence is 1/25 = 0.04.

3. **Annual Loss Expectancy (ALO):** The annualized loss expectancy (ALE) is the product of the annual rate of occurrence (ARO) and the single loss expectancy (SLE).

ALO = SLE * ARO.

Advantages of Quantitative Risk Assessments

- Allows for a description and communication of consequences of event occurrence in monetary terms.
- It facilitates costs and benefits analysis for the selection of controls for the mitigation.

Disadvantages of Quantitative Risk Assessments

- It is very difficult in some cases to assign a dollar value to assets under the scope of the risk assessment. Especially in case information is under the scope of risk assessment as it is very difficult to identify the exact value of the information.
- Requires extensive time and staff resources.
- Values and costs are only as good and meaningful as the scope and accuracy of the amounts used to calculate them.
- Results of the assessment may be not exact and may be confusing.

1.7.2 Qualitative Risk Assessment

Qualitative risk assessments do not assign a financial value to the assets being measured, or to the impact of an adverse event. They measure the criticality of the assets and impact in range of high, low and medium. This ranking most parts comes under subjective:

- Low – Minor inconvenience tolerated for a short period of time.
- Medium — can result in destruction to the organization’s assets which will require a moderate amount of time, effort, and money to repair.
- High— can result in loss of organization status. It will also result in a legal action or fine.

Advantages of Qualitative Risk Assessments

- Allows for ordering risks according to priority.
- Does not require extensive time and staff resources.
- It can recognize areas of greater risk in a short time and without significant expense.

Disadvantages of Qualitative Risk Assessments

- Results are estimates and subjective.
- Cost-benefit analysis during selection of mitigating controls is subjective.

1.8 SUMMARY

1. Asset management is all about discovery, ownership, value, acceptable use, protection, disposal of information related assets.
2. Information security vulnerabilities are weaknesses that expose an organization to risk. A vulnerability is a weakness in a system that could allow an attacker to compromise the security of the organization.
3. Threats can exploit the vulnerabilities to impact the performance of the systems. A threat, in the context of information security, refers to anything that has the potential to cause serious harm to a system.3

UNIT II: SECURITY CONTROLS

STRUCTURE

2.1 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Know security basics
- Implement physical controls
- Define access control models
- Understand desktop security
- Implement password security

2.2 INTRODUCTION

Securing the modern business network and IT infrastructure demands an end-to-end approachand a firm grasp of vulnerabilities and associated protective measures. While such knowledge cannot thwart all attempts at network incursion or system attack, it can empower network engineers to eliminate certain general problems, greatly reduce potential damages, and quickly detect breaches. With the ever-increasing number and complexity of attacks, vigilantapproaches to security in both large and small enterprises are a must. Prior to discussing Procedural / People security controls we will start by defining security controls in general.

Security Controls are safeguards or countermeasures to avoid, detect, counteract, or minimize security risks to physical property, information, computer systems, or other assets. Controls help to reduce the risk of damage or loss by stopping, deterring, or slowing down anattack against an asset.

People are truly the weakest link in any security schema. Most people are not careful about keeping secrets such as passwords and access codes that form the basis for most secure systems. All security systems rely on a set of measures employed to control access, verify identity and protect disclosure of sensitive information. These measures usually involve one or more “secrets”. Should a secret be revealed or stolen then the systems that are protected bythese secrets can be compromised. It may seem like a terribly obvious statement, but most systems are compromised in very basic ways. Leaving a Post-It note with a system password stuck to the side of a computer monitor may seem foolish, but many people in fact do such things. Another example, which is only slightly less obvious, is the tendency to leave factory default passwords in certain network devices. One such device might be a network management interface to a UPS. UPS systems, whether small in capacity or large enough to power 100 servers, are often overlooked in a security scheme. If such devices are left with default usernames and passwords, it could just be a matter of time before someone gains access knowing nothing more than the device type and its published default credentials. Imagine a server bank with rock solid security protocols on each web and mail server crashedby a simple power cycle on an unprotected UPS!

The control environment sets the tone of an organization, influencing the control consciousness of its people. It is the foundation for all other components of internal control, providing discipline and structure. Control environment factors include the integrity, ethical values, and competence of the entity’s people; management’s philosophy and operating style;and the way management assigns authority and organizes and develops its people

2.3 SECURITY BASICS

It is not possible to protect anything unless one clearly understands WHAT one wants to protect. Organizations of any size should have a set of documented resources, assets and systems. Each of these elements should have a relative value assigned in some manner as to their importance to the organization. Examples of things that should be considered are servers, workstations, storage systems, routers, switches, hubs, network and Telco links, and any other network elements such as printers, UPS systems and HVAC systems. Other important aspects of this task include documenting equipment location and any notes on dependencies. For instance most computers will rely on power backup systems such as UPSswhich themselves may be part of the network if they are managed. Environmental equipmentsuch as HVAC units and air purifiers may also be present.

The next step is to identify the potential “threats”. Threats can come from both internal and external sources. They may be human based, automated or even non-intentional natural phenomenon. The latter might more appropriately be categorized under system health threatsas opposed to security threats, but one issue can lead to the other. One example is a power outage to a burglar alarm. The power outage could be intentional or through some natural event such as a lightning strike. In either case security is diminished.

To help review or design security controls, they can be classified by several criteria, for example according to the time that they act, relative to a security incident:

a. **Preventive Controls** are intended to prevent an incident from occurring i.e. these controls are implemented before the event takes place. They exist to prevent the threatfrom coming in contact with the weakness, e.g. by locking out unauthorized intruders.
b. **Detective Controls** are applied during the event and are intended to identify and characterize an incident in progress. These exist to identify that the threat has landed in our systems, e.g. by sounding the intruder alarm and alerting the security guards orpolice.
c. **Corrective Controls** are executed after the event and are intended to limit the extent of any damage caused by the incident. In other word they exist to mitigate or lessen the effects of the threat being manifested, e.g. by recovering the organization to normal working status as efficiently as possible.

Computer security is often divided into three distinct master categories, commonly referred toas controls:

a. Physical
b. Technical
c. Administrative

These three broad categories define the main objectives of proper security implementation. Within these controls are sub-categories that further detail the controls and how to implementthem.

2.3.1 Physical Controls

The Physical control is the implementation of security measures in a defined structure used todeter or prevent unauthorized access to sensitive material. Examples of physical controls are:

i. Closed-circuit surveillance cameras
ii. Motion or thermal alarm systems
iii. Security guards
iv. Picture IDs
v. Locked and dead-bolted steel doors

2.3.2 Technical Controls

The Technical control uses technology as a basis for controlling the access and usage of sensitive data throughout a physical structure and over a network. Technical controls are far- reaching in scope and encompass such technologies as:

2.3.2.1 Encryption

2.3.2.2 Smart cards

2.3.2.3 Network authentication

2.3.2.4 User Access control

2.3.2.5 File integrity auditing software

2.3.3 Administrative Controls

Administrative controls define the human factors of security. It involves all levels of personnel within an organization and determines which users have access to what resources and information by such means as:

2.3.3.1 Training and awareness

2.3.3.2 Disaster preparedness and recovery plans

2.3.3.3 Personnel recruitment and separation strategies

2.3.3.4 Personnel registration and accounting

2.3.4 Physical security, protection on the inside

Most experts would agree that all security starts with physical security. Controlling physical access to machines and network attach points is perhaps more critical than any other aspect ofsecurity. Any type of physical access to an internal site creates a major exposure of the site. Secure files, passwords, certificates and all sorts of other data can usually be obtained if

physical access is possible. Fortunately there are all sorts of access control devices and securecabinets that can help with this problem.

2.3.5 Partitioning and protecting network boundaries with firewalls

Besides the basic physical security of a site, the next most important aspect is controlling digital access into and out of the organization's network. In most cases this means controllingthe points of connectivity to the outside world, typically the Internet. Almost every medium and large-scale company has a presence on the Internet and has an organizational network connected to it. In fact there is a large increase in the number of smaller companies and homes getting full time Internet connectivity. Partitioning the boundary between the outside Internet and the internal intranet is a critical security piece. Sometimes the inside is referred to as the "trusted" side and the external Internet as the "un-trusted" side. As a generality this is all right, however, as will be described, this is not specific enough.

A firewall is a mechanism by which a controlled barrier is used to control network traffic intoAND out of an organizational intranet. Firewalls are basically application specific routers. They run on dedicated embedded systems such as an internet appliance or they can be software programs running on a general server platform. In most cases these systems will have two network interfaces, one for the external network such as the Internet and one for theinternal intranet side. The firewall process can tightly control what is allowed to traverse from one side to the other. Firewalls can range from being fairly simple to very complex. As with most aspects of security, deciding what type of firewall to use will depend upon factors such as traffic levels, services needing protection and the complexity of rules required. The greater the number of services that must be able to traverse the firewall the more complex the requirement becomes. The difficulty for firewalls is distinguishing between legitimate and illegitimate traffic.

What do firewalls protect against and what protection do they not provide? Firewalls are like a lot of things; if configured correctly they can be a reasonable form of protection from external threats including some denial of service (DOS) attacks. If not configured correctly they can be major security holes in an organization. The most basic protection a firewall provides is the ability to block network traffic to certain destinations. This includes both IP addresses and particular network service ports. A site that wishes to provide external access to a web server can restrict all traffic to port 80 (the standard http port). Usually this restriction will only be applied for traffic originating from the un-trusted side. Traffic from the trusted side is not restricted. All other traffic such as mail traffic, ftp, snmp, etc. would notbe allowed across the firewall and into the intranet.

An even simpler case is a firewall often used by people with home or small business cable orDSL routers. Typically these firewalls are setup to restrict ALL external access and only allow services originating from the inside. A careful reader might realize that in neither of these cases is the firewall actually blocking all traffic from the outside. If that were the case how could one surf the web and retrieve web pages? What the firewall is doing is restricting connection requests from the outside. In the first case all connection requests from the inside are passed to the outside as well as all subsequent data transfer on that connection. From the exterior, only a connection request to the web server is allowed to complete and pass data, allothers are blocked. The second case is more stringent as connections can only be made from the interior to the exterior.

More complex firewall rules can utilize what is called “stateful inspection” techniques. This approach adds to the basic port blocking approach by looking at traffic behaviours and sequences to detect spoof attacks and denial of service attacks. The more complex the rules, the greater the computing power of the firewall required.

One problem most organizations face is how to enable legitimate access to “public” servicessuch as web, ftp and e-mail while maintaining tight security of the intranet. The typical approach is to form what is known as a DMZ (demilitarized zone), a euphemism from the cold war applied to the network. In this architecture there are two firewalls: one between the external network and the DMZ, and another between the DMZ and the internal network. All public servers are placed in the DMZ. With this setup, it is possible to have firewall rules which allow public access to the public servers but the interior firewall can restrict all incoming connections. By having the DMZ, the public servers are still provided more protection than if they were just placed outside a single firewall site.

2.4 SUER NCCEUU CEATRESU

What is an access control? We can say it’s a way to manage access to enterprise resources. It we go with the word definition, access control is a mechanism to control the flow of information between subject and object where subject is always as active entity while object is a passive entity. In its broadest meaning, access control is a three-step process that includes identification, authentication. In this white paper, the term authentication is generally used torepresent both identification and authentication, and access control is used for authorization.

The section discusses the importance of selecting an access control model that fits with your security needs to provide a lower total cost of ownership and enable strong identification. It also discusses the various authentication solutions and weights their need to your organization.

2.4.1 Why Access Controls are required

As the business of an enterprise increases, so does the demand for access control since it is the first line of defense to protect the organization’s resources. When you try to access a certain resource and you are asked to provide your identification – that is access control. Access control solutions provide protection, integrity, availability and auditing capability to the organisation.

2.4.2 What are Access Control Models

Access control model is a framework that dictates access control using various access-control technologies. There are standard access control models which are highly domain and implementation independent. Each access control model has its own merits and demerits, and the specific business objectives they serve depend on the organization’s need, culture, natureof business, etc. We will discuss these models and examine their fitness with respect to an organization’s security policy and business goals.

2.4.2.1 Discretionary Access Control (DAC)
2.4.2.2 Mandatory Access Control (MAC)
2.4.2.3 Role Based Access Control (RBAC)

2.4.2.1 Discretionary Access Control (DAC)

Discretionary Access Control is based on ownership and delegation. In a DAC Model, access is governed by the access rights granted to the user groups. An organisation/administrator/creator can identify a set of operations and assign them to an object and to set of users and to a set of users (belonging to user group).

The DAC model is flexible but complex. It creates a paradox in some complex situations. Forexample, A is owner of resource R of organisation O and he has delegated permission P1 andP2 to B who, in turn, has delegated permission P1 to C. Now, it A chooses to revoke permission to B what will happen to the permission that B granted to C?

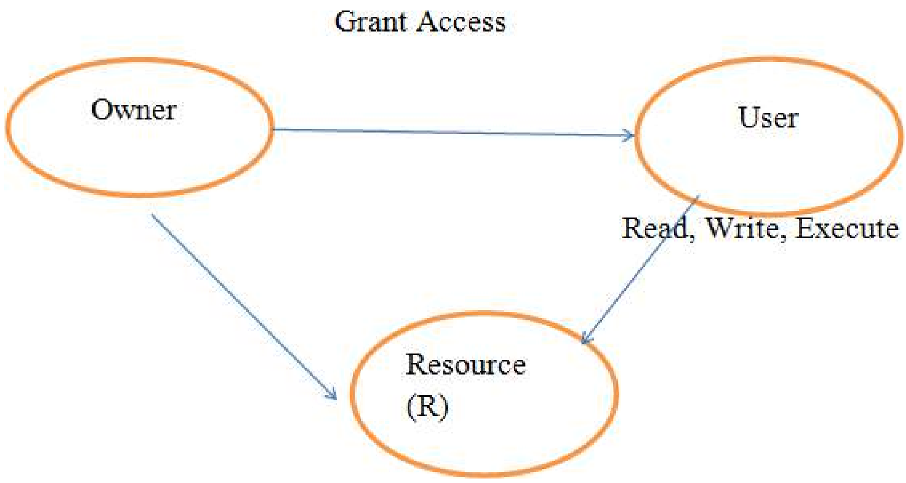


Figure 3: Discretionary Access Control

While the model above is complex, it is still flexible enough to handle various access controlneeds, and therefore is used in various network management applications.

2.4.2.2 Mandatory Access Control (MAC)

In MAC, the data owner has limited freedom to decide on access control. Information is classified into different categories and each category is assigned a particular security level. For example, resource R is a very confidential resource to the organisation and so has been assigned a "Very Confidential" security level. When a user, who has been assigned security level "Confidential", tries to access this resource, he is denied access because the security level assigned to him doesn't match. This model is appropriate when securing confidential ofdata is critical, as in—for example—military operation systems.

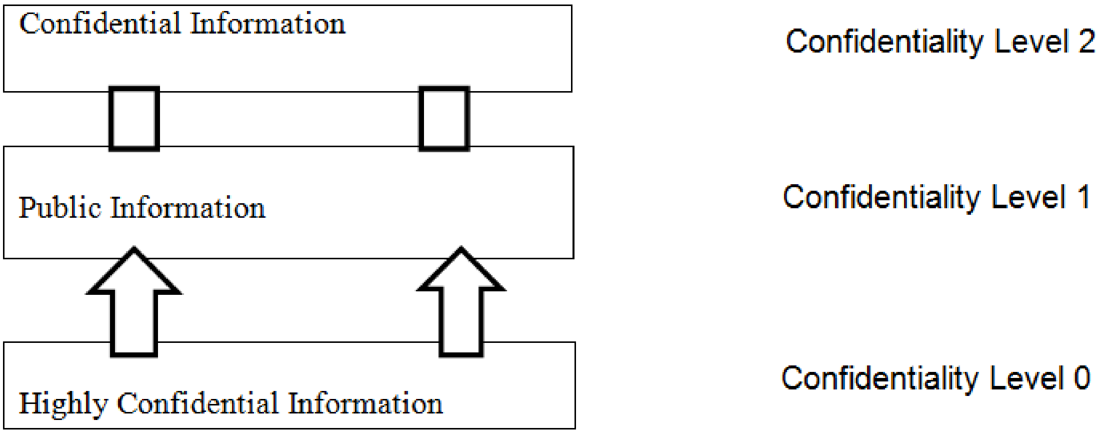


Figure 4: Mandatory Access Control

2.4.2.3 Role Based Access Control (RBAC)

RBAC is a widely used- and dominant- access control model, and most access control security products available in the market today are based on this model because its objectivesare architectural. Entrust Get Access is one such product. The model allows access to a resource, based on the role the user holds in the organization. It is based on the concept of "separation of duties". The privileges to the particular role are decided and thereafter mappedto the user. If the environment does not require a high level of security, the choices are usually discretionary and role based. The discretionary model gives data owners the ability toallow users to access resources, enabling choices to be made with full knowledge of what it entails. If the organization has a high turnover rate, the role based model is more appropriate.If the environment requires higher level of security and it is desired that only administrator should grant the access control, then MAC is the best choice.

2.4.3 Authentication

Authentication or identification is the first step in any access solution. It is the process of identifying the user to verify whether he/she is what he/she claims to be. Normally, identification is done with the help of information that is known to everyone (i.e., user name or user ID) and some personal information known only to the subject (i.e. password). Faced with the threat of identity theft and increasing consequences associated with failing to secure information, enterprises are increasingly looking for stronger forms of authentication to enhance their overall security capabilities. At the same time, enterprises and governments need to take into account other important considerations such as usability, total cost of deployment and maintenance, and integration with existing security solution offerings. Usernames and passwords are the most common authentication techniques. But most organizations do not depend on user name authentication alone since username and passwords are an authentication solution for low-value transactions and for accessing non- sensitive information over the network. Also, experience has shown that usernames and passwords provide relatively weak authentication because they can often be guessed or stolen. They are often difficult to deploy because each application may implement its own

scheme, adding to both development cost and user complexity. Also, it is very difficult to maintain and reset the password.. Determining the appropriate level of authentication that meets your budget requirements is essential when implementing your secure identity management solution. It is very crucial to identify the appropriate authentication technique depending upon the nature of the business and sensitivity of the information. One has to consider various authentication methods and their pros and cons. The means of authentication are often discussed in terms of "factors" of proof, such as:

2.4.3.1 Something your know to prove your identity (e.g.a PIN)
2.4.3.2 Something you have to prove your identity (e.g. a smart card)
2.4.3.3 Something you have to prove your identity (e.g. a fingerprint)

A good authentication technique contains at least two of the above methods. In a client server environment, strong authentication is a combination of server and client authentication:

2.4.3.4 Server authentication is when the server proves its identity to the client.

2.4.3.5 Client authentications are when clients prove their identity to the server.

2.4.1.1 User password Authentication

It is the most common form of providing identification. When user accesses the resource, access control framework asks for the user name password provided to the user. The credentials are validat3d against the one stored in the system's repository.

2.4.1.2Windows user based authentication

Usually, organizations have a list of users stored in the windows active directory. Access control framework should be able to provide authentication for the user of the Primary Domain Controller (PDC).

2.4.1.3Directory based authentication

With the rising volume of business over the web, millions of user's often try to access the resource simultaneously. In such a scenario, the authentication framework should be able to provide for faster authentication. One such technique is Directory Based Authentication where user credentials are validated against the one which is stored in the LADAP Directory.

2.4.1.4Certificate based authentication

This is probably one of the strongest authentication techniques where the user is asked to provide his/her digital ID. This digital ID, known as digital certificate, is validated against the trusted authority that issued the digital ID. There are various other parameters that ate checked to ensure the identification of the user.

2.4.1.5Smart card based authentication

This is also used as a second factor authentication. Smart cards are small devices containing co-processors to process cryptographic data.

2.4.1.6Biometrics

This is the strongest authentication. Known as third factor authentication, it is based on something the user is. It works after the users have provided something the know (User namepassword) and something they own (either a gird or token) or something they are (retina- scan, thumbprint or thermal scan). It is required in cases where data is top confidential, such as in Military/Defense.

2.4.1.7Grid based Authentication

This is used as a second factor authentication. It authenticates the user based on something he knows (User name password authentication) and then asks for something he owns (gird card information). Entrust Identity Guard provides such an authentication.

2.4.1.8Knowledge-based authentication

One of the simplest mechanisms for gaining additional confidence in a user's identity is to challenge the user to provide information that an attacker is unlikely to be able to provide. Based on "shared secrets", this allows for the organization to question the user, when appropriate, to confirm information that is already known about the user through a registration process, or from previous transactions.

2.4.1.9Machine Authentication

Machine authentication provides validation of the user’s computer in a way that secures against a variety of threats in a zero touch fashion, reducing user impact. This is an especially effective method of user authentication where users typically access their accounts from a regular set of machines, allowing for stronger authentication to be performed without any significant impact on the user experience.

2.4.1.10 One time Password (OTP)

A one-time password is dynamically generated and it is valid only for once. The advantage ofone time password is that if an intruder hacks it, he cannot reuse it. These are two types of OTP token generators: synchronous and asynchronous. A synchronous token device synchronizes with the authentication service by using time or an event as the core piece of the authentication process. A token device, which is using as asynchronous token generating method, uses a challenge response scheme to authenticate the user.

2.4.2 Access Control Framework (ACF)

The access control framework (ACF), presented in this paper, is like an umbrella that covers both authentication and authorization. Whenever the user accesses any enterprise resource, ACF can come up with one or more authentication techniques depending on the need of the enterprise. Once the authentication is done, ACF can authorize the request, following any model, depending on the need of the organization.

2.4.3 Access Control Techniques and Technologies

Once an organization decides on the type of access control model to be employed, the next step would be to decide on the techniques and technology to be used. Here are some techniques and technologies:

2.4.3.1Rule Based Access control

Rule based access control is based on rules defined on the object, as defined by the administrator who decides on the operations that can be performed by subject. A rule can be as simple as defining the day of the week on which the resource can be accessible.

2.4.3.2 Menu Based Access Control

In a menu based control, the user interface given to the user controls the operations that can be performed on the object, i.e., If A and B operations can be performed on object O, then theuser interface pertaining to A and B options is enabled and the rest of the user interface is disabled.

2.4.3.3Access Control List

Access control list is the list of subjects that are authorized to access a particular object. It also defines the level of authorization.

2.4.3.4Content Based Access Control

In content based access control (CBAC), the access to the object is determined by the contentwithin the object. For example, a manager can access the payroll database but only for employees reporting to him.

2.4.3.5 Access control Markup Language (XACML)

XACML is the access control markup language that is used to express the rules that are necessary for authentication and authorization. The vocabulary to express these rules is givenby the access control markup language. These rules are used to make decisions regarding the authentication. eXtensible Access Control Markup Language – or XAVML – provides a Mechanism to create policies and rules for controlling access to information.

A typical access control and authorization scenario includes three main entities - a subject, a resource, and an action - - and their attributes. A subject makes a request for permission to perform an action on a resource. For example, in the access request, “Allow the Sys-admin tocreate files in the root folder of the production server” the subject is the “Sys-admin”, the target resource is the “root folder of the production server”, and the action is “create files”.

2.4.3.6 Security Assertion Markup Language (SAML)

SAML is an XML standard for exchanging authentication and authorization data between security domains, that is, between an identify provider (a producer of assertions) and a service provider (a consumer of assertions). SAML is a product of the OASIS Security Services Technical Committee.

The single most important problem that SAML is trying to solve is the Web Browser single sign-on (SSO) problem. Single sign-on solutions are abundant at the intranet level (using cookies, for example) but extending these solutions beyond the intranet has been problematicand has led to the proliferation of non-interoperable proprietary technologies. SAML has become the definitive standard underlying many web single sign-on solutions in the enterprise problem space.

The whole thrust of access control is to restrict unauthorized users from accessing organization resources. The authentication techniques and access controls described in this white paper can be chosen based on an organization’s need. The authentication and access control framework should be flexible enough to serve all the authentication techniques and future evolution in the area such as Biometrics. The access control framework should be ableto handle an organization’s authentication and authorization (access control) needs. Entrust Get Access and Identify Guard is the products with such features.

2.5 TRAINING AND AWARENESS

One of the greatest threats to information security could actually come from within your company or organization. Inside ‘attacks’ have been noted to be some of the most dangerous since these people are already quite familiar with the infrastructure. It is not always disgruntled workers and corporate spies who are a threat. Often, it is the non-malicious, uninformed employee.

The focus will be on uninformed users who can do harm to your network by visiting websites infected with malware, responding to phishing e-mails, storing their login information in an unsecured location, or even giving out sensitive information over the phone when exposed tosocial engineering.

One of the best ways to make sure company employees will not make costly errors in regard to information security is to institute company-wide security-awareness training initiatives that include, but are not limited to classroom style training sessions, security awareness website(s), helpful hints via e-mail, or even posters. These methods can help ensure employees have a solid understanding of company security policy, procedure and best practices. Some of the more important items to cover in your security awareness training are your organization's security policy, data classification and handling, workspace and desktop security, wireless networks, password security, phishing, hoaxes, malware, file sharing and copyright.

2.5.1 Types of Training

Organizations are starting to realize there really is a need for security awareness training. According to a study conducted, the following statistics revealed a rather startling necessity:

- "One in five workers (21%) let family and friends use company laptops and PCs to access theInternet".
- "More than half (51%) connect their own devices or gadgets to their work PC... a quarter ofwho do so every day".
- "One in ten confessed to downloading content at work they should not".
- "Two thirds (62%) admitted they have a very limited knowledge of IT Security".
- "More than half (51%) had no idea how to update the anti-virus protection on their companyPC".

- "Five percent say they have accessed areas of their IT system they should not have".Security awareness training can be performed in a variety of ways that can be utilized alone or in conjunction with each other. Those mediums can consist of a more thorough classroom style training, creation of a security-awareness website, pushing helpful hints onto computerswhen they start up and/or e-mailing helpful hints on a weekly or monthly basis, and utilizing visual aids like posters.

2.5.1.1Classroom-Style Training

Utilizing a classroom setting for security-awareness training can offer the benefit of lecture- based and interactive learning as well as the availability of someone to answer questions in real time. There can also be a Q&A period after the materials are presented as well as contactinformation distributed for questions that might pop up afterward.

Some companies offer both live and web-based training and utilize a variety of methods suchas role-playing and simulation games so the interaction is more two-way than one-way. Othercompanies offer videos, web-based training, and live trainers. The method you use is by no means limited.

This type of training can vary in the amount of time it can take. The security awareness training I have implemented at Washtenaw Community College takes about two hours, but ithas no real interactivity such as role-playing or simulations; just PowerPoint and Q&A. Training time can depend on the effectiveness and the extent of the material discussed. Training sessions could possibly take a full day if need be.

2.5.1.2Security Awareness Website

Another way of implementing a security awareness program is through the creation of a security awareness website. This website could consist of different sections with the differentareas that need to be covered (e.g. Malware, hoaxes, file sharing and copyright, etc). The University of Tennessee implemented a very impressive security awareness website completewith videos, examples, and helpful external links.

Another implementation of the security awareness website could be a self-paced tutorial where users can log in and go through it, taking mini quizzes at the end of each section to make sure the material is actually being read and absorbed. Utilizing logins can also be a means of keeping track of who has (and more importantly who has not) taken the training. AnFAQ section could be implemented as well as contact information for users to ask questions that are not addressed in the FAQ.

2.5.2 Helpful Hints

Utilizing helpful hints and tips is more of a supplement to the training, be it via classroom style or online, and should not be used as a means of security awareness training on its own. Helpful hints can consist of tips and reminders that are pushed to user screens when they log in. These tips and reminders can consist of key points emphasized in the training (e.g. "Neverkeep your password in a place that can be accessed or viewed by anyone besides yourself"). Reminders can be as simple as reminding someone to change their password or run their virus scan.

2.5.2.1Visual Aids

Visual aids are another item that should not be used as the lone source of security awareness training, but more as a supplement. The University of Michigan recently created a series of catchy password security posters that compare passwords to underwear. One says to change them often, another says to not leave passwords lying around, and another one says to not share them with friends.

2.5.2.2Promotions

Security tips can appear on flyers distributed across the user base and one could even go so far as to hand out pencils and/or key chains with a catchy security-related phrase or reminder(e.g. "Unexpected attachments can mean unexpected chaos: Please do not open them"). Nowthat we have addressed possible methods in implementing security awareness training, what should be covered in the training will be addressed.

2.5.2.3Training Topics

Topics addressed by the security awareness training should consist of a combination of existing organizational policies and procedures (how they tie in with each aspect, if they do), physical security, desktop security, password security, phishing, hoaxes, malware (viruses, worms, Trojans, spyware, and adware), and copyright with regard to file sharing. These topics will help employees understand why security awareness is important and guide them inknowing how to prevent incidents from happening and what to do if one occurs.

2.5.2.4Physical Security

When addressing physical security, locking your doors and desk/file cabinet drawers should be the main focus. A helpful item to include could be the crime statistics, more specifically thefts, from the organization. Another item to lightly touch upon (but go into greater detail inDesktop Security) is the fact that if a potential attacker has access to a user's computer, they could install a key logger or actually get into a machine that has not been locked.

2.5.2.5Desktop Security

The desktop security section should go into detail as to why it is important to either have a password-protected screen saver or, even better, to get into the habit of locking computers when users walk away from them. A screensaver timeout should be utilized so if a user walksaway from their computer, the password-protected screensaver would come up. Personally, I have mine set to 5 minutes, but upon doing a Google search regarding typical screensaver timeouts, the average response was 10 minutes. This information can and should be supplemented with information on how to do this. Tactics a potential attacker could utilize (e.g. Shoulder surfing, key loggers, etc) also need to be addressed. The pill of having to take extra measures to make sure your desktop is secured can be swallowed more easily if users understand WHY they should be taking them.

Another item that could be addressed is to make sure users understand that it is important thatthey shut down their computers at the end of the day. Sometimes this allows for valuable updates to be applied and doing your own part for a greener environment. If somehow a potential attacker gains access to a computer that is turned off, they will be less likely to utilize it than one that is already turned on and unlocked.

2.5.2.6Wireless Networks and Security

The wireless networks and security section should address the unsecure nature of wireless networks as well as tips and tricks to exercise caution and harden laptops against the dangers of 'sniffing'. Emphasis should also be placed on not storing any kind of sensitive informationon laptops that will be accessing a wireless network. Another area that should be covered is the importance of firewalls. Windows Firewalls by themselves are not enough. Most times companies will provide a purchased firewall on company-supplied laptops and computers butpersonal laptops that may utilize the company wireless network need to have a firewall on them. For small office environments as well as those who remotely access their workstation from home, it is always helpful to provide information on free firewall options like ZoneAlarm and Comodo as well as relatively inexpensive firewall options like McAfee and Norton. The free firewalls are more for the personal user, though and not for commercial use.It may also be a benefit to the training to compare the price of a laptop to the price of a breach.

2.5.2.7Password Security

The password security section should include what constitutes a strong, secure password or passphrase, with an emphasis on passphrases since they are harder to guess and to crack. Thissection should also outline the minimum password requirements of the organization.

Sharing passwords as well as leaving them out where anyone but the user could access them should be strongly discouraged. Making this part of organization-wide policy could be very helpful in this arena. If this is incorporated into policy, this should be addressed in the training. Users need to be aware a policy is in place and general "rules of thumb" to make sure these policies are followed. Statistics could also be a good supplement. For example, a delegated individual could go around to all of the offices and see if they can uncover any unsecured passwords. They could even take this a step further and see how many computers are left on as well as without password-protected screensavers. No specific individuals wouldbe singled out; just a number of instances out of the total number of computers would suffice.

Helpful hints and rules of thumb should also be a part of this section. For example, passwordsshould not contain the username or any part of the user's full name. Passwords also should not be based on personal information such as a spouse name, favourite team, or pet. Another important point is to stress that the default password given to users should always be changedimmediately. Instructions on how to change passwords should also be included.

To round out the password security section, it can be very beneficial to define what constitutes a poor choice of password as well as a listing of the most common passwords used.

2.5.2.8Phishing

When discussing phishing, the term as well as the purpose should always be defined. Examples are key to this portion of security awareness training. Things to avoid (e.g. clicking on links provided in e-mail, submitting banking and password information via email, etc.) should be highly emphasized so people know what to look for. It could also be beneficial to have users take a Phishing IQ Test. This way the bits and pieces that can identify a phishing e-mail can be explained and displayed. Another item that should be addressed is how to actually fight phishing attacks. A couple of web sites actually encourage the reporting and tracking of phishing web sites and e-mails: PhishTank (www.phishtank.com) and The Anti- Abuse Project (www.antiabuse.org), which address these issues.

2.5.2.9Hoaxes

Hoaxes should be addressed in the training because a lot of time and resources can be spent reading and forwarding hoax emails. The types of hoaxes as well as examples should be the meat of this section. Using familiar hoaxes is the best option so it will be easier to grasp. It could also be beneficial to compare hoaxes to viruses in that they are spread by continually forwarding them. The dangers of hoaxes should also be addressed because some hoaxes warnof a virus and tell users to delete valid and sometimes important system files.

Preventing the spread of hoaxes should also be covered. Hoaxes can be prevented by checking a number of hoax sites on the Web and following a few rules of thumb. It is important to point out that if something sounds too good to be true, it probably is and if something seems suspicious it can be checked on one of the hoax sites

2.5.2.10 Malware

When addressing malware, it should always be defined and then broken down into its categories: viruses, worms, Trojans, spyware, and adware. After each category is broken down, address how they end up on systems.

2.5.2.11 Viruses

Start out by outlining what makes a virus a virus. It is important for users to be able to identify a potential virus when they see one or to identify characteristics of a virus that has already infiltrated the user’s system. What a virus is capable of is also something that should supplement the defining of what makes a virus what it is.

Defining what a virus is and how to identify one must be complemented with the important ofantivirus software. Most organizations will have this installed on all organization-wide computers, but this might not be installed on laptops used by employees. Users also need to learn the importance of not only performing regular scans of their computers, but also of any file they download from a web site, e-mail, or thumb drive.

Another important tip to include is how vital it is to keep systems and applications up-to-date.Never assume that a system or application is always going to update itself. Users should proactively see if the systems and applications they are using need updated.

Finally, it is important to let users know what to do if their system does become infected. Make sure not to incite a sense of panic that would steer employees toward hiding the infection until it has gotten out of control or their machine is beyond repair. The main procedure to address is what to do if and when a virus infects a work machine, since it woulddiffer considerably to what to do at home.

When your work machine becomes infected, do not do anything to the computer aside from performing a scan with the anti-virus software on the machine. Phone the I.T. Department of your business to come evaluate your machine and hopefully get rid of the virus.

If your machine at home (especially if you work from home) becomes infected, it is important to follow the following steps outlined on Viruslist.com:

- Do not panic
- Disconnect from the Internet and any Local Area Network it may be connected to.
- If computer cannot boot, try starting in Safe Mode or boot from the Windows bootdisk.
- Back up any important data you cannot afford to lose to an external drive (scan thefile with your anti-virus software first) (floppy, CD, jump drive, etc).
- If you do not have anti-virus software installed (which SHOULD not be the case),install it and then update it.
- Perform a full scan of your system.

2.5.2.12 Worms

The worms section can be handled much the same way the virus section is handled: Definition, how to spot, what it is capable of, how to prevent, what to do if one invades the system.

2.5.2.13 Trojans

Like the previous 2 sections, the Trojans portion should define what they are, what they can do, what can be done to prevent them, and what to do in the event of one making it onto the system. One item that should be emphasized is that Trojans are different from viruses and why they are two different things.

2.5.2.14 Spyware and Adware

Again, spyware and adware should be defined, what they can do should be outlined, prevention tips and tricks, and then what to do if it is found on the system. Spyware and adware identification and removal programs should also be addressed, most of which are free(e.g. Ad Aware, Spy Sweeper, etc).

2.6 SUMMARY

1. Security Controls are safeguards or countermeasures to avoid, detect, counteract, or minimize security risks to physical property, information, computer systems, or otherassets.
2. All security systems rely on a set of measures employed to control access, verify identity and protect disclosure of sensitive information.
3. The control environment sets the tone of an organization, influencing the control consciousness of its people.

4. Control environment factors include the integrity, ethical values, and competence ofthe entity's people; management's philosophy and operating style.
5. Controlling physical access to machines and network attach points is perhaps morecritical than any other aspect of security.
6. Besides the basic physical security of a site, the next most important aspect is controlling digital access into and out of the organization's network.
7. Partitioning the boundary between the outside Internet and the internal intranet is acritical security piece.
8. A firewall is a mechanism by which a controlled barrier is used to control networktraffic into and out of an organizational intranet.
9. The most basic protection a firewall provides is the ability to block network traffic tocertain destinations.
10. Authentication or identification is the first step in any access solution.

2.7 CHECK YOUR PROGRESS

1. Controls help to reduce the risk of damage or loss by stopping, deterring, or slowingdown an attack against an _________.
2. __ _________ are truly the weakest link in any security schema.
3. Organizations of any size should have a set of documented_________ , __ __and ________.
4. __ ___________ controls are intended to prevent an incident from occurring.
5. __ _________controls define the human factors of security.
6. __ _________ are basically application specific routers.
7. __ ___________ is the process of identifying the user to verify whether he/she is what he/she claims to be.
8. __ _______ is known as third factor authentication.

2.8 ANSWERS TO CHECK YOUR PROGRESS

1. Asset
2. People
3. Resources, assets and systems
4. Preventive
5. Administrative
6. Firewalls
7. Authentication
8. Biometrics

2.9 MODEL QUESTIONS

1. What are security controls?
2. What is threat? What are the possible sources of threats?
3. What are Detective Controls? Explain.

4. What are the three distinct master categories
   into which Computer security iscategorized?
5. What is a firewall?
6. What are stateful inspection techniques?
7. What is demilitarized zone in a network?
8. What is access control? Why access controls are required?
9. What are the different access control models?
10. What is access control framework?
11. Why training and awareness is an import aspect of security controls?

UNIT III: SECURITY

STRUCTURE

3.0 OBJECTIVES

After going through this unit, you will be able to:

- Information Security Frameworks
- Types of framework and their advantage.
- Organization structure, roles and responsibilities.
- Overview of Policy, Procedures and Guidelines.

3.1 INTRODUCTION TO INFORMATION SECURITY FRAMEWORK

Information security framework is a collection of documented procedures that are used to describe policies, procedures and guidelines around the implementation and management of Information security controls as per the security requirements of the enterprise requirements. These frame works are used as plans or blueprints to design the security of an information security program to mitigate risk and bring down the impact of the risks under the acceptance criteria. Frameworks are often customized as per the requirement of the organizations. Framework assist enterprise to achieve their objectives and deliver values through effective governance and management.

Advantage of Information Security framework

Framework helps in achieving organizational objective in a systematic and uniformed manner. Few main advantages of using information security framework is given below:

- Maintaining of processed information to support business decision.
- Achieving strategic objectives and provide benefit through effective use of resources.
- Maintain risk at acceptance level.
- Optimize the cost of IT services and technology.
- Supporting compliance with relevant laws, regulation, contractual agreement and policies.

Many Standards, Best Practices and Frameworks

In the coming section, we will see many profitable and non-profitable organizations have made their own methodologies to security management, security control objectives, process management and enterprise management

Basic break of these standards, frameworks are given below:

Security Program Development

- **ISO/IEC 27000 series** developed by ISO and IEC for the development and management of ISMS.

Enterprise Architecture Development

- **Zachman framework** is developed by ZohnZazhman for the development of enterprise architectures.
- **TOGAF** Model developed by the open group for the enterprise architectures development.
- **MODAF** Architecture framework used mainly in military support missions developed by the British Ministry of Defense.

Security Enterprise Architecture Development

- **SABSA model,** Model and methodology for the development of information security enterprise architectures.

Security Controls Development

- **COBIT** Set of control objectives for IT management developed by Information Systems Audit and Control Association (ISACA) and the IT Governance Institute (ITGI)
- **SP 800-53**Set of controls to protect U.S. federal systems developed by the National Institute of Standards and Technology(NIST).

Corporate Governance

- **COSO**is a set of internal corporate controls to help decrease the risk of financial fraud developed by the Committee of Sponsoring Organizations (COSO) of the Treadway Commission.

Process Management

- **ITIL** Processes to permit for IT service management developed by the United Kingdom's Office of Government Commerce.
- **Six Sigma** Business management strategy that can be used to carry out process improvement.
- **Capability Maturity Model Integration (CMMI)** Organizational development for process improvement developed by Carnegie Mellon.

ISO 27001:2013

ISO 27001:2013, is an information security standard that was published in September'2013.It is revised version of ISO 27001: 2005, and is published by ISO (International Organization of Standards) and IEC(International Electrotechnical commission). This standard is specifically target to develop and maintain Information Security Management System in the organization. ISO 27001 gives the requirement for the implementation of ISMS (Information Security Management System).

Information Security Management System (ISMS) defines the control that need to be placed (configuration management, physical security management, data protection, auditing etc.) and explains how these should be treated during their whole lifecycle. ISMS provide a complete picture of the security by aligning and placing controls strategically in the organization. ISMS components should be integrated within the whole organization is should not be practiced in certain departments of the organization.

ISO 27001:2013 has 14 domains and 114 controls. Refer to ISO 27001:2013 to understand exactly the control structure. Due to intellectual property right we could list exact controls of ISO 27001:2013 but control structure and their purposes are given below:

1. **A.5 Information security policies** – It defines the control on how policies are documented and reviewed.
2. **A.6 Organization of information security** – It defines the control on the responsibilities given to different individuals.
3. **A.7 Human resources security** – It defines the control before the employment, during the employment and after employee left the organization.
4. **A.8 Asset management** – It defines the controls on information classification,

media handling and inventory of assets.

5. **A.9 Access control** – It defines controls on accessing user access management, application, server and user responsibilities along with them.
6. **A.10 Cryptography** – It defines control related to encryption and decryption.
7. **A.11 Physical and environmental security** – It defines controls mentioning secure areas, access control for entrance and exit, equipment security, protection against threats, secure disposal, clear desk and clear screen.
8. **A.12 Operational security**– It defines controls related to change management, capacity management, backup, logging, monitoring, installation, vulnerabilities etc.
9. **A.13 Communications security**– It defines control related to network security, network services, transfer of information.
10. **A.14 System acquisition, development and maintenance** – It defines control for mentioning security requirement and development and support process.
11. **A.15 Supplier relationships** – It defines control on agreements controls on what to include in agreements, and how to monitor the suppliers
12. **A.16 Information security incident management** – It defines controls for reporting incidents, defining weakness, response procedure and collection of evidence.
13. **A.17 Information security aspects of business continuity management** – It defines the controls related to the plan of business continuity, procedures, verification and reviewing.
14. **A.18 Compliance** – It defines controls requiring the identifying applicable laws and regulation on intellectual property protection of personal data etc.

Table 1: Comparison of ISO 27001:2005 to ISO 27001:2013

Context	ISO 27001:2005	ISO 27001:2013
Process	The standard clearly states that it follows PDCA (Plan-Do-Check-Act) model.	The Standard does not mention any specific process model.
Risk Assessment	In ISO 27001:2005 asset owner determines how to treat the risk, accepting residual risk.	The Risk assessment and risk treatment plan process are aligned to ISO 31000.
Controls	There are 133 controls across 11 domains.	There are 114 controls across 14 domains.

Documentation	Standard used records and documentation to cover all the requirements. Document include policies, procedure and guidelines. Records include audit, schedules etc. There is no such distinction between control and records.

COSO

COBIT was derived from COSO framework which was developed by the Committee of Sponsoring Organizations of the Treadway Commission (COSO). In 1985 to deal with enterprise risk management, fraudulent activities, internal control and financial reporting. The COSO internal control framework comprises of five interconnected components derived from the way management manages a business. COSO assures that these components provide an effective framework for describing and evaluating According to COSO, these components provide an effective framework for describing and analyzing internal control system integrated in an organization. The five components are the following:

1. **Control environment:** The control environment sets the tone of an organization, influencing the control consciousness of its people. It is the foundation for all other components of internal control, providing discipline and structure. Control environment factors include the integrity, ethical values, management's operating style, delegation of authority systems, as well as the processes for managing and developing people in the organization.
2. **Risk assessment:** Every entity faces a variety of risks from external and internal sources that must be assessed. A precondition to risk assessment is establishment of objectives and thus risk assessment is the identification and analysis of relevant risks to the achievement of assigned objectives. Risk assessment is a prerequisite for determining how the risks should be managed.
3. **Control activities:** Control activities are the policies and procedures that help ensure management directives are carried out. They help ensure that necessary actions are taken to address the risks that may hinder the achievement of the entity's objectives. Control activities occur throughout the organization, at all levels and in all functions. They include a range of activities as diverse as approvals, authorizations, verifications, reconciliations, reviews of operating performance, security of assets and segregation of duties.
4. **Information and communication:** Information systems play a key role in internal control systems as they produce reports, including operational, financial and compliance-related information that make it possible to run and control the business. In a broader sense, effective communication must ensure information flows down, across and up the organization. For example, formalized procedures exist for people to report suspected fraud. Effective communication should also be ensured with external parties, such as customers, suppliers, regulators and shareholders about related policy positions.
5. **Monitoring**: Internal control systems need to be monitored—a process that assesses the quality of the system's performance over time. This is accomplished through ongoing monitoring activities or separate evaluations. Internal control deficiencies detected through these monitoring activities should be reported upstream and corrective actions should be taken to ensure continuous improvement

of the system.

The framework mentioned 17 principles associated with each components.

Table 2: COSO framework

Internal Control Component	**Principles**
Control environment	1. Demonstrate commitment to integrity and ethical values 2. Ensure that board exercises oversight responsibility 3. Establish structures, reporting lines, authorities and responsibilities 4. Demonstrate commitment to a competent workforce 5. Hold people accountable
Risk assessment	6. Specify appropriate objectives 7. Identify and analyze risks 8. Evaluate fraud risks 9. Identify and analyze changes that could significantly affect internal controls
Control activities	10. Select and develop control activities that mitigate risks 11. Select and develop technology controls 12. Deploy control activities through policies and procedures
Information and communication	13. Use relevant, quality information to support the internal control function 14. Communicate internal control information internally 15. Communicate internal control information externally
Monitoring	16. Perform ongoing or periodic evaluations of internal controls (or a combination of the two) 17. Communicate internal control deficiencies

There are certain limitation as Framework recognize that as internal control provide assurance of achieving the organizations objective, but limitation do exist as internal control do not overcome bad judgments, external events etc. which can cause failing of achieving its operational goal. Organization can face the failure from multiple factors:

- Breakdown due to human failures.
- Cases in which management override internal control.
- External event beyond the organizations control.
- Mistakes due to human intervention.

COBIT (IT Governance Framework)

Before understand the COBIT framework we need to understand use and motive of IT governance frameworks. IT governance is a management initiative to develop a structured framework which allow organization to align the IT with the business goals while reducing risk and improving continually. IT governance is a top down approach which require strong management support to be successful in the organization. IT Governance focuses majorly on five areas given below:

1. **Strategic alignment** emphases on guaranteeing the linkage of business and IT plans; describing, maintaining and validating the IT value; and aligning IT operations with enterprise goals.
2. **Value delivery** is about accomplishing the value proposition throughout the delivery cycle, confirming that IT delivers the promised benefits against the strategy, focused on optimizing costs and proving the value of IT.
3. **Resource management** is about the optimum investment in, and the appropriate management of, critical IT resources: applications, information, infrastructure and people.
4. **Risk management** needs risk awareness by senior officers, a clear picture of the enterprise's acceptance of the risk, understanding of compliance and technical requirements.
5. **Performance measurement** monitors strategy implementation, project completion, resource usage, process performance and service delivery, using balanced scorecards that translate strategyinto action to achieve goals.

COBIT (Control objective for information and related technology) is a framework for developing, implementing, monitoring and improving Information technology governance and management practices. The COBIT framework is published by ISACA in 1996. The framework support organization governance by aligning IT goals with business goals. It helps enterprise to drive optimal value from IT by maintaining balance between resources use, benefits and optimizing risk levels. Adoption of COBIT will allow the organization to achieve the following goals:

- Alignment of IT with the business goals.
- Increased the importance of IT to business.
- Risk reduction.
- Continual improvement of IT.
- Development of goals and scorecards for measurement of IT in a structured way.

COBIT has 5 key principles:

- Principle 1: Meeting Stakeholders Needs.
- Principle 2: Covering the enterprise end to end.
- Principle 3: Applying a single, integrated framework.
- Principle 4: enabling a Holistic approach.
- Principle 5: Separating governance from management.

Table 3: Differences between COSO and COBIT

COSO	COBIT

COSO is a model for cooperate governance.	COBIT is a model for IT governance.
COSO deals more at strategic level.	COBIT focuses more at operational level.

3.2 POLICIES, STANDARDS, BASELINES, GUIDELINES AND PROCEDURES

Security Policy

A Security policy is a statement given by the top management that reflects the role of security in the organization. It can be organizational policy, issue related policy or specific system related policy. Organization Security policy defines how the organization security program will be executed, program goals, roles and responsibilities and outlines how enforcement should be carried out. The organization security policy outlines how all security related activities will be carried out in the organization.

Organization security policy should have several important characteristic that should be understood and implemented:

- Policy should be aligned with the business objective, business should not be aligned with the policy.
- It should be easily understood document that is used as a reference point for all employee and management.
- It should be used to induce security into the business functions.
- It should be changed with any business function such as merger with the new company, adoption of new technology or change of management/ownership.
- It should be tracked through version control.
- It should have clear and declarative statements.
- It should be reviewed on regular basis.

The Types of policies are given below:

1 **Regulatory** This type of policy ensures that the organization is following standards set by specific industry regulations (HIPAA, GLBA, SOX, PCI-DSS, etc.). It is very detailed and specific to a type of industry. It is used in financial institutions, healthcare facilities, public utilities, and other government-regulated industries.

2 **Advisory** This type of policy strongly advises employees as to which types of behaviors and activities should and should not take place within the organization. It also outlines possible ramifications if Employees do not comply with the established behaviors and activities. This policy type can be used, for example, to describe how to handle medical or financial information.

3 **Informative** This type of policy informs employees of certain topics. It is not an enforceable policy, but rather one that teaches individuals about specific issues relevant to the company. It could explain how the company interacts with partners, the company’s goals and mission, and a general reporting structure in different situations.

A common hierarchy of security policies is outlined here:

- Organizational policy
- Acceptable use policy
- Risk management policy
- Vulnerability management policy

- Data protection policy
- Access control policy
- Business continuity policy
- Log aggregation and auditing policy
- Personnel security policy
- Physical security policy
- Secure application development policy
- Change control policy
- E-mail policy
- Incident response policy

Guidelines

Guidelines are recommended actions and operational instructions to users, IT staffs, where specific standard does not apply. A guideline is used to determine the course of action according to a set routine. Guidelines are the best practices used to achieve the goals mentioned in the security policy.

Procedure

Procedure are detailed step-by-step that should be accomplished to reach a certain goal. This apply to IT staff, Information security group members and others who need to carry out specific tasks. Procedures are at the lower level where in the documentation series because they are near to the computers and users. They provide detailed steps for configuration.

Procedure practically shows how policy, procedure and guidelines are actually implemented in the practical scenario. If policy states that password should be alpha numeric then procedure specifically explains how to configure the same on the systems.

3.3 Security Control Introduction

Securing the modern business network and IT infrastructure demands an end-to-end approach and a firm grasp of vulnerabilities and associated protective measures. While such knowledge cannot thwart all attempts at network incursion or system attack, it can empower network engineers to eliminate certain general problems, greatly reduce potential damages, and quickly detect breaches. With the ever-increasing number and complexity of attacks, vigilant approaches to security in both large and small enterprises are a must. Prior to discussing Procedural / People security controls we will start by defining security controls in general.

Security Controls are safeguards or countermeasures to avoid, detect, counteract, or minimize security risks to physical property, information, computer systems, or other assets. Controls help to reduce the risk of damage or loss by stopping, deterring, or slowing down an attack against an asset.

People are truly the weakest link in any security schema. Most people are not careful about keeping secrets such as passwords and access codes that form the basis for most secure systems. All security systems rely on a set of measures employed to control access, verify identity and protect disclosure of sensitive information. These measures usually involve one or more “secrets”. Should a secret be revealed or stolen then the systems that are protected by these secrets can be compromised. It may seem like a terribly obvious statement, but most systems are compromised in very basic ways. Leaving a Post-It note with a system password stuck to the side of a computer monitor may seem foolish, but many people in fact do such things. Another example, which is only slightly less obvious, is the tendency to leave factory default passwords in certain network devices. One such device might be a network management interface to a UPS. UPS systems, whether small in capacity or large enough to power 100 servers, are often overlooked in a security scheme. If such devices are left with default usernames and passwords, it could just be a matter of time before someone gains access knowing nothing more than the device type and its published default credentials. Imagine a server bank with rock solid security protocols on each web and mail server crashed by a simple power cycle on an unprotected UPS!

The control environment sets the tone of an organization, influencing the control consciousness of its people. It is the foundation for all other components of internal control, providing discipline and structure. Control environment factors include the integrity, ethical

values, and competence of the entity's people; management's philosophy and operating style; and the way management assigns authority and organizes and develops its people

3.4 SECURITY BASICS

It is not possible to protect anything unless one clearly understands WHAT one wants to protect. Organizations of any size should have a set of documented resources, assets and systems. Each of these elements should have a relative value assigned in some manner as to their importance to the organization. Examples of things that should be considered are servers, workstations, storage systems, routers, switches, hubs, network and Telco links, and any other network elements such as printers, UPS systems and HVAC systems. Other important aspects of this task include documenting equipment location and any notes on dependencies. For instance most computers will rely on power backup systems such as UPSs which themselves may be part of the network if they are managed. Environmental equipment such as HVAC units and air purifiers may also be present.

The next step is to identify the potential "threats". Threats can come from both internal and external sources. They may be human based, automated or even non-intentional natural phenomenon. The latter might more appropriately be categorized under system health threats as opposed to security threats, but one issue can lead to the other. One example is a power outage to a burglar alarm. The power outage could be intentional or through some natural event such as a lightning strike. In either case security is diminished.

To help review or design security controls, they can be classified by several criteria, for example according to the time that they act, relative to a security incident:

a. **Preventive Controls** are intended to prevent an incident from occurring i.e. these controls are implemented before the event takes place. They exist to prevent the threat from coming in contact with the weakness, e.g. by locking out unauthorized intruders.

b. **Detective Controls** are applied during the event and are intended to identify and characterize an incident in progress. These exist to identify that the threat has landed in our systems, e.g. by sounding the intruder alarm and alerting the security guards or police.
c. **Corrective Controls** are executed after the event and are intended to limit the extent of any damage caused by the incident. In other word they exist to mitigate or lessen the effects of the threat being manifested, e.g. by recovering the organization to normal working status as efficiently as possible.

Computer security is often divided into three distinct master categories, commonly referred to as controls:

a. Physical
b. Technical
c. Administrative

These three broad categories define the main objectives of proper security implementation. Within these controls are sub-categories that further detail the controls and how to implement them.

Physical Controls

The Physical control is the implementation of security measures in a defined structure used to deter or prevent unauthorized access to sensitive material. Examples of physical controls are:

i. Closed-circuit surveillance cameras
ii. Motion or thermal alarm systems
iii. Security guards
iv. Picture IDs
v. Locked and dead-bolted steel doors

Technical Controls

The Technical control uses technology as a basis for controlling the access and usage of sensitive data throughout a physical structure and over a network. Technical controls are far- reaching in scope and encompass such technologies as:

vi. Encryption
vii. Smart cards
viii. Network authentication
ix. User Access control
x. File integrity auditing software

Administrative Controls

Administrative controls define the human factors of security. It involves all levels of personnel within an organization and determines which users have access to what resources and information by such means as:

xi. Training and awareness

xii. Disaster preparedness and recovery plans

xiii. Personnel recruitment and separation strategies

xiv. Personnel registration and accounting

Physical security, protection on the inside

Most experts would agree that all security starts with physical security. Controlling physical access to machines and network attach points is perhaps more critical than any other aspect of security. Any type of physical access to an internal site creates a major exposure of the site. Secure files, passwords, certificates and all sorts of other data can usually be obtained if

physical access is possible. Fortunately there are all sorts of access control devices and secure cabinets that can help with this problem.

Partitioning and protecting network boundaries with firewalls

Besides the basic physical security of a site, the next most important aspect is controlling digital access into and out of the organization's network. In most cases this means controlling the points of connectivity to the outside world, typically the Internet. Almost every medium and large-scale company has a presence on the Internet and has an organizational network connected to it. In fact there is a large increase in the number of smaller companies and homes getting full time Internet connectivity. Partitioning the boundary between the outside Internet and the internal intranet is a critical security piece. Sometimes the inside is referred to as the "trusted" side and the external Internet as the "un-trusted" side. As a generality this is all right, however, as will be described, this is not specific enough.

A firewall is a mechanism by which a controlled barrier is used to control network traffic into AND out of an organizational intranet. Firewalls are basically application specific routers. They run on dedicated embedded systems such as an internet appliance or they can be software programs running on a general server platform. In most cases these systems will have two network interfaces, one for the external network such as the Internet and one for the internal intranet side. The firewall process can tightly control what is allowed to traverse from one side to the other. Firewalls can range from being fairly simple to very complex. As with most aspects of security, deciding what type of firewall to use will depend upon factors such as traffic levels, services needing protection and the complexity of rules required. The greater the number of services that must be able to traverse the firewall the more complex the requirement becomes. The difficulty for firewalls is distinguishing between legitimate and illegitimate traffic.

What do firewalls protect against and what protection do they not provide? Firewalls are like a lot of things; if configured correctly they can be a reasonable form of protection from external threats including some denial of service (DOS) attacks. If not configured correctly they can be major security holes in an organization. The most basic protection a firewall provides is the ability to block network traffic to certain destinations. This includes both IP addresses and particular network service ports. A site that wishes to provide external access to a web server can restrict all traffic to port 80 (the standard http port). Usually this restriction will only be applied for traffic originating from the un-trusted side. Traffic from the trusted side is not restricted. All other traffic such as mail traffic, ftp, snmp, etc. would not be allowed across the firewall and into the intranet.

An even simpler case is a firewall often used by people with home or small business cable or DSL routers. Typically these firewalls are setup to restrict ALL external access and only allow services originating from the inside. A careful reader might realize that in neither of these cases is the firewall actually blocking all traffic from the outside. If that were the case how could one surf the web and retrieve web pages? What the firewall is doing is restricting connection requests from the outside. In the first case all connection requests from the inside are passed to the outside as well as all subsequent data transfer on that connection. From the exterior, only a connection request to the web server is allowed to complete and pass data, all others are blocked. The second case is more stringent as connections can only be made from the interior to the exterior.

More complex firewall rules can utilize what is called "stateful inspection" techniques. This approach adds to the basic port blocking approach by looking at traffic behaviours and sequences to detect spoof attacks and denial of service attacks. The more complex the rules, the greater the computing power of the firewall required.

One problem most organizations face is how to enable legitimate access to "public" services such as web, ftp and e-mail while maintaining tight security of the intranet. The typical approach is to form what is known as a DMZ (demilitarized zone), a euphemism from the cold war applied to the network. In this architecture there are two firewalls: one between the external network and the DMZ, and another between the DMZ and the internal network. All public servers are placed in the DMZ. With this setup, it is possible to have firewall rules which allow public access to the public servers but the interior firewall can restrict all incoming connections. By having the DMZ, the public servers are still provided more protection than if they were just placed outside a single firewall site.

3.5 SU R NCC UU CEA RESU

What is an access control? We can say it's a way to manage access to enterprise resources. It we go with the word definition, access control is a mechanism to control the flow of information between subject and object where subject is always as active entity while object is a passive entity. In its broadest meaning, access control is a three-step process that includes identification, authentication. In this white paper, the term authentication is generally used to represent both identification and authentication, and access control is used for authorization.

The section discusses the importance of selecting an access control model that fits with your security needs to provide a lower total cost of ownership and enable strong identification. It also discusses the various authentication solutions and weights their need to your organization.

Why Access Controls are required

As the business of an enterprise increases, so does the demand for access control since it is the first line of defense to protect the organization's resources. When you try to access a certain resource and you are asked to provide your identification – that is access control. Access control solutions provide protection, integrity, availability and auditing capability to the organisation.

What are Access Control Models

Access control model is a framework that dictates access control using various access-control technologies. There are standard access control models which are highly domain and implementation independent. Each access control model has its own merits and demerits, andthe specific business objectives they serve depend on the organization's need, culture, nature of business, etc. We will discuss these models and examine their fitness with respect to an organization's security policy and business goals.

- Discretionary Access Control (DAC)
- Mandatory Access Control (MAC)
- Role Based Access Control (RBAC)

Discretionary Access Control (DAC)

Discretionary Access Control is based on ownership and delegation. In a DAC Model, access is governed by the access rights granted to the user groups. An organisation/administrator/creator can identify a set of operations and assign them to an object and to set of users and to a set of users (belonging to user group).

The DAC model is flexible but complex. It creates a paradox in some complex situations. For example, A is owner of resource R of organisation O and he has delegated permission P1and P2 to B who, in turn, has delegated permission P1 to C. Now, it A chooses to revoke permission to B what will happen to the permission that B granted to C?

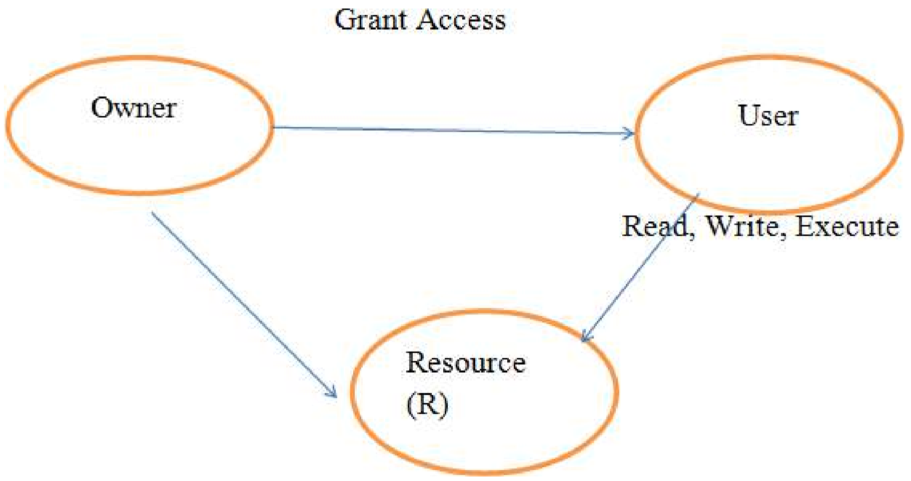


Figure 3: Discretionary Access Control

While the model above is complex, it is still flexible enough to handle various access control needs, and therefore is used in various network management applications.

Mandatory Access Control (MAC)

In MAC, the data owner has limited freedom to decide on access control. Information is classified into different categories and each category is assigned a particular security level. For example, resource R is a very confidential resource to the organisation and so has been assigned a "Very Confidential" security level. When a user, who has been assigned security level "Confidential", tries to access this resource, he is denied access because the security level assigned to him doesn't match. This model is appropriate when securing confidential of data is critical, as in—for example—military operation systems.

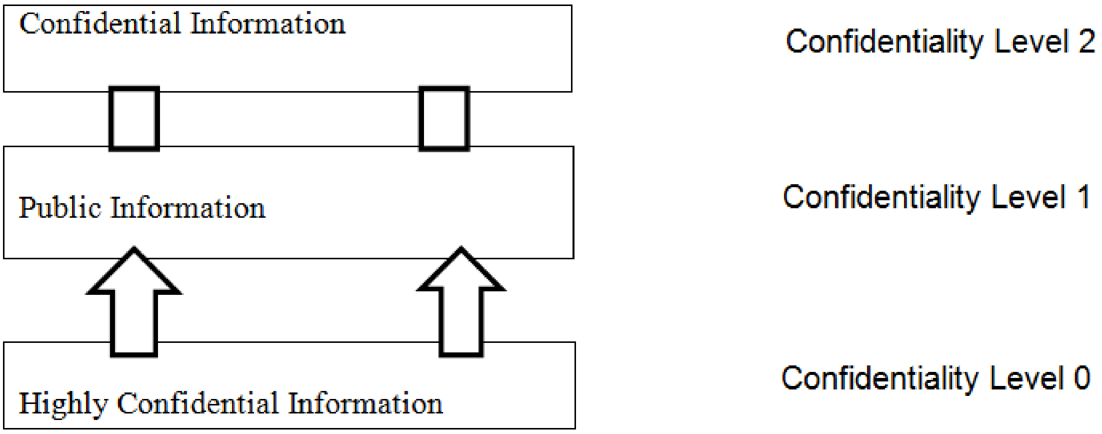


Figure 4: Mandatory Access Control

Role Based Access Control (RBAC)

RBAC is a widely used- and dominant- access control model, and most access control security products available in the market today are based on this model because its objectives are architectural. Entrust Get Access is one such product. The model allows access to a resource, based on the role the user holds in the organization. It is based on the concept of "separation of duties". The privileges to the particular role are decided and thereafter mapped to the user. If the environment does not require a high level of security, the choices are usually discretionary and role based. The discretionary model gives data owners the ability to allow users to access resources, enabling choices to be made with full knowledge of what it entails. If the organization has a high turnover rate, the role based model is more appropriate. If the environment requires higher level of security and it is desired that only administrator should grant the access control, then MAC is the best choice.

Authentication

Authentication or identification is the first step in any access solution. It is the process of identifying the user to verify whether he/she is what he/she claims to be. Normally, identification is done with the help of information that is known to everyone (i.e., user name or user ID) and some personal information known only to the subject (i.e. password). Faced with the threat of identity theft and increasing consequences associated with failing to secure information, enterprises are increasingly looking for stronger forms of authentication to enhance their overall security capabilities. At the same time, enterprises and governments need to take into account other important considerations such as usability, total cost of deployment and maintenance, and integration with existing security solution offerings. Usernames and passwords are the most common authentication techniques. But most organizations do not depend on user name authentication alone since username and passwords are an authentication solution for low-value transactions and for accessing non- sensitive information over the network. Also, experience has shown that usernames and passwords provide relatively weak authentication because they can often be guessed or stolen. They are often difficult to deploy because each application may implement its own

scheme, adding to both development cost and user complexity. Also, it is very difficult to maintain and reset the password.. Determining the appropriate level of authentication that meets your budget requirements is essential when implementing your secure identity management solution. It is very crucial to identify the appropriate authentication technique depending upon the nature of the business and sensitivity of the information. One has to consider various authentication methods and their pros and cons. The means of authentication are often discussed in terms of "factors" of proof, such as:

- Something your know to prove your identity (e.g.a PIN)
- Something you have to prove your identity (e.g. a smart card)
- Something you have to prove your identity (e.g. a fingerprint)

A good authentication technique contains at least two of the above methods. In a client server environment, strong authentication is a combination of server and client authentication:

- Server authentication is when the server proves its identity to the client.
- Client authentications are when clients prove their identity to the server.

User password Authentication

It is the most common form of providing identification. When user accesses the resource, access control framework asks for the user name password provided to the user. The credentials are validat3d against the one stored in the system's repository.

Windows user based authentication

Usually, organizations have a list of users stored in the windows active directory. Access control framework should be able to provide authentication for the user of the Primary Domain Controller (PDC).

Directory based authentication

With the rising volume of business over the web, millions of user's often try to access the resource simultaneously. In such a scenario, the authentication framework should be able to provide for faster authentication. One such technique is Directory Based Authentication where user credentials are validated against the one which is stored in the LADAP Directory.

Certificate based authentication

This is probably one of the strongest authentication techniques where the user is asked to provide his/her digital ID. This digital ID, known as digital certificate, is validated against the trusted authority that issued the digital ID. There are various other parameters that ate checked to ensure the identification of the user.

Smart card based authentication

This is also used as a second factor authentication. Smart cards are small devices containing co-processors to process cryptographic data.

Biometrics

This is the strongest authentication. Known as third factor authentication, it is based on something the user is. It works after the users have provided something the know (User name password) and something they own (either a gird or token) or something they are (retina- scan, thumbprint or thermal scan). It is required in cases where data is top confidential, such as in Military/Defense.

Grid based Authentication

This is used as a second factor authentication. It authenticates the user based on something he knows (User name password authentication) and then asks for something he owns (gird card information). Entrust Identity Guard provides such an authentication.

Knowledge-based authentication

One of the simplest mechanisms for gaining additional confidence in a user's identity is to challenge the user to provide information that an attacker is unlikely to be able to provide.

Based on “shared secrets”, this allows for the organization to question the user, when appropriate, to confirm information that is already known about the user through a registration process, or from previous transactions.

Machine Authentication

Machine authentication provides validation of the user’s computer in a way that secures against a variety of threats in a zero touch fashion, reducing user impact. This is an especially effective method of user authentication where users typically access their accounts from a regular set of machines, allowing for stronger authentication to be performed without any significant impact on the user experience.

One time Password (OTP)

A one-time password is dynamically generated and it is valid only for once. The advantage of one time password is that if an intruder hacks it, he cannot reuse it. These are two types of OTP token generators: synchronous and asynchronous. A synchronous token device synchronizes with the authentication service by using time or an event as the core piece of the authentication process. A token device, which is using as asynchronous token generating method, uses a challenge response scheme to authenticate the user.

Access Control Framework (ACF)

The access control framework (ACF), presented in this paper, is like an umbrella that covers both authentication and authorization. Whenever the user accesses any enterprise resource, ACF can come up with one or more authentication techniques depending on the need of the enterprise. Once the authentication is done, ACF can authorize the request, following any model, depending on the need of the organization.

Access Control Techniques and Technologies

Once an organization decides on the type of access control model to be employed, the next step would be to decide on the techniques and technology to be used. Here are some techniques and technologies:

Rule Based Access control

Rule based access control is based on rules defined on the object, as defined by the administrator who decides on the operations that can be performed by subject. A rule can be as simple as defining the day of the week on which the resource can be accessible.

Menu Based Access Control

In a menu based control, the user interface given to the user controls the operations that can be performed on the object, i.e., If A and B operations can be performed on object O, then the user interface pertaining to A and B options is enabled and the rest of the user interface is disabled.

Access Control List

Access control list is the list of subjects that are authorized to access a particular object. It also defines the level of authorization.

Content Based Access Control

In content based access control (CBAC), the access to the object is determined by the content within the object. For example, a manager can access the payroll database but only for employees reporting to him.

Access control Markup Language (XACML)

XACML is the access control markup language that is used to express the rules that are necessary for authentication and authorization. The vocabulary to express these rules is given by the access control markup language. These rules are used to make decisions regarding the authentication. eXtensible Access Control Markup Language – or XAVML – provides a Mechanism to create policies and rules for controlling access to information.

A typical access control and authorization scenario includes three main entities - a subject, a resource, and an action - - and their attributes. A subject makes a request for permission to perform an action on a resource. For example, in the access request, "Allow the Sys-admin to create files in the root folder of the production server" the subject is the "Sys-admin", the target resource is the "root folder of the production server", and the action is "create files".

Security Assertion Markup Language (SAML)

SAML is an XML standard for exchanging authentication and authorization data between security domains, that is, between an identify provider (a producer of assertions) and a service provider (a consumer of assertions). SAML is a product of the OASIS Security Services Technical Committee.

The single most important problem that SAML is trying to solve is the Web Browser single sign-on (SSO) problem. Single sign-on solutions are abundant at the intranet level (using cookies, for example) but extending these solutions beyond the intranet has been problematic and has led to the proliferation of non-interoperable proprietary technologies. SAML has become the definitive standard underlying many web single sign-on solutions in the enterprise problem space.

The whole thrust of access control is to restrict unauthorized users from accessing organization resources. The authentication techniques and access controls described in this white paper can be chosen based on an organization's need. The authentication and access control framework should be flexible enough to serve all the authentication techniques and future evolution in the area such as Biometrics. The access control framework should be able to handle an organization's authentication and authorization (access control) needs. Entrust Get Access and Identify Guard is the products with such features.

3.6 TRAINING AND AWARENESS

One of the greatest threats to information security could actually come from within your company or organization. Inside 'attacks' have been noted to be some of the most dangerous since these people are already quite familiar with the infrastructure. It is not always disgruntled workers and corporate spies who are a threat. Often, it is the non-malicious, uninformed employee.

The focus will be on uninformed users who can do harm to your network by visiting websites infected with malware, responding to phishing e-mails, storing their login information in an unsecured location, or even giving out sensitive information over the phone when exposed to social engineering.

One of the best ways to make sure company employees will not make costly errors in regard to information security is to institute company-wide security-awareness training initiatives that include, but are not limited to classroom style training sessions, security awareness website(s), helpful hints via e-mail, or even posters. These methods can help ensure employees have a solid understanding of company security policy, procedure and best practices. Some of the more important items to cover in your security awareness training are your organization's security policy, data classification and handling, workspace and desktop security, wireless networks, password security, phishing, hoaxes, malware, file sharing and copyright.

3.6.1 Types of Training

Organizations are starting to realize there really is a need for security awareness training. According to a study conducted, the following statistics revealed a rather startling necessity:

- "One in five workers (21%) let family and friends use company laptops and PCs to access the Internet".
- "More than half (51%) connect their own devices or gadgets to their work PC... a quarter of who do so every day".
- "One in ten confessed to downloading content at work they should not".
- "Two thirds (62%) admitted they have a very limited knowledge of IT Security".
- "More than half (51%) had no idea how to update the anti-virus protection on their company PC".
- "Five percent say they have accessed areas of their IT system they should not have".

Security awareness training can be performed in a variety of ways that can be utilized alone or in conjunction with each other. Those mediums can consist of a more thorough classroom style training, creation of a security-awareness website, pushing helpful hints onto computers when they start up and/or e-mailing helpful hints on a weekly or monthly basis, and utilizing visual aids like posters.

Classroom-Style Training

Utilizing a classroom setting for security-awareness training can offer the benefit of lecture- based and interactive learning as well as the availability of someone to answer questions in real time. There can also be a Q&A period after the materials are presented as well as contact information distributed for questions that might pop up afterward.

Some companies offer both live and web-based training and utilize a variety of methods such as role-playing and simulation games so the interaction is more two-way than one-way. Other companies offer videos, web-based training, and live trainers. The method you use is by no means limited.

This type of training can vary in the amount of time it can take. The security awareness training I have implemented at Washtenaw Community College takes about two hours, but it has no real interactivity such as role-playing or simulations; just PowerPoint and Q&A. Training time can depend on the effectiveness and the extent of the material discussed. Training sessions could possibly take a full day if need be.

Security Awareness Website

Another way of implementing a security awareness program is through the creation of a security awareness website. This website could consist of different sections with the different areas that need to be covered (e.g. Malware, hoaxes, file sharing and copyright, etc). The University of Tennessee implemented a very impressive security awareness website complete with videos, examples, and helpful external links.

Another implementation of the security awareness website could be a self-paced tutorial where users can log in and go through it, taking mini quizzes at the end of each section to make sure the material is actually being read and absorbed. Utilizing logins can also be a means of keeping track of who has (and more importantly who has not) taken the training. An FAQ section could be implemented as well as contact information for users to ask questions that are not addressed in the FAQ.

3.6.2 Helpful Hints

Utilizing helpful hints and tips is more of a supplement to the training, be it via classroom style or online, and should not be used as a means of security awareness training on its own. Helpful hints can consist of tips and reminders that are pushed to user screens when they log in. These tips and reminders can consist of key points emphasized in the training (e.g. “Never keep your password in a place that can be accessed or viewed by anyone besides yourself”). Reminders can be as simple as reminding someone to change their password or run their virus scan.

Visual Aids

Visual aids are another item that should not be used as the lone source of security awareness training, but more as a supplement. The University of Michigan recently created a series of catchy password security posters that compare passwords to underwear. One says to change them often, another says to not leave passwords lying around, and another one says to not share them with friends.

Promotions

Security tips can appear on flyers distributed across the user base and one could even go so far as to hand out pencils and/or key chains with a catchy security-related phrase or reminder (e.g. “Unexpected attachments can mean unexpected chaos: Please do not open them”). Now that we have addressed possible methods in implementing security awareness training, what should be covered in the training will be addressed.

Training Topics

Topics addressed by the security awareness training should consist of a combination of existing organizational policies and procedures (how they tie in with each aspect, if they do), physical security, desktop security, password security, phishing, hoaxes, malware (viruses, worms, Trojans, spyware, and adware), and copyright with regard to file sharing. These topics will help employees understand why security awareness is important and guide them in knowing how to prevent incidents from happening and what to do if one occurs.

Physical Security

When addressing physical security, locking your doors and desk/file cabinet drawers should be the main focus. A helpful item to include could be the crime statistics, more specifically thefts, from the organization. Another item to lightly touch upon (but go into greater detail in Desktop Security) is the fact that if a potential attacker has access to a user's computer, they could install a key logger or actually get into a machine that has not been locked.

Desktop Security

The desktop security section should go into detail as to why it is important to either have a password-protected screen saver or, even better, to get into the habit of locking computers when users walk away from them. A screensaver timeout should be utilized so if a user walks away from their computer, the password-protected screensaver would come up. Personally, I have mine set to 5 minutes, but upon doing a Google search regarding typical screensaver timeouts, the average response was 10 minutes. This information can and should be supplemented with information on how to do this. Tactics a potential attacker could utilize (e.g. Shoulder surfing, key loggers, etc) also need to be addressed. The pill of having to take extra measures to make sure your desktop is secured can be swallowed more easily if users understand WHY they should be taking them.

Another item that could be addressed is to make sure users understand that it is important that they shut down their computers at the end of the day. Sometimes this allows for valuable updates to be applied and doing your own part for a greener environment. If somehow a potential attacker gains access to a computer that is turned off, they will be less likely to utilize it than one that is already turned on and unlocked.

Wireless Networks and Security

The wireless networks and security section should address the unsecure nature of wireless networks as well as tips and tricks to exercise caution and harden laptops against the dangers of 'sniffing'. Emphasis should also be placed on not storing any kind of sensitive information on laptops that will be accessing a wireless network. Another area that should be covered is the importance of firewalls. Windows Firewalls by themselves are not enough. Most times companies will provide a purchased firewall on company-supplied laptops and computers but personal laptops that may utilize the company wireless network need to have a firewall on them. For small office environments as well as those who remotely access their workstation from home, it is always helpful to provide information on free firewall options like ZoneAlarm and Comodo as well as relatively inexpensive firewall options like McAfee and Norton. The free firewalls are more for the personal user, though and not for commercial use. It may also be a benefit to the training to compare the price of a laptop to the price of a breach.

Password Security

The password security section should include what constitutes a strong, secure password or passphrase, with an emphasis on passphrases since they are harder to guess and to crack. This section should also outline the minimum password requirements of the organization.

Sharing passwords as well as leaving them out where anyone but the user could access them should be strongly discouraged. Making this part of organization-wide policy could be very helpful in this arena. If this is incorporated into policy, this should be addressed in the training. Users need to be aware a policy is in place and general "rules of thumb" to make sure these policies are followed. Statistics could also be a good supplement. For example, a delegated individual could go around to all of the offices and see if they can uncover any unsecured passwords. They could even take this a step further and see how many computers are left on as well as without password-protected screensavers. No specific individuals would be singled out; just a number of instances out of the total number of computers would suffice.

Helpful hints and rules of thumb should also be a part of this section. For example, passwords should not contain the username or any part of the user's full name. Passwords also should not be based on personal information such as a spouse name, favourite team, or pet. Another important point is to stress that the default password given to users should always be changed immediately. Instructions on how to change passwords should also be included.

To round out the password security section, it can be very beneficial to define what constitutes a poor choice of password as well as a listing of the most common passwords used.

Phishing

When discussing phishing, the term as well as the purpose should always be defined. Examples are key to this portion of security awareness training. Things to avoid (e.g. clicking on links provided in e-mail, submitting banking and password information via email, etc.) should be highly emphasized so people know what to look for. It could also be beneficial to have users take a Phishing IQ Test. This way the bits and pieces that can identify a phishing e-mail can be explained and displayed. Another item that should be addressed is how to actually fight phishing attacks. A couple of web sites actually encourage the reporting and tracking of phishing web sites and e-mails: PhishTank (www.phishtank.com) and The Anti- Abuse Project (www.antiabuse.org), which address these issues.

Hoaxes

Hoaxes should be addressed in the training because a lot of time and resources can be spent reading and forwarding hoax emails. The types of hoaxes as well as examples should be the meat of this section. Using familiar hoaxes is the best option so it will be easier to grasp. It could also be beneficial to compare hoaxes to viruses in that they are spread by continually forwarding them. The dangers of hoaxes should also be addressed because some hoaxes warn of a virus and tell users to delete valid and sometimes important system files.

Preventing the spread of hoaxes should also be covered. Hoaxes can be prevented by checking a number of hoax sites on the Web and following a few rules of thumb. It is important to point out that if something sounds too good to be true, it probably is and if something seems suspicious it can be checked on one of the hoax sites

Malware

When addressing malware, it should always be defined and then broken down into its categories: viruses, worms, Trojans, spyware, and adware. After each category is broken down, address how they end up on systems.

Viruses

Start out by outlining what makes a virus a virus. It is important for users to be able to identify a potential virus when they see one or to identify characteristics of a virus that has already infiltrated the user’s system. What a virus is capable of is also something that should supplement the defining of what makes a virus what it is.

Defining what a virus is and how to identify one must be complemented with the important of antivirus software. Most organizations will have this installed on all organization-wide computers, but this might not be installed on laptops used by employees. Users also need to learn the importance of not only performing regular scans of their computers, but also of any file they download from a web site, e-mail, or thumb drive.

Another important tip to include is how vital it is to keep systems and applications up-to-date. Never assume that a system or application is always going to update itself. Users should proactively see if the systems and applications they are using need updated.

Finally, it is important to let users know what to do if their system does become infected. Make sure not to incite a sense of panic that would steer employees toward hiding the infection until it has gotten out of control or their machine is beyond repair. The main

procedure to address is what to do if and when a virus infects a work machine, since it would differ considerably to what to do at home.

When your work machine becomes infected, do not do anything to the computer aside from performing a scan with the anti-virus software on the machine. Phone the I.T. Department of your business to come evaluate your machine and hopefully get rid of the virus.

If your machine at home (especially if you work from home) becomes infected, it is important to follow the following steps outlined on Viruslist.com:

- Do not panic
- Disconnect from the Internet and any Local Area Network it may be connected to.
- If computer cannot boot, try starting in Safe Mode or boot from the Windows boot disk.
- Back up any important data you cannot afford to lose to an external drive (scan the file with your anti-virus software first) (floppy, CD, jump drive, etc).
- If you do not have anti-virus software installed (which SHOULD not be the case), install it and then update it.
- Perform a full scan of your system.

Worms

The worms section can be handled much the same way the virus section is handled: Definition, how to spot, what it is capable of, how to prevent, what to do if one invades the system.

Trojans

Like the previous 2 sections, the Trojans portion should define what they are, what they can do, what can be done to prevent them, and what to do in the event of one making it onto the system. One item that should be emphasized is that Trojans are different from viruses and why they are two different things.

Spyware and Adware

Again, spyware and adware should be defined, what they can do should be outlined, prevention tips and tricks, and then what to do if it is found on the system. Spyware and adware identification and removal programs should also be addressed, most of which are free (e.g. Ad Aware, Spy Sweeper, etc).

3.7 SECURITY CONTROL DESIGN

Security controls are safeguards or countermeasures to avoid, detect, counteract, or minimize security risks to physical property, information, computer systems, or other assets. Controls help to reduce the risk of damage or loss by stopping, deterring, or slowing down an attack against an asset. To help review or design security controls, they can be classified by several criteria. The salient criteria are listed below:

a. Categorizing according to the time that they act, relative to a security incident.

   i. **Preventive controls** are intended to prevent an incident from occurring e.g. by locking out unauthorized intruders.

   ii. **Detective controls** are intended to identify and characterize an incident in progress
   e.g. by sounding the intruder alarm and alerting the security guards or police.

   iii. **Corrective controls** are intended to limit the extent of any damage caused by the incident e.g. by recovering the organization to normal working status as efficiently as possible.

b. Security controls can also be categorized according to their nature.

i. **Physical controls e.g.** fences, doors, locks and fire extinguishers

ii. **Procedural controls** e.g. incident response processes, management oversight, security awareness and training

iii. **Technical controls** e.g. user authentication (login) and logical access controls, antivirus software, firewalls

iv. **Legal and regulatory or compliance controls** e.g. privacy laws, policies and clauses

In other words Security controls are technical or administrative safeguards to minimize loss or unavailability due to threats acting on their matching vulnerability, i.e., security risk. However, this unit restricts our scope to Technical Security Controls.

3.8 TECHNICAL SECURITY CONTROLS

Technical security involves the use of safeguards incorporated in computer hardware, operations or applications software, communications hardware and software, and related devices. Technical controls are also referred to as logical controls.

Preventive Technical Controls

Preventive technical controls are used to prevent unauthorized personnel or programs from gaining remote access to computing resources. Examples of these controls include:

- Access control software.
- Antivirus software.
- Library control systems.
- Passwords.
- Smart cards.
- Encryption.
- Dial-up access control and callback systems.

Access Control Software

The purpose of access control software is to control sharing of data and programs between users. In many computer systems, access to data and programs is implemented by access control lists that designate which users are allowed access. Access control software provides the ability to control access to the system by establishing that only registered users with an authorized log-on ID and password can gain access to the computer system. After access to the system has been granted, the next step is to control access to the data and programs residing in the system. The data or program owner can establish rules that designate who is authorized to use the data or program.

Antivirus Software

Viruses have reached epidemic proportions throughout the microcomputing world and can cause processing disruptions and loss of data as well as significant loss of productivity while cleanup is conducted. In addition, new viruses are emerging at an ever-increasing rate — currently about one every 48 hours. It is recommended that antivirus software be installed on all microcomputers to detect, identify, isolate, and eradicate viruses. This software must be updated frequently to help fight new viruses. In addition, to help ensure that viruses are intercepted as early as possible, antivirus software should be kept active on a system, not used intermittently at the discretion of users.

Library Control Systems

These systems require that all changes to production programs be implemented by library control personnel instead of the programmers who created the changes. This practice ensures separation of duties, which helps prevent unauthorized changes to production programs.

Passwords

Passwords are used to verify that the user of an ID is the owner of the ID. The ID-password combination is unique to each user and therefore provides a means of holding users accountable for their activity on the system. Fixed passwords that are used for a defined period of time are often easy for hackers to compromise; therefore, great care must be exercised to ensure that these passwords do not appear in any dictionary. Fixed passwords are often used to control access to specific data bases. In this use, however, all persons who have authorized access to the data base use the same password; therefore, no accountability can be achieved. Currently, dynamic or one-time passwords, which are different for each log-on, are preferred over fixed passwords. Dynamic passwords are created by a token that is programmed to generate passwords randomly.

Smart Cards

Smart cards are usually about the size of a credit card and contain a chip with logic functions and information that can be read at a remote terminal to identify a specific user's privileges. Smart cards now carry prerecorded, usually encrypted access control information that is compared with data that the user provides (e.g., a personal ID number or biometric data) to verify authorization to access the computer or network.

Encryption

Encryption is defined as the transformation of plaintext (i.e., readable data) into ciphertext (i.e., unreadable data) by cryptographic techniques. Encryption is currently considered to be the only sure way of protecting data from disclosure during network transmissions. Encryption can be implemented with either hardware or software. Software-based encryption is the least expensive method and is suitable for applications involving low-volume transmissions; the use of software for large volumes of data results in an unacceptable increase in processing costs. Because there is no overhead associated with hardware encryption, this method is preferred when large volumes of data are involved.

Dial-Up Access Control and Callback Systems

Dial-up access to a computer system increases the risk of intrusion by hackers. In networks that contain personal computers or are connected to other networks, it is difficult to determine whether dial-up access is available or not because of the ease with which a modem can be added to a personal computer to turn it into a dial-up access point. Known dial-up access points should be controlled so that only authorized dial-up users can get through.

Currently, the best dial-up access controls use a microcomputer to intercept calls, verify the identity of the caller (using a dynamic password mechanism), and switch the user to authorized computing resources as requested. Previously, call-back systems intercepted dial- up callers, verified their authorization and called them back at their registered number, which at first proved effective; however, sophisticated hackers have learned how to defeat this control using call-forwarding techniques.

3.8.1 Detective Technical Controls

Detective technical controls warn personnel of violations or attempted violations of preventive technical controls. Examples of these include audit trails and intrusion detection expert systems, which are discussed in the following sections.

Audit Trails

An audit trail is a record of system activities that enables the reconstruction and examination of the sequence of events of a transaction, from its inception to output of final results. Violation reports present significant, security-oriented events that may indicate either actual or attempted policy transgressions reflected in the audit trail. Violation reports should be frequently and regularly reviewed by security officers and data base owners to identify and investigate successful or unsuccessful unauthorized accesses.

Intrusion Detection Systems

These expert systems track users (on the basis of their personal profiles) while they are using the system to determine whether their current activities are consistent with an established norm. If not, the user's session can be terminated or a security officer can be called to investigate. Intrusion detection can be especially effective in cases in which intruders are pretending to be authorized users or when authorized users are involved in unauthorized activities.

3.8.2 Corrective Technical Controls

Corrective controls exist to mitigate or lessen the effects of the threat being manifested. Examples of these include OS upgrades, Backup Data restoral and Vulnerability Mitigation which are discussed in the following sections.

OS Upgrade

Keep any and all original software media from which to restore the system. The latest upgrades of operating system should be maintained for system integrity.

Backup Data restoral

A good backup methodology should be in place to back up data. Commercial third party products may be used for the same. A back up will allow an organisation to restore critical data once a system has been rebuilt.

Vulnerability Mitigation

To mitigate the damage caused due to the vulnerability in the security system, the periodic testing of the backups for data integrity of the archived data.

3.9 PROTECTION FROM MALICIOUS ATTACKS

Although the technology behind the development of protection methods continues to improve in leaps and bounds, the threats against computers and the data, they contain, still remain. Keeping your defensive mechanism regularly updated is a critical aspect of your online security. Always keep in mind that hackers don't stick to the same threat tactics. They are also constantly looking for ways to bypass and counter the protective programs that people install in their computers. Whatever operating system you use, always see to it that you have at least two anti-virus software programs installed. The logic is simple - if the first line of defense didn't catch and contain the threat then the second program should do the trick. In fact, a lot of people make use of more than two security programs to protect their computers and data from malicious attacks.

In trying to find the best protection for your computer, there are several factors that you need to consider. For instance, what kind of data do you store in your computer? And what's the scope of this data? We need to understand that individuals with malicious intent usually make use of different types of attacks when sabotaging different types of data. Consequently, the methodology to be chosen should have the capability to protect whatever type of data you store in your computer. Fortunately, there's no shortage of software companies that focus on developing security programs. These defence systems are also constantly updated to ensure that they can hinder new threats.

Here are some practical tips on how you can efficiently prevent malicious code from wreaking havoc to your computer data.

a. Choose reliable anti-virus programs
b. Install real-time anti-spyware protection
c. Keep anti-malware applications current
d. Disable auto-run
e. Perform daily scans
f. Block suspicious files sent to your organization via email
g. Surf smart
h. Use a hardware-based firewall
i. Deploy DNS protection

Discussing each of the methods we can be more aware of the attacks and the mitigating techniques that can be adopted for the safety of our systems.

1. **Choose reliable anti-virus programs:** Many computer users believe free antivirus applications, such as those included with an Internet service provider's bundled service offering, are sufficient to protect a computer from virus or spyware infection. However, such free anti-malware programs typically don't provide adequate protection from the ever-growing list of threats. Instead, all users should install professional, business-grade antivirus software on their PCs. Pro-grade antivirus programs update more frequently throughout the day (thereby providing timely protection against fast-emerging vulnerabilities), protect against a wider range of threats (such as rootkits), and enable additional protective features (such as custom scans).
2. **Install real-time anti-spyware protection:** Many computer users mistakenly believe that a single antivirus program with integrated spyware protection provides sufficient safeguards from adware and spyware. Others think free anti-spyware applications, combined with an antivirus utility, deliver capable protection from the skyrocketing number of spyware threats. Unfortunately, that's just not the case. Most free anti- spyware programs do not provide real-time, or active, protection from adware, Trojan, and other spyware infections. While many free programs can detect spyware threats once they've infected a system, typically professional (or fully paid and licensed) anti- spyware programs are required to prevent infections and fully remove those infections already present.
3. **Keep anti-malware applications current:** Antivirus and anti-spyware programs require regular signature and database updates. Without these critical updates, anti-malware programs are unable to protect systems from the latest threats. Statistics reveal that a lot of serious computer threats are secretive and fast-moving. Many of these infections are short-lived, but they're estimated to infect as many as 100,000 to 300,000 new Web sites a day. Computer users must keep their antivirus and anti-

spyware applications up to date. All users must take measures to prevent license

expiration, thereby ensuring that their anti-malware programs stay current and continue providing protection against the most recent threats.

4. **Perform daily scans:** Occasionally, virus and spyware threats escape a system's active protective engines and infect a system. The sheer number and volume of potential and new threats make it inevitable that particularly inventive infections will outsmart security software. In other cases, users may inadvertently instruct anti-malware software to allow a virus or spyware program to run. Regardless of the infection source, enabling complete, daily scans of a system's entire hard drive adds another layer of protection. These daily scans can be invaluable in detecting, isolating, and removing infections that initially escape security software's attention.
5. **Disable auto run:** Many viruses work by attaching themselves to a drive and automatically installing themselves on any other media connected to the system. As a result, connecting any network drives, external hard disks, or even thumb drives to a system can result in the automatic propagation of such threats. Computer users can disable the autorun feature by following the providers recommendations, which differ by operating system.
6. **Block suspicious files sent to your organization via email:** It's a mantra most users have heard repeatedly: Don't click on email links or attachments. Yet users frequently fail to heed the warning. Whether distracted, trustful of friends or colleagues they know, or simply fooled by a crafty email message, many users forget to be wary of links and attachments included within email messages, regardless of the source. Simply clicking on an email link or attachment can, within minutes, corrupt Windows, infect other machines, and destroy critical data. Users should never click on email attachments without at least first scanning them for viruses using a business- class anti-malware application. As for clicking on links, users should access Web sites by opening a browser and manually navigating to the sites in question.

7. **Surf smart:** Many business-class anti-malware applications include browser plug-ins that help protect against drive-by infections, phishing attacks (in which pages purport to serve one function when in fact they try to steal personal, financial, or other sensitive information), and similar exploits. Still others provide "link protection," in which Web links are checked against databases of known-bad pages. Whenever possible, these preventive features should be deployed and enabled. Unless the plug- ins interferes with normal Web browsing, users should leave them enabled. The same is true for automatic pop-up blockers, included in browser toolbars. Regardless, users should never enter user account, personal, financial, or other sensitive information on any Web page at which they haven't manually arrived. They should instead open a Web browser, enter the address of the page they need to reach, and enter their information that way, instead of clicking on a hyperlink and assuming the link has directed them to the proper URL. Hyperlinks contained within an e-mail message often redirect users to fraudulent, fake, or unauthorized Web sites. By entering Web addresses manually, users can help ensure that they arrive at the actual page they intend which is also not fool proof.
8. **Use a hardware-based firewall:** Technology professionals and others argue the benefits of software- versus hardware-based firewalls. Often, users encounter trouble

   trying to share printers, access network resources, and perform other tasks when deploying third-party software-based firewalls. As a result, users in many cases simply disable firewalls altogether. But a reliable firewall is indispensable, as it protects computers from a wide variety of exploits, malicious network traffic, viruses, worms, and other vulnerabilities. Unfortunately, by itself, the software-based firewall included with operating system isn't sufficient to protect systems from the myriad robotic attacks affecting all Internet-connected systems. For this reason, all systems connected to the Internet should be secured behind a capable hardware-based firewall.
9. **Deploy DNS protection:** Internet access introduces a

wide variety of security risks. Among the most disconcerting may be drive-by infections, in which users only need to visit a compromised Web page to infect their own PCs (and potentially begin infecting those of customers, colleagues, and other staff). Another worry is Web sites that distribute infected programs, applications, and Trojan files. Still another threat exists in the form of poisoned DNS attacks, whereby a compromised DNS server directs you to an unauthorized Web server. These compromised DNS servers are typically your ISP's systems. Users can protect themselves from all these threats by changing the way their computers process DNS services. While a computer professional may be required to implement the switch, OpenDNS offers free DNS services to protect users against common phishing, spyware, and other Web-based hazards.

3.10 NETWORKS AND COMMUNICAITON

Data refers to the raw facts that are collected while information refers to processed data that enables us to take decisions e.g. When result of a particular test is declared it contains data of all students, when you find the marks you have scored you have the information that lets you know whether you have passed or failed. The word data refers to any information which is presented in a form that is agreed and accepted upon is creators and users.

Data Communication

Data Communication is a process of exchanging data or information In case of computer networks this exchange is done between two devices over a transmission medium. This process involves a communication system which is made up of hardware and software. The hardware part involves the sender and receiver devices and the intermediate devices through which the data passes. The software part involves certain rules which specify what is to be communicated, how it is to be communicated and when. It is also called as a Protocol. The following sections describe the fundamental characteristics that are important for the effective working of data communication process and are followed by the components that make up a data communications system.

Characteristics of Data Communication

The effectiveness of any data communications system depends upon the following four fundamental characteristics:-

a. **Delivery:** The data should be delivered to the correct destination and correct user.
b. **Accuracy:** The communication system should deliver the data accurately, without introducing any errors. The data may get corrupted during transmission affecting the accuracy of the delivered data.
c. **Timeliness:** Audio and Video data has to be delivered in a timely manner without any delay; such a data delivery is called real time transmission of data.
d. **Jitter:** It is the variation in the packet arrival time. Uneven Jitter may affect the timeliness of data being transmitted.

Components of Data Communication

A Data Communication system has five components as shown in the diagram below:-

e. **Message:** Message is the information to be communicated by the sender to the receiver.
f. **Sender:** The sender is any device that is capable of sending the data (message).
g. **Receiver:** The receiver is a device that the sender wants to communicate the data (message).
h. **Transmission Medium:** It is the path by which the message travels from sender to receiver. It can be wired or wireless and many subtypes in both.
i. **Protocol:** It is an agreed upon set or rules used by the sender and receiver to communicate data. A protocol is a set of rules that governs data communication without which the communicating entities are like two persons trying to talk to each other in a different language without know the other language.

Data Representation

Data is collection of raw facts which is processed to deduce information. There may be different forms in which data may be represented. Some of the forms of data used in communications are as follows:-

j. **Text:** Text includes combination of alphabets in small case as well as upper case. It is stored as a pattern of bits.

Prevalent encoding system: ASCII, Unicode etc.

k. **Numbers:** Numbers include combination of digits from 0 to 9. It is stored as a pattern of bits. Prevalent encoding system: ASCII, Unicode etc.
l. **Images:** In computers images are digitally stored. A Pixel is the smallest element of an image. To put it in simple terms, a picture or image is a matrix of pixel elements. The pixels are represented in the form of bits. Depending upon the type of image (black n white or color) each pixel would require different number of bits to represent the value of a pixel. The size of an image depends upon the number of pixels (also called resolution) and the bit pattern used to indicate the value of each pixel. Commonly used Image formats: jpg, png, bmp, etc.
m. **Audio Data:** Audio Data can also be in the form of sound which can be recorded and broadcasted. Example: What we hear on the radio is a source of data or information. Audio data is continuous, not discrete. Commonly used audio formats: mp3 etc.
n. **Video:** Video refers to broadcasting of data in form of picture or movie. Commonly used audio formats: mp4, mkv etc.

Data Flow

Devices on a network communicate with each other by sending and receiving data. The data can flow between the two devices in the following ways.

o. Simplex
p. Half Duplex
q. Full Duplex

Simplex

In Simplex, communication is unidirectional only one of the devices sends the data and the other one only receives the data. Example: a CPU sends data while a monitor only receives data.

Half Duplex

In half duplex both the stations can transmit as well as receive but not at the same time. When one device is sending other can only receive and vice- versa Example: A walkie-talkie.

Full Duplex

In Full duplex mode, both stations can transmit and receive at the same time. Example: mobile phones

3.11 COMPUTER NETWORK

A computer network can be defined as a collection of nodes which is used for data communications. A node can be any device capable of transmitting or receiving data. The communicating nodes have to be connected by communication links. A Compute network should ensure:

a. **Reliability** of the data communication process
b. **Security** of the data
c. **Performance** by achieving higher throughput and smaller delay times

Categories of Network

Networks are categorized on the basis of their size. The three basic categories of computer networks are:-

a. **Local Area Network (LAN)** is usually limited to a few kilometers of area. It may be privately owned and could be a network inside an office on one of the floor of a building or a LAN could be a network consisting of the computers in an entire building.
b. **Wide Area Network (WAN)** is made of all the networks in a (geographically) large area. The network in an entire state could be termed as a WAN.
c. **Metropolitan Area Network (MAN)** is of size between LAN & WAN. It is larger than LAN but smaller than WAN. It may comprise the entire network in a city.

Protocol

A Protocol is one of the components of a data communications system which specifies the rules for communication between two or more parties. Without protocol communication cannot occur. The sending device cannot just send the data and expect the receiving device to receive and further interpret it correctly. When the sender sends a message it may consist of text, number, images, etc. which are converted into bits and grouped into blocks to be transmitted and often certain additional information called control information is also added to help the receiver interpret the data. For successful communication to occur, the sender and receiver must agree upon certain rules called protocol. A Protocol is defined as a set of rules that governs data communications. A protocol defines what is to be communicated, how it is to be communicated and when it is to be communicated.

Elements of a Protocol

There are three key elements of a protocol:

a. **Syntax**: It means the structure or format of the data. It is the arrangement of data in a particular order.
b. **Semantics**: It tells the meaning of each section of bits and indicates the interpretation of each section. It also tells what action/decision is to be taken based on the interpretation.
c. **Timing:** It tells the sender about the readiness of the receiver to receive the data and also intimates the sender at what rate the data should be sent to the receiver to avoid overwhelming the receiver.

External Services

Any service that is not sourced from a particular organisations network may be termed as an External Service. Such services are provided by an **external service** provider (ESP). Examples of this are: Consumer email **services** such as Yahoo, live.com, or Google mail etc.

Policy on Use of External Services

Sensitive data should neither be stored on nor sent to, from or through any external service provider (ESP) unless one of the two following conditions has been met:

a. The organisation has a contract with the ESP that specifically addresses such use of sensitive data.

b. Encryption methods that meet the requirements of the organisation are implemented to protect this data.

3.12 CLOUD COMPUTING

Cloud computing is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage. With the advent of this technology, the cost of computation, application hosting, content storage and delivery is reduced significantly. Cloud computing is a practical approach to experience direct cost benefits and it has the potential to transform a data centre from a capital-intensive set up to a variable priced environment.

The idea of cloud computing is based on a very fundamental principal of „reusability of IT capabilities'. The difference that cloud computing brings compared to traditional concepts of "grid computing", "distributed computing", "utility computing", or "autonomic computing" is to broaden horizons across organizational boundaries.

Cloud Computing Models

Cloud Providers offer services that can be grouped into three categories viz.

a. **Software as a Service (SaaS):** In this model, a complete application is offered to the customer, as a service on demand. A single instance of the service runs on the cloud & multiple end users are serviced. On the customers‟ side, there is no need for upfront investment in servers or software licenses, while for the provider, the costs are lowered, since only a single application needs to be hosted & maintained. Today SaaS is offered by companies such as Google, Salesforce, Microsoft, Zoho, etc.

b. **Platform as a Service (Paas):** Here, a layer of software, or development environment is encapsulated & offered as a service, upon which other higher levels of service can be built. The customer has the freedom to build his own applications, which run on the provider's infrastructure. To meet manageability and scalability requirements of the applications, PaaS providers offer a predefined combination of OS and application servers, such as LAMP platform (Linux, Apache, MySql and PHP), restricted J2EE, Ruby etc. Google‟s App Engine, Force.com, etc are some of the popular PaaS examples.

c. **Infrastructure as a Service (Iaas):** IaaS provides basic storage and computing capabilities as standardized services over the network. Servers, storage systems, networking equipment, data centre space etc. are pooled and made available to handle workloads. The customer would typically deploy his own software on the infrastructure. Some common examples are Amazon, GoGrid, 3 Tera, etc.

Understanding Public and Private Clouds

Enterprises can choose to deploy applications on Public, Private or Hybrid clouds. Cloud Integrators can play a vital part in determining the right cloud path for each organization.

Public Cloud

Public clouds are owned and operated by third parties; they deliver superior economies of scale to customers, as the infrastructure costs are spread among a mix of users, giving each individual client an attractive low-cost, “Pay-as-you-go” model. All customers share the same infrastructure pool with limited configuration, security protections, and availability variances. These are managed and supported by the cloud provider. One of the advantages of a Public cloud is that they may be larger than an enterprises cloud, thus providing the ability to scale seamlessly, on demand.

Private Cloud

Private clouds are built exclusively for a single enterprise. They aim to address concerns on data security and offer greater control, which is typically lacking in a public cloud. There are two variations to a private cloud: - On-premise Private Cloud: On-premise private clouds, also known as internal clouds are hosted within one’s own data center. This model provides a more standardized process and protection, but is limited in aspects of size and scalability. IT departments would also need to incur the capital and operational costs for the physical resources. This is best suited for applications which require complete control and configurability of the infrastructure and security. - Externally hosted Private Cloud: This type of private cloud is hosted externally with a cloud provider, where the provider facilitates an exclusive cloud environment with full guarantee of privacy. This is best suited for enterprises that don’t prefer a public cloud due to sharing of physical resources.

Hybrid Cloud

Hybrid Clouds combine both public and private cloud models. With a Hybrid Cloud, service providers can utilize 3rd party Cloud Providers in a full or partial manner thus increasing the flexibility of computing. The Hybrid cloud environment is capable of providing on-demand, externally provisioned scale. The ability to augment a private cloud with the resources of a public cloud can be used to manage any unexpected surges in workload.

Cloud Computing Benefits

Enterprises would need to align their applications, so as to exploit the architecture models that Cloud Computing offers. Some of the typical benefits are listed below:

d. **Reduced Cost:** There are a number of reasons to attribute Cloud technology with lower costs. The billing model is pay as per usage; the infrastructure is not purchased thus lowering maintenance. Initial expense and recurring expenses are much lower than traditional computing.

e. **Increased Storage**: With the massive Infrastructure that is offered by Cloud providers today, storage & maintenance of large volumes of data is a reality. Sudden workload spikes are also managed effectively & efficiently, since the cloud can scale dynamically.

f. **Flexibility**: This is an extremely important characteristic. With enterprises having to adapt, even more rapidly, to changing business conditions, speed to deliver is critical. Cloud computing stresses on getting applications to market very quickly, by using the most appropriate building blocks necessary for deployment.

Cloud Computing Challenges

Despite its growing influence, concerns regarding cloud computing still remain. In our opinion, the benefits outweigh the drawbacks and the model is worth exploring. Some common challenges are:

g. **Data Protection**: Data Security is a crucial element that warrants scrutiny. Enterprises are reluctant to buy an assurance of business data security from vendors. They fear losing data to competition and the data confidentiality of consumers. In many instances, the actual storage location is not disclosed, adding onto the security concerns of enterprises. In the existing models, firewalls across data centres (owned by enterprises) protect this sensitive information. In the cloud model, Service providers are responsible for maintaining data security and enterprises would have to rely on them.

h. **Data Recovery and Availability**: All business applications have Service level agreements that are stringently followed. Operational teams play a key role in management of service level agreements and runtime governance of applications. In production environments, operational teams support.

i. Appropriate clustering and Fail over

Data Replication

ii. System monitoring (Transactions monitoring, logs monitoring and others)

iii. Maintenance (Runtime Governance)

iv. Disaster recovery

v. Capacity and performance management

If, any of the above mentioned services is under-served by a cloud provider, the damage & impact could be severe.

i. **Management Capabilities**: Despite there being multiple cloud providers, the management of platform and infrastructure is still in its infancy. Features like „Auto- scaling‟ for example, are a crucial requirement for many enterprises. There is huge potential to improve on the scalability and load balancing features provided today.

j. **Regulatory and Compliance Restrictions**: In some of the European countries, Government regulations do not allow customer's personal information and other sensitive information to be physically located outside the state or country. In order to meet such requirements, cloud providers need to setup a data center or a storage site exclusively within the country to comply with regulations. Having such an infrastructure may not always be feasible and is a big challenge for cloud providers.

With cloud computing, the action moves to the interface — that is, to the interface between service suppliers and multiple groups of service consumers. Cloud services will demand expertise in distributed services, procurement, risk assessment and service negotiation — areas that many enterprises are only modestly equipped to handle.

IT Infrastructure

IT infrastructure refers to the composite hardware, software, network resources and services required for the existence, operation and management of an enterprise IT environment. It allows an organization to deliver IT solutions and services to its employees, partners and/or customers and is usually internal to an organization and deployed within owned facilities. In other words, IT infrastructure consists of all components that somehow play a role in overall IT and IT-enabled operations. It can be used for internal business operations or developing customer IT or business solutions. Typically, a standard IT infrastructure consists of the following components:

a. **Hardware**: Servers, computers, data centres, switches, hubs and routers, etc.
b. **Software:** Enterprise resource planning (ERP), customer relationship management (CRM), productivity applications and more.
c. **Network:** Network enablement, Internet connectivity, firewall and security.
d. **Meatware:** Human users, such as network administrators (NA), developers, designers and generic end users with access to any IT appliance or service are also part of an IT infrastructure, specifically with the advent of user-centric IT service development.

3.13 SUMMARY

1. Security controls are safeguards or countermeasures to avoid, detect, counteract, or minimize security risks to physical property, information, computer systems, or other assets.
2. Technical security involves the use of safeguards incorporated in computer hardware, operations or applications software, communications hardware and software, and related devices.
3. The purpose of access control software is to control sharing of data and programs between users.
4. After access to the system has been granted, the next step is to control access to the data and programs residing in the system.
5. Passwords are used to verify that the user of an ID is the owner of the ID.
6. Dial-up access to a computer system increases the risk of intrusion by hackers.
7. In networks that contain personal computers or are connected to other networks, it is difficult to determine whether dial-up access is available or not because of the ease with which a modem can be added to a personal computer to turn it into a dial-up access point.
8. An audit trail is a record of system activities that enables the reconstruction and examination of the sequence of events of a transaction, from its inception to output of

final results.

9. Although the technology behind the development of protection methods continues to improve in leaps and bounds, the threats against computers and the data, they contain, still remain.
10. The word data refers to any information which is presented in a form that is agreed and accepted upon is creators and users.
11. Data Communication is a process of exchanging data or information In case of computer networks this exchange is done between two devices over a transmission medium.
12. IT infrastructure refers to the composite hardware, software, network resources and services required for the existence, operation and management of an enterprise IT environment.

3.14 CHECK YOUR PROGRESS

1. ___ _ _ _ help to reduce the risk of damage or loss by stopping, deterring, or slowing down an attack against an asset.
2. __ _________ controls are technical or administrative safeguards to minimize loss or
   unavailability due to threats acting on their matching vulnerability.
3. Technical controls are also referred to as _____ _ controls.
4. ____ ___technical controls are used to prevent unauthorized personnel or programs from gaining remote access to computing resources.
5. In many computer systems, access to data and programs is implemented
   __ ___________ __that designate which users are allowed access.
6. ___ _ _ control systems require that all changes to production programs be implemented by library control personnel instead of the programmers who created the changes.
7. ___ _ _ are usually about the size of a credit card and contain a chip with logic functions and information that can be read at a remote terminal to identify a specific user’s privileges.
8. Currently, the best dial-up access controls use a _________ to intercept calls, verify
   the identity of the caller (using a dynamic password mechanism), and switch the user to authorized

computing resources as requested.

9. ___ _ _ __ technical controls warn personnel of violations or attempted violations of preventive technical controls.
10. ___ __ __ reports should be frequently and regularly reviewed by security officers and data base owners to identify and investigate successful or unsuccessful unauthorized accesses.
11. ___ _ _ _ controls exist to mitigate or lessen the effects of the threat being manifested.
12. A computer______ _can be defined as a collection of nodes which is used for data communications.
13. __ _________ is a computing paradigm, where a large pool of systems are connected in private or public networks, to provide dynamically scalable infrastructure for application, data and file storage.
14. In ________ ___model, a complete application is offered to the customer, as a service on demand.

ANSWERS TO CHECK YOUR PROGRESS

1. Controls
2. Security
3. Logical
4. Preventive
5. By access control lists
6. Library
7. Smart cards
8. Microcomputer
9. Detective
10. Violation
11. Corrective
12. Network
13. Cloud computing
14. Software as a service

MODEL QUESTIONS

1. What are security controls? What are the basic criteria based on which security controls are classified?
2. What are preventive security controls? Give examples.
3. What are smart cards? Why they are used?

4. What is encryption?
5. What is an audit trial?
6. What is an Intrusion Detection Systems?
7. What are the methods to safeguard your system from malicious attacks?

UNIT IV: SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

STRUCTURE

4.0 LEARNING OBJECTIVES

After going through this unit, you will be able to:

- Know the basics of Software Development Lifecycle (SDLC)
- Know the various stages involved in the SDLC
- Study the various models of SDLC

4.1 INTRODUCTION

We will initially start by briefly studying the concept of Software Development Life Cycle (SDLC). SDLC is a process used by the industry to design, develop and test high quality software. The process of SDLC aims to produce high quality software that meets or exceeds customer experience and expectations while meeting the financial and time constraints. This is also known as the Software Development Process for which tasks are pre – defined in the Software Development Life Cycle (SDLC). ISO/IEC 12207 is the international standard for software lifecycle processes. It aims to be the standard that defines all the tasks required for developing and maintaining software.

4.2 SOFTWARE DEVELOPMENT LIFECYCLE (SDLC)

4.2.1 Definition

SDLC is a process that an organisation follows for the development of a software project. It consists of a detailed plan describing how to develop, maintain, replace and alter or enhance specific software. The life cycle defines a methodology for improving the quality of software and the overall development process.

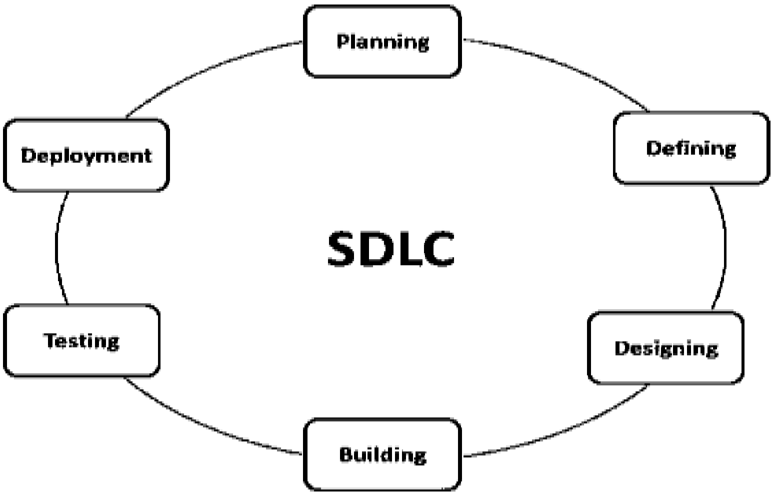


Figure 5: Stages of Software Development Life Cycle

Stages in SDLC

The various steps of Software Development Life Cycle are briefly described below:

1. **Stage 1: Planning and Requirement Analysis:** The most important and fundamental stage in SDLC is Planning, which may also be referred to as Requirement Analysis. It is performed by a team, generally having adequate seniority and experience, with inputs from the customer, the sales department, market surveys and domain experts in the industry. This information is then used to plan the basic project approach and to conduct product feasibility study in the operational, technical and economical areas. Planning for the quality assurance requirements and identification of the risks associated with the project is also done in the planning stage. The outcome of the technical feasibility study is to define the various technical approaches that can be followed to implement the project successfully with minimum risks.
2. **Stage 2: Defining Requirements:** After the requirement analysis, the next step is to clearly define and document the product requirements. This documentation is then discussed with and is approved after incorporation of suggestions/variations by the customer/analyst. This is done through SRS – Software Requirement Specification document which consists of all the product requirements to be designed and developed during the project life cycle.
3. **Stage 3: Designing the product architecture:** Software Requirement Specification (SRS) document forms the basis for best product architecture of the product to be developed. Based on the requirements specified in SRS, usually more than one design approach for the product architecture is proposed and documented in Detailed Design Document which may also be referred to as a DDS - Design Document Specification. This DDS is reviewed by all the important stakeholders and based on various parameters as risk assessment, product robustness, design modularity , budget and time constraints , the best design approach is selected for the product. A design approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules if any. The internal design of all the modules of the proposed architecture should be clearly defined with the minutest of the details in DDS.
4. **Stage 4: Building or Developing the Product:** During this stage of SDLC the actual development starts based on the DDS and the process for building of the product is initiated. The programming code is written as per DDS during this stage. If the designing is accomplished in a detailed and structured manner, error free code generation can be achieved within short time frames. Developers have to follow the coding guidelines defined by their organization and programming tools like compilers, interpreters, debuggers etc. are used to generate the code. Different high

level programming languages such as C, C++, Pascal, Java, and PHP are used for coding. The programming language is chosen with respect to the type of software being developed.

5. **Stage 5: Testing the Product:** This stage is usually a subset of all the stages as in the modern SDLC models, the testing activities are mostly involved in all the stages of SDLC. However this stage refers to the testing only stage of the product where products defects are reported, tracked, fixed and retested, until the product reaches the quality standards defined in the SRS.
6. **Stage 6: Deployment in the Market and Maintenance:** After the product is tested and ready it is released formally deployed in the appropriate market based on the Business strategy of the organization. The product at this stage may be subjected to Site Acceptance Trials (SAT) in a limited segment and later, tested in the real business environment with the User testing the product in the real world scenario. Then based on the feedback, the product may be released as it is or with suggested enhancements in the targeting market segment. After the product is released in the market, its maintenance is done for the existing customer base.

4.3 SDLC MODELS

There are various software development life cycle models defined and designed which are followed during software development process. These models are also referred as "Software Development Process Models". Each process model follows a Series of steps unique to its type, in order to ensure success in process of software development. Following are the most important and popular SDLC models followed in the industry:

- Waterfall Model
- Iterative Model
- Spiral Model
- V-Model
- Big Bang Model
- Agile Model
- Rapid Application Development Model
- Software Prototyping Model

In the further chapters we would consider the detailed description of the various SDLC Models.

4.4 WATERFALL MODEL

The waterfall model is a popular version of the systems development life cycle model for software engineering. It is often considered as the classic approach to the systems development life cycle. The waterfall model describes a development method that is linear and sequential. Waterfall development has distinct goals for each phase of development. In a waterfall model, each phase must be completed before the next phase can begin and there is no overlapping in the phases. Once a phase of development is completed, the development proceeds to the next phase and there is no turning back. Waterfall model is the earliest SDLC approach that was used for software development.

4.4.1 Waterfall Model Design

Waterfall approach was one of the earliest of SDLC Models to be used widely in Software Engineering and considering its linearity it ensured success of the project. In "The Waterfall" approach, the whole process of software development is divided into separate phases which

neither overlap nor are retractable. In Waterfall model the outcome of one phase acts as the input for the next phase sequentially.

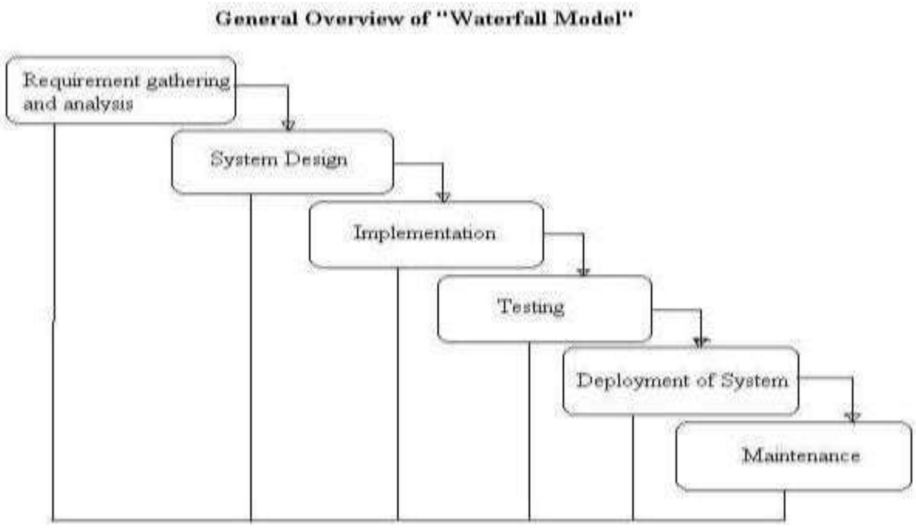

General Overview of "Waterfall Model"
Requirement gathering and analysis
System Design
Implementation
Testing
Deployment of System
Maintenance

Figure 6 : Waterfall model

4.4.2 Stages of the Waterfall model

The various stages of the Waterfall model are:

1. **Requirement Gathering and analysis:** All possible requirements of the system to be developed are captured in this phase and documented in a Software Requirement Specification document. In order to understand the requirements, various brainstorming and walkthrough sessions are organized. During this stage the requirements feasibility test is also carried out to ensure that the requirements are testable.
2. **System Design:** The requirement specifications from first phase are studied in this phase and system design is prepared. System Design helps in specifying hardware and system requirements and also helps in defining overall system architecture.
3. **Implementation:** With inputs from system design, the system is first developed in small programs/codes called units, which are integrated in the next phase. Each unit is developed and tested for its functionality which is referred to as Unit Testing.
4. **Integration and Testing:** All the units developed in the implementation phase are integrated into a system after testing of each unit to ensure that it works as expected. The progress on testing is tracked through tools like traceability matrices etc. Post integration the entire system is tested for any faults and failures.
5. **Deployment of system:** Once the functional and non-functional testing is done, the product is deployed in the customer environment or released into the market and a

sanity check is performed in the environment after the application is deployed to ensure the application does not break.

6. **Maintenance:** There are some issues which come up in the client environment. To fix those issues patches are released. Also to enhance the product some better versions are released. Maintenance is done to deliver these changes in the customer environment.All these phases are cascaded to each other in which progress is seen as flowing steadily downwards like a waterfall through the phases. The next phase is started only after the defined set of goals are achieved for previous phase and it is signed off, so the name "Waterfall Model". In this model phases do not overlap.

4.4.3 Application

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

a. Requirements are very well documented, clear and fixed.
b. Product definition is stable.
c. Technology is understood and is not dynamic.
d. There are no ambiguous requirements.
e. Ample resources with required expertise are available to support the product.
f. The project is short.

4.4.4 Advantages and Disadvantages of Waterfall Model

Advantages

The advantage of waterfall development is that it allows for departmentalization and control. A schedule can be set with deadlines for each stage of development and a product can proceed through the development process model phases one by one.

Development moves from concept, through design, implementation, testing, installation, troubleshooting, and ends up at operation and maintenance. Each phase of development proceeds in strict order.

Disadvantages

The disadvantage of waterfall development is that it does not allow for much reflection or revision. Once an application is in the testing stage, it is very difficult to go back and change something that was not well-documented or thought upon in the concept stage.

4.5 ITERATIVE MODEL

In Iterative model, iterative process starts with a simple implementation of a small set of the software requirements and iteratively enhances the evolving versions until the complete system is implemented and ready to be deployed. An iterative life cycle model does not attempt to start with a full specification of requirements. Instead, development begins by specifying and implementing just part of the software, which is then reviewed in order to identify further requirements. This process is then repeated, producing a new version of the software at the end of each iteration of the model.

4.5.1 Iterative Model design

Iterative process starts with a simple implementation of a subset of the software requirements and iteratively enhances the evolving versions until the full system is implemented. At each iteration, design modifications are made and new functional capabilities are added. The basic idea behind this method is to develop a system through repeated cycles iterative and in smaller portions at a time incremental.

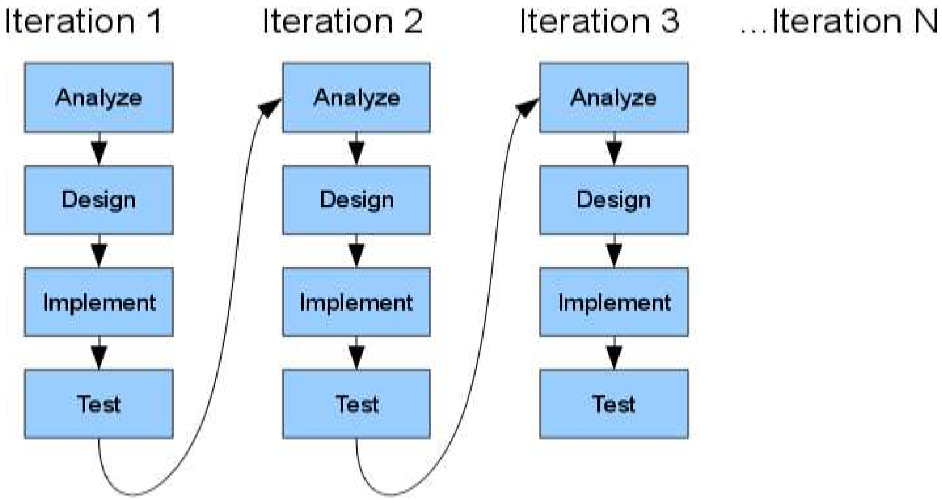


Figure 7: Iterative Model

Iterative and Incremental development is a combination of both iterative design or iterative method and incremental build model for development. "During software development, more than one iteration of the software development cycle may be in progress at the same time." and "This process may be described as an "evolutionary acquisition" or "incremental build" approach.

In incremental model the whole requirement is divided into various builds. During each iteration, the development module goes through the requirements, design, implementation and testing phases. Each subsequent release of the module adds function to the previous release. The process continues till the complete system is ready as per the requirement.
The key to successful use of an iterative software development lifecycle is rigorous validation of requirements, and verification & testing of each version of the software against those requirements within each cycle of the model. As the software evolves through successive cycles, tests have to be repeated and extended to verify each version of the software.

4.5.2 Application

Like other SDLC models, Iterative and incremental development has some specific applications in the software industry. This model is most often used in the following scenarios:

a. Requirements of the complete system are clearly defined and understood.
b. Major requirements must be defined; however, some functionalities or requested enhancements may evolve with time.

c. There is a time to the market constraint.
d. A new technology is being used and is being learnt by the development team while working on the project.
e. Resources with needed skill set are not available and are planned to be used on contract basis for specific iterations.
f. There are some high risk features and goals which may change in the future.

4.5.3 Advantages and Disadvantages of Iterative Model

Advantages

The advantage of this model is that there is a working model of the system at a very early stage of development which makes it easier to find functional or design flaws. Finding issues at an early stage of development enables to take corrective measures in a limited budget.

Disadvantages

The disadvantage with this SDLC model is that it is applicable only to large and bulky software development projects. This is because it is hard to break a small software system into further small serviceable increments/modules.

4.6 SPIRAL MODEL

The spiral model combines the idea of iterative development with the systematic, controlled aspects of the waterfall model. The spiral model is a risk-driven process model generator for software projects. Based on the unique risk patterns of a given project, the spiral model guides a team to adopt elements of one or more process models, such as incremental, waterfall, or evolutionary prototyping.

Spiral model is a combination of iterative development process model and sequential linear development model i.e. waterfall model with very high emphasis on risk analysis. It allows for incremental releases of the product, or incremental refinement through each iteration around the spiral.

4.6.1 Spiral Model design

The spiral model has four distinct phases. A software project repeatedly passes through these phases in iterations called Spirals.

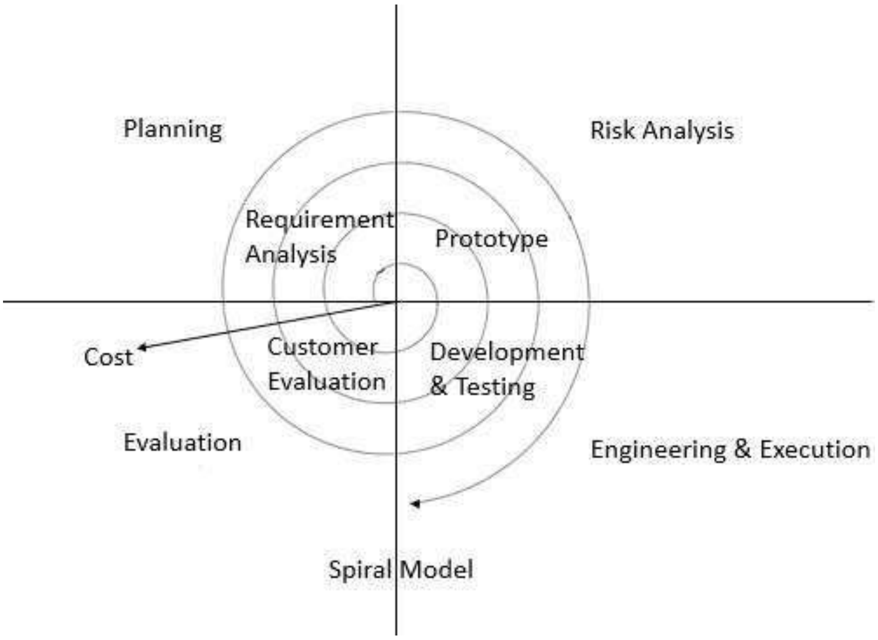


Figure 8: Spiral Model

1. **Planning (Determine Objectives):** This phase starts with gathering and analyzing the business requirements in the baseline spiral. In the subsequent spirals as the product

   matures, identification of system requirements, subsystem requirements and unit requirements are all done in this phase.

   This also includes understanding the system requirements by continuous communication between the customer and the system analyst. At the end of the spiral the product is deployed in the identified market.
2. **Design:** Design phase starts with the conceptual design in the baseline spiral and involves architectural design, logical design of modules, physical product design, which may be referred to as a prototype, and final design in the subsequent spirals.
3. **Construct or Build:** Construct phase refers to production of the actual software product at every spiral. In the baseline spiral when the product is just thought of and the design is being developed a Proof of Concept (POC) is developed in this phase to get customer feedback. Then in the subsequent spirals with higher clarity on requirements and design details a working model of the software called build is produced with a version number. These builds are sent to customer for feedback.
4. **Evaluation and Risk Analysis:** Risk Analysis includes identifying, estimating, and monitoring technical feasibility and management risks, such as schedule slippage and cost overrun. After testing the build, at the end of first iteration, the customer evaluates the software and provides feedback.

4.6.2 Application

Spiral Model is very widely used in the software industry as it is in sync with the natural development process of any product i.e. learning with maturity and also involves minimum risk for the customer as well as the development firms. Following are the typical uses of Spiral model:

a. When there is a budget constraint and risk evaluation is important.

b. For medium to high-risk projects.

c. Long-term project commitment because of potential changes to economic priorities as the requirements change with time.

d. Customer is not sure of their requirement which is usually the case.

e. Requirements are complex and need evaluation to get clarity.

f. New product line which is released in phases to get enough customer feedback.

g. Significant changes are expected in the product during the development cycle.

4.6.3 Advantages and Disadvantages of Spiral Model

Advantages

The advantage of spiral lifecycle model is that it allows for elements of the product to be added in when they become available or known. This assures that there is no conflict with previous requirements and design. This method is consistent with approaches that have multiple software builds and releases and allows for making an orderly transition to a maintenance activity. Another positive aspect is that the spiral model forces early user involvement in the system development effort.

Disadvantages

On the other side, it takes very strict management to complete such products and there is a risk of running the spiral in indefinite loop. So the discipline of change and the extent of taking change requests is very important to develop and deploy the product successfully.

4.7 V – MODEL

The V - model is SDLC model where execution of processes happens in a sequential manner in V shape. It is also known as Verification and Validation model. V - Model is an extension of the waterfall model and is based on association of a testing phase for each corresponding development stage. This means that for every single phase in the development cycle there is a directly associated testing phase. This is a highly disciplined model and next phase starts only after completion of the previous phase.

4.7.1 V- Model design

Under V-Model, the corresponding testing phase of the development phase is planned in parallel. So there are Verification phases on one side of the V and Validation phases on the other side (Please see Figure 5 below). Coding phase joins the two sides of the V-Model.

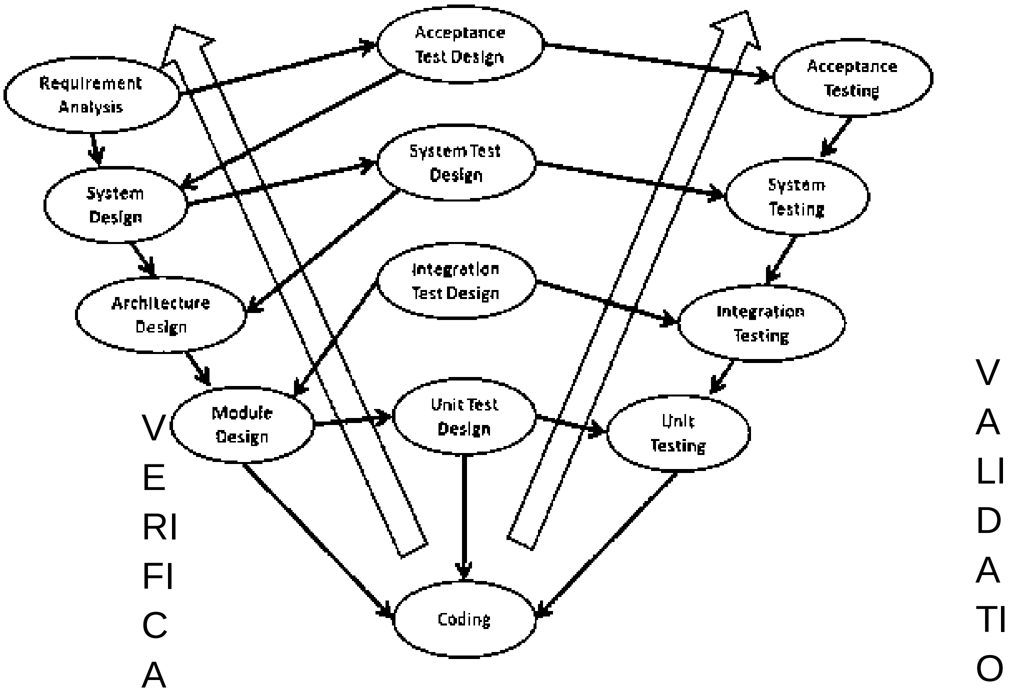


Figure 9:V Model

4.7.2 Verification Phases

Following are the Verification phases in V-Model:

1. **Business Requirement Analysis:** This is the first phase in the development cycle where the product requirements are understood from the customer perspective. This phase involves detailed communication with the customer to understand his expectations and exact requirement. This is a very important activity and need to be

   managed well, as most of the customers are not sure about what exactly they need. The acceptance test design planning is done at this stage as business requirements can be used as an input for acceptance testing.
2. **System Design:** Once you have the clear and detailed product requirements, it’s time to design the complete system. System design would comprise of understanding and detailing the complete hardware and communication setup for the product under development. System test plan is developed based on the system design. Doing this at an earlier stage leaves more time for actual test execution later.
3. **Architectural Design:** Architectural specifications are understood and designed in this phase. Usually more than one technical approach is proposed and based on the technical and financial feasibility the final decision is taken. System design is broken down further into modules taking up different functionality. This is also referred to as High Level Design (HLD). The data transfer and communication between the internal modules and with the other systems is clearly understood and defined in this stage. With this information, integration tests can be designed and documented during this stage.

4. **Module Design:** In this phase the detailed internal design for all the system modules is specified, referred to as Low Level Design (LLD). It is important that the design is compatible with the other modules in the system architecture and the other external systems. Unit tests are an essential part of any development process and helps eliminate the maximum faults and errors at a very early stage. Unit tests can be designed at this stage based on the internal module designs.

4.7.3 Coding Phase

The actual coding of the system modules designed in the design phase is taken up in the Coding phase. The best suitable programming language is decided based on the system and architectural requirements. The coding is performed based on the coding guidelines and standards. The code goes through numerous code reviews and is optimized for best performance before the final build is checked into the repository.

4.7.4 Validation Phases

Following are the Validation phases in V-Model:

5. **Unit Testing:** Unit tests designed in the module design phase are executed on the code during this validation phase. Unit testing is the testing at code level and helps eliminate bugs at an early stage, though all defects cannot be uncovered by unit testing.
6. **Integration Testing:** Integration testing is associated with the architectural design phase. Integration tests are performed to test the coexistence and communication of the internal modules within the system.
7. **System Testing:** System testing is directly associated with the System design phase. System tests check the entire system functionality and the communication of the system under development with external systems. Most of the software and hardware compatibility issues can be uncovered during system test execution.
8. **Acceptance Testing:** Acceptance testing is associated with the business requirement analysis phase and involves testing the product in user environment. Acceptance tests

   uncover the compatibility issues with the other systems available in the user environment. It also discovers the non-functional issues such as load and performance defects in the actual user environment.

4.7.5 Application

V- Model application is almost same as waterfall model, as both the models are of sequential type. Requirements have to be very clear before the project starts, because it is usually expensive to go back and make changes. This model is used in the medical development field, as it is strictly disciplined domain. Following are the suitable scenarios to use V-Model:

a. Requirements are well defined, clearly documented and fixed.
b. Product definition is stable.
c. Technology is not dynamic and is well understood by the project team.
d. There are no ambiguous or undefined requirements.
e. The project is short.

4.7.6 Advantages and Disadvantages of Software Prototyping V-Model

Advantage

The advantage of V-Model is that it's very easy to understand and apply. The simplicity of this model also makes it easier to manage.

Disadvantage

The disadvantage is that the model is not flexible to changes and just in case there is a requirement change, which is very common in today's dynamic world, it becomes very expensive to make the change.

4.8 BIG BANG MODEL

The Big Bang model is SDLC model where we do not follow any specific process. The development just starts with the required money and efforts as the input, and the output is the software developed which may or may not be as per customer requirement. Big Bang Model is SDLC model where there is no formal development followed and very little planning is required. Even the customer is not sure about what exactly he wants and the requirements are implemented on the fly without much analysis. Usually this model is followed for small projects where the development teams are very small.

4.8.1 Big Bang Model design and Application

Big bang model comprises of focusing all the possible resources in software development and coding, with very little or no planning. The requirements are understood and implemented as they come. Any changes required may or may not need to revamp the complete software.

This model is ideal for small projects with one or two developers working together and is also useful for academic or practice projects. It's an ideal model for the product where requirements are not well understood and the final release date is not given.

4.8.2 Advantages and Disadvantages of Big Bang Model

Advantages

The advantage of Big Bang is that it is very simple and requires very little or no planning. It is easy to manage and no formal procedures arerequired.

Disadvantage

The Big Bang model is a very high risk model and changes in the requirements or misunderstood requirements may even lead to complete reversal or scraping of the project. It is ideal for repetitive or small projects with minimum risks.

4.9 AGILE MODEL

Agile SDLC model is a combination of iterative and incremental process models with focus on process adaptability and customer satisfaction by rapid delivery of working software product. Agile Methods break the product into small incremental builds. These builds are provided in iterations. Each iteration typically lasts from about one to three weeks. Every iteration involves cross functional teams working simultaneously on various areas like planning, requirements analysis, design, coding, unit testing, and acceptance testing.

At the end of the iteration a working product is displayed to the customer and important stakeholders.

4.9.1 Concept of Agility

Agile model believes that every project needs to be handled differently and the existing methods need to be tailored to best suit the project requirements. In agile the tasks are divided to time boxes (small time frames) to deliver specific features for a release.

Iterative approach is taken and working software build is delivered at the end of each iteration. Each build is incremental in terms of features; the final build holds all the features required by the customer.

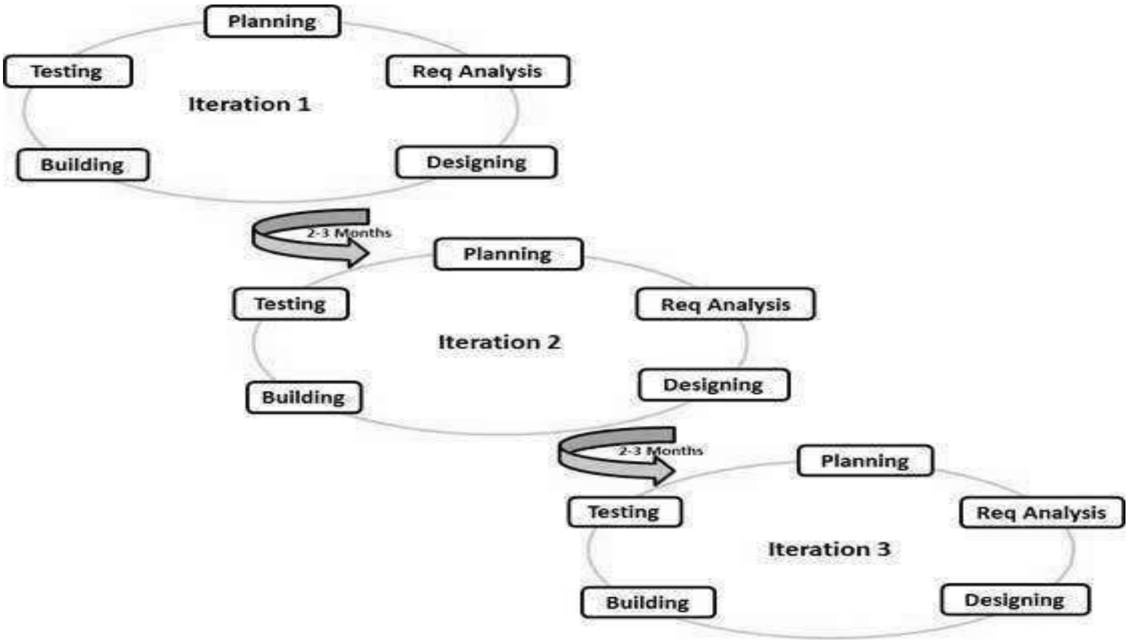


Figure 10: Agile Model

Agile thought process had started early in the software development and started becoming popular with time due to its flexibility and adaptability. The most popular agile methods include Rational Unified Process 1994, Scrum 1995, Crystal Clear, Extreme Programming 1996, Adaptive Software Development, Feature Driven Development, and Dynamic Systems Development Method DSDM 1995. These are now collectively referred to as agile methodologies, after the Agile Manifesto was published in 2001.

Agile Manifesto principles are as follows:

1. **Individuals and interactions** - in Agile development, the factors that gain importance are self-organization, motivation, interactions, co-location and pair programming.
2. **Working software** - Demonstration working software is considered the best means of communication with the customer to understand their requirement, instead of just depending on documentation.
3. **Customer collaboration** - As the requirements cannot be gathered completely in the beginning of the project due to various factors, continuous customer interaction

is very important to get proper product requirements.

4. **Responding to change** - Agile development is focused on quick responses to change and continuous development.

4.10 RAPID APPLICATION DEVELOPMENT MODEL

The Rapid Application Development (RAD) model is based on prototyping and iterative development with no specific planning involved. The process of writing the software itself involves the planning required for developing the product. Rapid Application development focuses on gathering customer requirements through workshops or focus groups, early testing of the prototypes by the customer using iterative concept, reuse of the existing prototypes components, continuous integration and rapid delivery.

RAD Concept

Rapid application development RAD is a software development methodology that uses minimal planning in favor of rapid prototyping. A prototype is a working model that is functionally equivalent to a component of the product. In RAD model the functional modules are developed in parallel as prototypes and are integrated to make the complete product for faster product delivery. Since there is no detailed preplanning, it makes it easier to incorporate the changes within the development process. RAD projects follow iterative and incremental model and have small teams comprising of developers, domain experts, customer representatives and other IT resources working progressively on their component or prototype. The most important aspect for this model to be successful is to make sure that the prototypes developed are reusable.

RAD Model Design

RAD model distributes the analysis, design, build, and test phases into a series of short, iterative development cycles. Following are the phases of RAD Model:

1. **Business Modeling:** The business model for the product under development is designed in terms of flow of information and the distribution of information between various business channels. A complete business analysis is performed to find the vital information for business, how it can be obtained, how and when is the information processed and what are the factors driving successful flow of information.
2. **Data Modeling:** The information gathered in the Business Modeling phase is reviewed and analyzed to form sets of data objects vital for the business. The attributes of all data sets is identified and defined. The relation between these data objects are established and defined in detail in relevance to the business model.
3. **Process Modeling:** The data object sets defined in the Data Modeling phase are converted to establish the business information flow needed to achieve specific business objectives as per the business model. The process model for any changes or enhancements to the data object sets is defined in this phase. Process descriptions for adding, deleting, retrieving or modifying a data object are given.
4. **Application Generation:** The actual system is built and coding is done by using automation tools to convert process and data models into actual prototypes.

5. **Testing and Turnover:** The overall testing time is reduced in RAD model as the prototypes are independently tested during iterations. However the data flow and the interfaces between all the components need to be thoroughly tested with complete test coverage. Since most of the programming components have already been tested, it reduces the risk of any major issues.

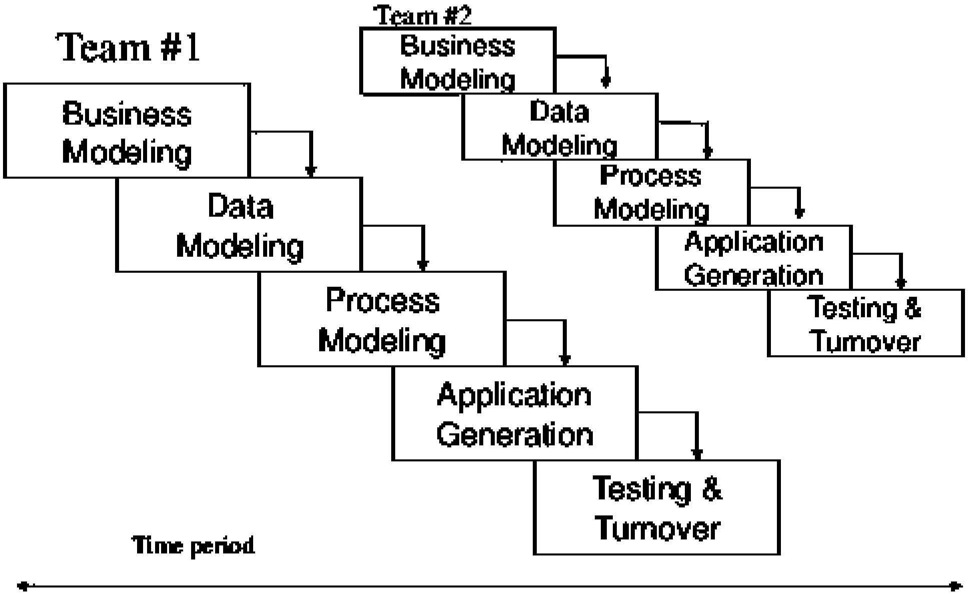


Figure 11: Rapid Application Development Model

RAD Model Applications

RAD model can be applied successfully to the projects in which clear modularization is possible. If the project cannot be broken into modules, RAD may fail. Following are the typical scenarios where RAD can be used:

a. RAD should be used only when a system can be modularized to be delivered in incremental manner.
b. It should be used if there’s high availability of designers for modeling.
c. It should be used only if the budget permits use of automated code generating tools.
d. RAD SDLC model should be chosen only if domain experts are available with relevant business knowledge.
e. Should be used where the requirements change during the course of the project and working prototypes are to be presented to customer in small iterations of 2-3 months.

Advantages and Disadvantages of RAD Model

3.11.4.1 Advantage

RAD model enables rapid delivery as it reduces the overall development time due to reusability of the components and parallel development.

3.11.4.2 Disadvantage

RAD works well only if high skilled engineers are available and the customer is also committed to achieve the targeted prototype in the given time frame. If there is commitment lacking on either side the model may fail.

4.11 SOFTWARE PROTOTYPING MODEL

The Software Prototyping refers to building software application prototypes which display the functionality of the product under development but may not actually hold the exact logic of the original software. Software prototyping is becoming very popular as a software development model, as it enables to understand customer requirements at an early stage of development. It helps get valuable feedback from the customer and helps software designers and developers understand about what exactly is expected from the product under development.

4.11.1 Software Prototyping Concept

- Prototype is a working model of software with some limited functionality.
- The prototype does not always hold the exact logic used in the actual software application and is an extra effort to be considered under effort estimation.
- Prototyping is used to allow the users evaluate developer proposals and try them out before implementation.
- It also helps understand the requirements which are user specific and may not have been considered by the developer during product design.

4.11.2 Steps involved in Software Prototyping

Stepwise approach to design a software prototype is as follows:

1. **Basic Requirement Identification:** This step involves understanding the very basics product requirements especially in terms of user interface. The more intricate details of the internal design and external aspects like performance and security can be ignored at this stage.
2. **Developing the initial Prototype:** The initial Prototype is developed in this stage, where the basic requirements are showcased and user interfaces are provided. These features may not exactly work in the same manner internally in the actual software developed and the workarounds are used to give the same look and feel to the customer in the prototype developed.
3. **Review of the Prototype:** The prototype developed is then presented to the customer and the other important stakeholders in the project. The feedback is collected in an

   organized manner and used for further enhancements in the product under development.
4. **Revise and enhance the Prototype:** The feedback and the review comments are discussed during this stage and some negotiations happen with the customer based on factors like, time and budget constraints and technical feasibility of actual implementation. The changes accepted are again incorporated in the new Prototype developed and the cycle repeats until customer expectations are met.

Prototypes can have horizontal or vertical dimensions. Horizontal prototype displays the user interface for the product and gives a broader view of the entire system, without concentrating on internal functions. A vertical prototype on the other side is a detailed elaboration of a specific function or a sub system in the product.

4.11.3 Software Prototyping Types

There are different types of software prototypes used in the industry. Following are the major software prototyping types used widely:

1. **Throwaway/Rapid Prototyping:** Throwaway prototyping is also called as rapid or close ended prototyping. This type of prototyping uses very little efforts with minimum requirement analysis to build a prototype. Once the actual requirements are understood, the prototype is discarded and the actual system is developed with a much clear understanding of user requirements.
2. **Evolutionary Prototyping:** Evolutionary prototyping also called as breadboard prototyping is based on building actual functional prototypes with minimal functionality in the beginning. The prototype developed forms the heart of the future prototypes on top of which the entire system is built. Using evolutionary prototyping only well understood requirements are included in the prototype and the requirements are added as and when they are understood.
3. **Incremental Prototyping:** Incremental prototyping refers to building multiple functional prototypes of the various sub systems and then integrating all the available prototypes to form a complete system.
4. **Extreme Prototyping:** Extreme prototyping is used in the web development domain. It consists of three sequential phases. First, a basic prototype with all the existing pages is presented in the html format. Then the data processing is simulated using a prototype services layer. Finally the services are implemented and integrated to the final prototype. This process is called Extreme Prototyping used to draw attention to the second phase of the process, where a fully functional UI is developed with very little regard to the actual services.

4.11.4 Software Prototyping Application

Software Prototyping is most useful in development of systems having high level of user interactions such as online systems. Systems which need users to fill out forms or go through various screens before data is processed can use prototyping very effectively to give the exact look and feel even before the actual software is developed. Software that involves too much of data processing and most of the functionality is internal with very little user interface does not usually benefit from prototyping. Prototype development could be an extra overhead in such projects and may need lot of extra efforts. Software prototyping is used in typical casesand the decision should be taken very carefully so that the efforts spent in building the prototype add considerable value to the final software developed.

4.12 SUMMARY

1. This was about the various SDLC models available and the scenarios in which these SDLC models are used. The information in this tutorial will help the project managers decide what SDLC model would be suitable for their project and it would

also help the developers and testers understand basics of the development model being used for their project.

2. We have discussed all the popular SDLC models in the industry, both traditional and Modern. This tutorial also gives you an insight into the applications, advantages and disadvantages of the SDLC models discussed.
3. Waterfall and V-model are traditional SDLC models and are of sequential type. Sequential means that the next phase can start only after the completion of first phase. Such models are suitable for projects with very clear product requirements and where the requirements will not change dynamically during the course of project completion.
4. Iterative and Spiral models are more accommodative in terms of change and are suitable for projects where the requirements are not so well defined, or the market requirements change quite frequently.
5. Big Bang model is a random approach to Software development and is suitable for small or academic projects.
6. Agile is the most popular model used in the industry. Agile introduces the concept of fast delivery to customers using prototype approach. Agile divides the project into small iterations with specific deliverable features. Customer interaction is the backbone of Agile methodology, and open communication with minimum documentation are the typical features of Agile development environment.
7. Rapid Application Development and Software Prototype are modern techniques to understand the requirements in a better way early in the project cycle. These techniques work on the concept of providing a working model to the customer and stockholders to give the look and feel and collect the feedback. This feedback is used in an organized manner to improve the product.

CHECK YOUR PROGRESS

1. The process of SDLC aims to produce high quality software that meets or exceeds customer experience and expectations while meeting the ____ and __ __________ constraints.
2. __ _________ is the international standard for software lifecycle processes.
3. SRS Stands for _____ ________ _.
4. A _____________ approach clearly defines all the architectural modules of the product along with its communication and data flow representation with the external and third party modules if any.

5. After the product is______ _ and ready it is
   released formally deployed in the appropriate
   market based on the Business strategy of the
   organization.
6. __ ______ model is often considered as the classic approach to the systems
   development life cycle.
7. The advantage of waterfall development is that it allows for ______ _ __
   and______ __ __.
8. The advantage of______ __model is that there is a working model of the system
   at a very early stage of development which makes it
   easier to find functional or design flaws.
9. ___ _ model is also known as Verification and Validation model.
10. __ ______ SDLC model is a combination of iterative and incremental process models
    with focus on process adaptability and customer
    satisfaction by rapid delivery of working software
    product.
11. __ _______ is a working model of software with some limited functionality.

ANSWERS TO CHECK YOUR PROGRESS

1 Financial, time
2 ISO/IEC 12207
3 Software Requirement Specification
4 Design
5 Tested
6 Waterfall
7 Departmentalization, control
8 Iterative
9V
10 Agile
11 Prototype

MODEL QUESTIONS

1. Define SDLC. Explain different stages of SDLC with the help of a diagram.
2. Name the different SDLC models.
3. Discuss waterfall model. What are the applications
   of waterfall model. Discuss its advantages and

disadvantages.

4. Discuss iterative model design.
5. Discuss spiral model.
6. Explain V-Model.
7. Explain Big Bang model.
8. What is the Concept of Agility?
9. Explain RAD Concept.
10. Explain the concept of Software Prototyping model.

UNIT V: AUTHENTICATION

STRUCTURE

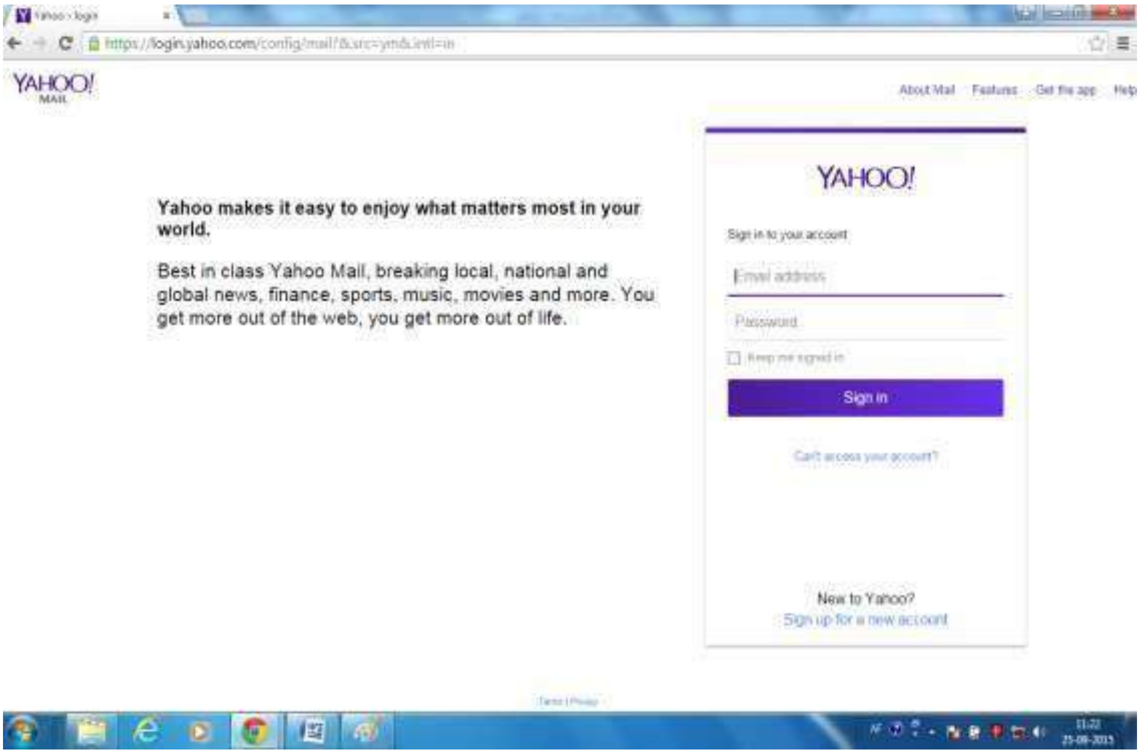
5.0 OBJECTIVES

After going through this unit, you will be able to:

- Know the basics authentication
- Differentiate identity, authentication and authorization
- Know types of authentication factors
- Study different types of authentication methods and protocols
- Study different types of authentication methods and protocols
- Set a strong passwords for your accounts
- Know the bad password combinations

5.1 INTRODUCTION

I hope all the readers are internet users! What is the first screen you encounter whenever you open your Gmail, Yahoomail or Rediffmail account? Yes you have guessed it right, you are landed to login page of the website.



Yahoo makes it easy to enjoy what matters most in your world.

Best in class Yahoo Mail, breaking local, national and global news, finance, sports, music, movies and more. You get more out of the web, you get more out of life.

Figure 12: Login Screen of Yahoo

Have you ever thought why we require to login? Why we are not landed to our mailbox directly? You need to authenticate yourself before you are given access to your mailbox so that the unauthorized person does not have access to your communication. So login-password is the methods of authentication. Let us first discuss what authentication is!

5.2 AUTHENTICATION

5.2.1 Definition of Authentication

Authentication[11] is the act of confirming the truth of an attribute of a single piece of data (a datum) claimed true by an entity. In contrast with **identification** which refers to the act of stating or otherwise indicating a claim purportedly attesting to a person or thing's identity, authentication is the process of actually confirming that identity. It might involve confirming the identity of a person by validating their identity documents, verifying the validity of a Website with a digital certificate, tracing the age of an artifact by carbon dating, or ensuring that a product is what its packaging and labeling claim to be. In other words, authentication often involves verifying the validity of at least one form of identification. In digital world, we perform electronic authentication. Let us now talk about e-authentication.

5.2.2 Definition of Electronic Authentication

Electronic authentication[12], also referred to as **e-authentication** is the process of establishing confidence in user identities electronically presented to an information system. In online environments, the username identifies the user, while the password authenticates that the user is who he claim to be. E-authentication presents a technical challenge when this process involves the remote authentication of individual people over a network, for the purpose of electronic government and commerce.

Authentication in the context of a user accessing an application tells an application who the current user is and whether or not they're present[13]. A full authentication protocol will probably also tell you a number of attributes about this user, such as a unique identifier, an email address, and what to call them when the application says "Good Morning". Authentication is all about the user and their presence with the application, and an internet-scale authentication protocol needs to be able to do this across network and security boundaries.

5.2.3 Authentication vs. Authorization

The word authentication is often confused with authorization. Let us discuss the difference between the two. As stated in the above paragraph, authentication is process of verifying the claim of the person whom he claims to be by the use of some personal identifiers. Whereas authorization is the process of allocation of access rights to the used after the identity of the person is confirmed through authorization. Often the organizations have hierarchical structure. There are workers at the lower level, supervisors at the middle level and manager & higher management like General Manager at the top level. Different kinds of privileges are granted at different levels. For example, let us discuss an example of a School/College Management System. The hierarchical structure is as follows:

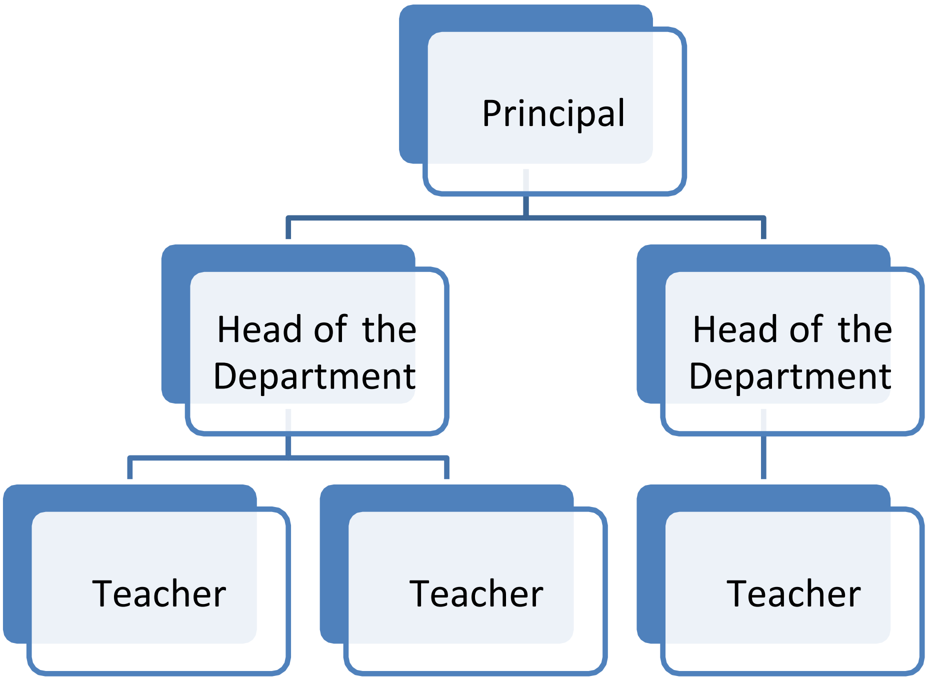


Figure 13: Organizational structure in a school

The teacher feeds the marks/attendance of the student in the School Management System. So the teacher is given the write permission to enter the marks/attendance of a particular subject and he can view/enter details pertaining to his subject only. The head of the department is one level-up in the hierarchy and is responsible for all the subjects, teachers and students of a particular department. So he may be given both read and write permission so that he can view/enter the subjects allocated to him as well as view the details of the subjects under his department. Principal is the head of the school. He is responsible for all the subjects, teachers and students of the school. Some students may have short attendance and therefore are not allowed to sit in the final examination due to short attendance. He student may submit medical certificate or may produce an evidence of attending a school activity like sports, cultural festival etc. After approval, he may need to modify the existing attendance. Marks, etc. therefore, Principal may need all the three permissions like enter, view and modify.

Whenever a person login into the system, he may be asked for user login and password. Login is for verifying who the user is and the password is for verifying the user who he claims to be. Once it is verified that it's a teacher login, head login or principal log based on authorization, the user may be allowed to perform one/all of the activities from write, view or modify. After a person, program or computer has successfully been identified and authenticated then it must be determined what informational resources they are permitted to access and what actions they will be allowed to perform (run, view, create, delete, or change). This is called authorization.

5.2.4 Types of Authentication Factors

In authentication, we generally talk about three "factors" for determining identity[14]. A "factor" is a broad category for establishing that you are who you claim to be. The three types of authentication factors are:

A. Something you know (a password, a PIN, the answer to a "security question", etc.) Most of us are familiar with password authentication. To log onto a computer or network, you enter a user account name and the password assigned to that account. This password is checked against a database that contains all authorized users and their passwords. In a Windows 2000 network, for example, this information is contained in Active Directory. To preserve the security of the network, passwords must be "strong," that is, they should contain a combination of alpha and numeric characters and symbols, they should not be words that are found in a dictionary, and they should be relatively long (eight characters or more). In short, they should not be easily guessed. Password authentication is vulnerable to a password "cracker" who uses a brute force attack (trying every possible combination until hitting upon the right one) or who uses a protocol "sniffer" to capture packets if passwords are not encrypted when they are sent over the network.

B. Something you have (an ATM card, a smart card, a one-time-password token, etc.) Smart cards are credit card-sized devices that hold a small computer chip, which is used to store public and private keys and other personal information used to identify a person and authenticate him or her to the system. Logging onto the network with a smart card requires that you physically insert the card into (or slide it through) a reader and then enter a Personal Identification Number (PIN) in much the same way that you use an ATM card to access an automatic teller machine. Smart cards use cryptography-based authentication and provide stronger security than a password because in order to gain access, the user must be in physical possession of the card and must know the PIN.

C. Something you are (your fingerprint, retinal pattern, DNA)
An even more secure type of authentication than smart cards, biometric authentication involves the use of biological statistics that show that the probability of two people having identical biological characteristics such as fingerprints is infinitesimally small; thus, these biological traits can be used to positively identify a person. In addition to fingerprints, voice, retinal, and iris patterns are virtually unique to each individual and can be used for authentication purposes. This method of proving one's identity is very difficult to falsify, although it requires expensive equipment to input the fingerprint, voice sample, or eye scan. Another advantage over smart cards is that the user does not have to remember to carry a device; his or her biological credentials are never left at home.

Historically, most people have used the first of these three factors. Whenever you've logged into Facebook, you entered something you know: your user name and password. One of the less common, but growing, authentication methods is biometrics. A couple years ago, a major PC manufacturer ran a number of television commercials advertising their laptop models with a fingerprint scanner. The claim was that it was easy and secure to unlock the machine with a swipe of a finger. Similarly, Google introduced a service to unlock an Android smartphone by using facial recognition with the phone's built-in camera.

Pay attention readers, because I am about to remove the scales from your eyes. Those three factors I listed above? I listed them in decreasing order of security. "But how can that be?", you may ask. "How can my unchangeable physical attributes be less secure than a password? Everyone knows passwords aren't secure." The confusion here is due to subtle but important definitions in the meaning of "security".

Most common passwords these days are considered "insecure" because people tend to use short passwords which by definition have a limited entropy pool (meaning it takes a smaller amount of time to run through all the possible combinations in order to brute-force the password or run through a password dictionary). However, the pure computational complexity of the authentication mechanism is not the only contributor to security.

The second factor above, "something you have" (known as a token), is almost always of significantly higher entropy than anything you would ever use as a password. This is to eliminate the brute-force vulnerability of passwords. But, it comes with a significant downside as well: something you have is also something that can be physically removed from you. Where a well-chosen password can only be removed from you by social engineering (tricking you into giving it to an inappropriate recipient), a token might be slipped off your desk while you are at lunch.

Both passwords and tokens have an important side-effect that most people never think about until an intrusion has been caught: remediation. When someone has successfully learned your password or stolen your token, you can call up your helpdesk and immediately ask them to reset the password or disable the cryptographic seed in the token. Your security is now restored and you can choose a new password and have a new token sent to you.

However, this is not the case with a biometric system. By its very nature, it is dependent upon something that you cannot change. Moreover, the nature of its supposed security derives from this very fact. The problem here is that it's significantly easier to acquire a copy of someone's fingerprint, retinal scan or even blood for a DNA test than it is to steal a password or token device and in many cases it can even be done without the victim knowing. Many consumer retinal scanners can be fooled by a simple reasonably-high-resolution photograph of the person's eye (which is extremely easy to accomplish with today's cameras). Some of the more expensive models will also require a moving picture, but today's high-resolution

smartphone cameras and displays can defeat many of these mechanisms as well. It's well-documented that Android's face-unlock feature can be beaten by a simple photograph.

These are all technological limitations and as such it's plausible that they can be overcome over time with more sensitive equipment. However, the real problem with biometric security lies with its inability to replace a compromised authentication device. Once someone has a copy of your ten fingerprints, a drop of your blood from a stolen blood-sugar test, or a close- up video of your eye from a scoped video camera, there is no way to change this data. You can't ask helpdesk to send you new fingers, an eyeball, or DNA. Therefore, I contend that I lied to you above. There is no full third factor for authentication, because, given a sufficient amount of time, any use of biometrics will eventually degenerate into a non-factor. Given this serious limitation, one should never under any circumstances use biometrics as the sole form of authentication for any purpose whatsoever.

One other thought: have you ever heard the argument that you should never use the same password on multiple websites because if it's stolen on one, they have access to the others? Well, the same is true of your retina. If someone sticks malware on your cellphone to copy an image of your eye that you were using for "face unlock", guess what? They can probably use that to get into your lab too.

The moral of the story is this: biometrics are minimally useful, since they are only viable until the first exposure across all sites where they are used. As a result, if you are considering initiating a biometric-based security model, I encourage you to look into a two-factor solution involving passwords and a token of some kind.

5.2.5 Multi Factor and Two Factor Authentication

When two or more access methods are included as part of the authentication process, your implementing a multi-factor system[15]. A system that uses smartcards and passwords is referred to as a two-factor system. Two-factor authentication (also known as 2FA) is a technology patented in 1984 that provides identification of users by means of the combination of two different components. These components may be something that the user knows, something that the user possesses or something that is inseparable from the user. A good example from everyday life is the withdrawing of money from a cash machine. Only the correct combination of a bank card (something that the user possesses) and a PIN (personal identification number, i.e. something that the user knows) allows the transaction to be carried out. 2FA is ineffective against modern threats, like ATM skimming, phishing, and malware etc. Two-factor authentication is a type of multi-factor authentication. If you’ve ever used Google’s two-factor authentication to log in, you probably used a code stored on your smartphone to do so.

5.3 AUTHENTICATOIN METHODS AND PROTOCOLS

There are a large number of authentication methods and protocols that can be used, depending on the application and security requirements. In the following sections, we will discuss different types of popular authentication methods and protocols.

Kerberos

Kerberos[16] is a network authentication protocol. It is designed to provide strong authentication or client-server applications by using secret-key cryptography.It allows nodes communicating over a non-secure network to prove their identity to one another in a secure manner[17]. Its designers aimed it primarily at a client–server model and it provides mutual authentication—both the user and the server verify each other's identity. Kerberos protocol messages are protected against eavesdropping and replay attacks. It was developed by Massachusetts Institute of Technology (MIT) developed Kerberos to protect network services provided by Project Athena. The protocol is based on the earlier Needham–Schroeder symmetric key protocol. The client authenticates itself to the Authentication Server (AS) which forwards the username to a key distribution center (KDC). The KDC issues a ticket- granting ticket (TGT), which is time stamped, encrypts it using the user's password and returns the encrypted result to the user's workstation. This is done infrequently, typically at user logon; the TGT expires at some point, though may be transparently renewed by the user's session manager while they are logged in. When the client needs to communicate with another node ("principal" in Kerberos parlance) the client sends the TGT to the ticket- granting service (TGS), which usually shares the same host as the KDC. After verifying the TGT is valid and the user is permitted to access the requested service, the TGS issues a ticket and session keys, which are returned to the client.

The client then sends the ticket to the service server (SS) along with its service request.

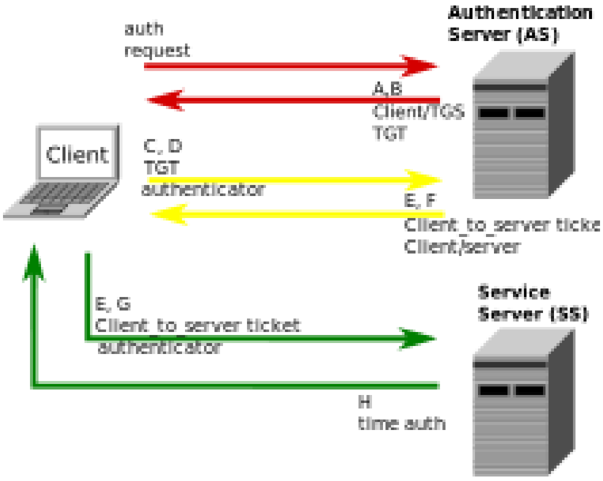


Figure 14: Kerberos Negotiations

Secure Sockets Layer(SSL)

The first thing you need to know about Secure Sockets Layer is that it is no longer called that[18]. For purist reasons, the protocol formerly known as SSL is now called Transport Layer Security, or TLS. The reasons for this name change are fairly esoteric and originate partly in a description of networking architecture as the Open Systems Interconnection seven-layer networking model. SSL hovered uncomfortably between the transport layer (4) and the presentation layer(6), and some experts felt, long after the OSI model had fallen out of fashion, that SSL was not really a layer at all. In the more recent TCP/IP model, which has fewer layers, SSL operates somewhere between the transport and application layers. Also, the protocol can - in principle - be applied to other networking protocols than just sockets, even though the vast majority of global communication are now made using the socket programming interface. Almost every product that implements TLS continues to refer to itas SSL, usually with some weasel words added like more properly known as TLS. The SSL protocol was originally invented by the Netscape Corporation as a way of providing secure browsing in their web browser. The Netscape Corporation was absorbed into AOL now.

The symmetric-key encryption is considerably more efficient than public key encryption. So, for encrypting bulk data transfer with high performance, symmetric-key encryption is a must. But to use it, both partners in the conversation must know a single shared secret key. But how do you share a secret key with a partner that you have never communicated with before? This is known as the key exchange problem. An early solution to this problem was for a trusted courier to carry the key, physically locked in a secure container, from one location to another. This is highly secure, but expensive and inconvenient, and hardly practical for electronic commerce. Furthermore, the same key is used for encrypting all traffic, which somewhat simplifies the possibility of an attacker breaking the key.

SSL, in brief, is a solution to the key exchange problem that is suitable for electronic communication. The two partners in the conversation must be identified as the client and the server as their roles are different: the conversation is not symmetric. The conversation is initiated by the client, who provides a list of suggested encryption techniques. The server responds with a certificate containing the server's public key, and an encryption technique that is acceptable to the client. The client validates the server's certificate, and uses the public key within it to encrypt a random string called the pre-master-secret, which it sends to the server. The server uses its private key to decrypt the pre-master-secret. At this point, the key exchange problem is solved: the client and server can both use the pre-master-secret to generate the key required by the mutually chosen encryption technique.

Both the client and server now possess a mutually chosen encryption algorithm and a key to use with it. They are now in a position to exchange secret encrypted messages using fast symmetric-key encryption, using a shared key that has never appeared in plain text in the conversation. Furthermore, a different key is used for each conversation, limiting the opportunity for an attacker to break it. Microsoft NTLM

NTLM[19] is a suite of authentication and session security protocols used in various Microsoft network protocol implementations and supported by the NTLM Security Support Provider ("NTLMSSP"). Originally used for authentication and negotiation of secure DCE/RPC, NTLM is also used throughout Microsoft's systems as an integrated single sign-on mechanism. It is probably best recognized as part of the "Integrated Windows Authentication" stack for HTTP authentication; however, it is also used in Microsoft implementations of SMTP, POP3, IMAP (all part of Exchange), CIFS/SMB, Telnet, SIP, and possibly others.

The NTLM Security Support Provider provides authentication, integrity, and confidentiality services within the Window Security Support Provider Interface (SSPI) framework. SSPI specifies a core set of security functionality that is implemented by supporting providers; the NTLMSSP is such a provider. The SSPI specifies, and the NTLMSSP implements, the following core operations:

a. Authentication - NTLM provides a challenge-response authentication mechanism, in which clients are able to prove their identities without sending a password to the server.
b. Signing - The NTLMSSP provides a means of applying a digital "signature" to a message. This ensures that the signed message has not been modified (either accidentally or intentionally) and that that signing party has knowledge of a shared secret. NTLM implements a symmetric signature scheme (Message Authentication Code, or MAC); that is, a valid signature can only be generated and verified by parties that possess the common shared key.
c. Sealing - The NTLMSSP implements a symmetric-key encryption mechanism, which provides message confidentiality. In the case of NTLM, sealing also implies signing (a signed message is not necessarily sealed, but all sealed

messages are signed).

NTLM has been largely supplanted by Kerberos as the authentication protocol of choice for domain-based scenarios. However, Kerberos is a trusted-third-party scheme, and cannot be used in situations where no trusted third party exists; for example, member servers (servers that are not part of a domain), local accounts, and authentication to resources in an untrusted domain. In such scenarios, NTLM continues to be the primary authentication mechanism (and likely will be for a long time).

Password Authentication Protocol

Password authentication protocol (PAP) is an authentication protocol that uses a password[20].PAP is used by Point to Point Protocol to validate users before allowing them access to server resources. Almost all network operating system remote servers support PAP.

PAP transmits unencrypted ASCII passwords over the network and is therefore considered insecure. PAP is the protocol where two entities share a password in advance and use the password as the basis of authentication. Existing password authentication schemes can be categorized into two types: weak-password authentication schemes and strong-password authentication schemes. When compared to strong-password schemes, weak-password schemes tend to have lighter computational overhead, the designs are simpler, and implementation is easier, making them especially suitable for some constrained environments.

PAP is a two step authentication protocol which are:

d. Authentication Request: the device which initiated the communication send Authenticate-Request message to the responder. This Authenticate-Request message contains a name and a password.

e. Authentication Reply: Once the Authenticate-Request message receives the responder, it authenticates the message by checking the username and the password. If the authentication is successful, it reply back with Authenticate-Ack message else Authenticate-Nak massage is send back to the initiator.

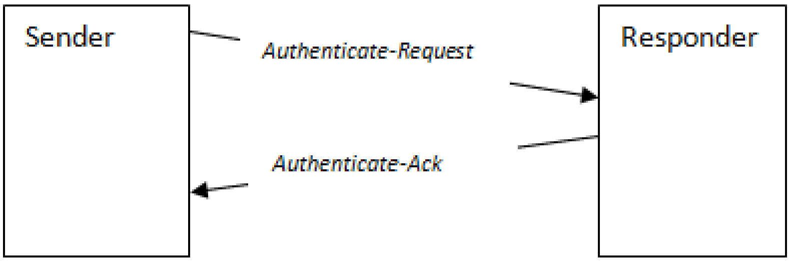


Figure 15: Authorization-Ack when authorization is secussful

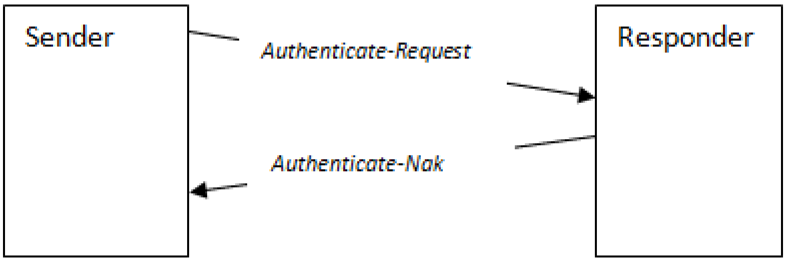


Figure 16: Authorization-Nak when authorization is not secussful

PAP is simple authentication protocol but have some serious security issues associated with it. First of all, it sends the username and the password as plain text, due to this becomes an easy prey for the hackers. Secondly, it does not keep any upper limit on the number of unsuccessful attempts on incorrect username and password, unlike most of the sites which block the attempts after three consecutive unsuccessful attempts.

Challenge-Handshake Authentication Protocol(CHAP)

Challenge-Handshake Authentication Protocol challenges a system to verify identity[21]. CHAP doesn't use userID/Password mechanism. Instead, the initiator sends a logon request from the client to the server. The server sends a challenge back to the client. The challenge is encrypted and then sent back to the server. The server compares the value from the client and, if the information matches, grants authorization. If the response fails, the session fails, and the request phase starts over.

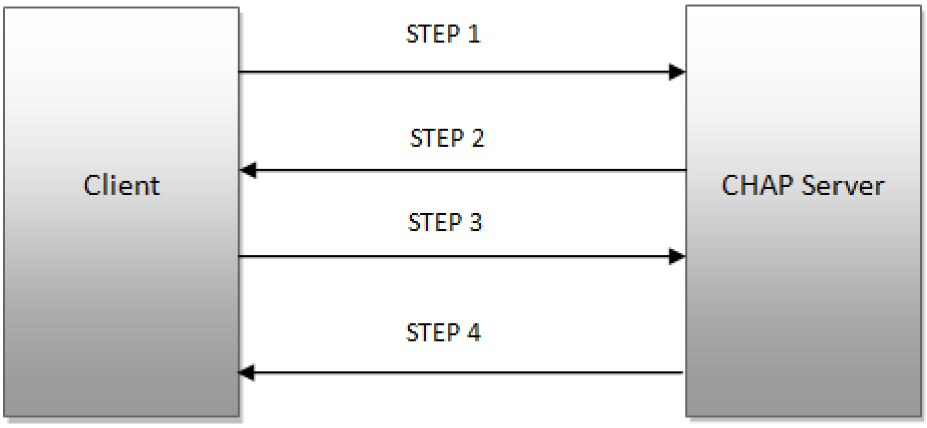


Figure 17: CHAP Authentication process

Following are the Steps[22]
:

f. STEP1: the authenticator (main server) sends a "challenge" message to the peer (client);
g. STEP2: the peer responds with a value calculated using a one-way hash function, using SHA checksum hash ;
h. STEP3: the authenticator checks the response against its own calculation of the expected hash value. If the values match, the authenticator acknowledges the authentication; otherwise it should terminate the connection ;
i. STEP4: at random intervals the authenticator sends a new challenge to the peer and repeats steps 1 through 3.

The challenge is the hashed value of the client password concatenated to a random number. The whole is hashed and represents the challenge. The main server sends the random number to the client which in the same way calculates the challenge and sends it to the main server for comparison.

Microsoft Challenge Handshake Authentication Protocol(MS-CHAP) It is the Microsoft version of the Challenge-Handshake Authentication Protocol[23]. MS-CHAP is used as one authentication option in Microsoft's implementation of the PPTP protocol

for virtual private networks. It is also used as an authentication option with RADIUS servers which are used for WiFi security using the WPA-Enterprise protocol. It is further used as the main authentication option of the Protected Extensible Authentication Protocol (PEAP).

Compared with CHAP, MS-CHAP:

a. Is enabled by negotiating CHAP Algorithm 0x80 (0x81 for MS-CHAPv2) in LCP option 3, Authentication Protocol
b. provides an authenticator-controlled password change mechanism
c. provides an authenticator-controlled authentication retry mechanism
d. defines failure codes returned in the Failure packet message field

Extensible Authentication Protocol

Extensible Authentication Protocol[24], or EAP, is an authentication framework frequently used in wireless networks and point-to-point connections. EAP is an authentication framework providing for the transport and usage of keying material and parameters generated by EAP methods. There are many methods defined by RFCs and a number of vendor specific methods and new proposals exist. EAP is not a wire protocol; instead it only defines message formats. Each protocol that uses EAP defines a way to encapsulate EAP messages within that protocol's messages.

Remote Authentication Dial-In User Service (RADIUS)

RADIUS is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for users who connect and use a network service[25]. RADIUS was developed by Livingston Enterprises, Inc. in 1991 as an access server authentication and accounting protocol and later brought into the Internet Engineering Task Force (IETF) standards. Because of the broad support and the ubiquitous nature of the RADIUS protocol, it is often used by ISPs and enterprises to manage access to the Internet or internal networks, wireless networks, and integrated e-mail services. These networks may incorporate modems, DSL, access points, VPNs, network ports, web servers, etc.

Certificates

This is another common form of authentication[26]. A server or certificate authority (CA) can issue a certificate that will be accepted by the challenging system. Certificates can either be physical access devices, such as smart cards, or electronic certificates that are used as part of the logon process. A certificate practice statement (CPS) outlines the rules used for issuing and managing certificates. A certificate revocation list (CRL) lists the revocations that must be addressed (often due to expiration) in order to stay current. A simple way to think of certificates is like hall passes at school.Security Tokens

These are similar to certificates. They contain the rights and access privileges of the token bearer as part of the token. Think of a token as a small piece of data that holds a sliver of information about the user. Many operating systems generate a token that is applied to every action taken on the computer system. If your token don’t grant you access to certain information, then either that information won’t be displayed or your access will be denied. The authentication system creates a token every time a user connects or a session begins. At the completion of a session, the token is destroyed.

5.4 Selecting a Strong Password

The weakest point in any security for your online accounts is usually your password[27]. One should pay attention to make sure your content is secure, protected, and can't be accessed by anyone other than the owner. But if someone else is able to guess or retrieve your password, they bypass almost every security measure provided by the host server as it will see this person as you. They could then make any changes they wish to your account. To avoid this scenario, this section will help you create strong passwords that are hard to guess or crack. Password-cracking techniques have matured quickly and significantly in the past few decades, but the way we create our passwords hasn't kept pace. As a result, the most common advice you'll hear about creating a strong password today is very outdated and impractical. A password created with that advice, like **jal43#Koo%a**, is very easy for a computer to break and very difficult for a human to remember and type.

The latest and most effective types of password attacks can attempt up to **350 billion guesses per second**, and that number will no doubt increase significantly over the next few years. Creating a strong password today requires modern techniques. There are many different approaches to generating a strong password, but password managers and passphrases are the best. Creating a strong password and changing it frequently is one of the smartest things a user can do to protect themselves when working online[28]. For many institutions, usernames and domains are generated based on a common formula: for Uttarakhand Open University, first initial and last name, with the jpande@uou.ac.in

Because of this, it's really not enough to rely fully on the security systems of the places that you use your password on. Often times, the least important passwords becomes the most important because hackers target those first. Sure, your bank is pretty secure. But if you use the same password in multiple locations, or iterations of the same password, it doesn't matter how secure your bank's security system is. Hackers start with the stupid websites that require a username and password to process web payments, or logins to read the news or comment on forums, or whatever. If the password you used to order a pizza last weekend is the same as the one you use to access your student loans…. well, no amount of security is going to stop someone when they know your password already.

5.4.1 Bad Password Combinations

Here are some of the most common passwords or password configurations people use. If this is you – it's time to change!

a. Your partner, child, or pet's name, possibly followed by a 0 or 1 (because they're always making you use a number, aren't they?)
b. The last 4 digits of your social security number.
c. 123 or 1234 or 123456.
d. "password"
e. Your city, or college, football team name.
f. Date of birth – yours, your partner's or your child's.
g. "god"
h. "money"
i. "love"

5.4.2 Tips for a Strong Password

c. **Length**. As computers that process Brute Force attacks (just running different combinations of passwords and usernames repeatedly until they get a result) become more intelligent, length becomes the defining variable in passwords that will take longer to process.
d. In conjunction with length, choose a **pass phrase** that means something to you instead of a word, a name, or a title. For example: "mydogslovetoeatswisscheese" is a better pass phrase than "Fido123".
e. Although it's not the 'rule' for unbreakable passwords anymore, **randomizing your capitalization** and using special characters are still useful and effective in making your password harder to crack.
f. Turn a phrase into a string of characters that look completely unrelated, but in reality is easy to remember. So "all creatures great and small" would become "acg8@s" or something similar.

g. **Don't use words that can be found in a dictionary or someone's name** . Password generators can crack these in no time. At the very least, use something slightly off the beaten path.

h. Have problems remembering lots of different passwords? Try using an encrypted password utility like Roboform, LastPass and KeePass for Windows, or 1Password for Macs.

i. Changing your password often is important too, but not terribly effective if all you do is change the number at the end of a word. Using passphrases instead of passwords means you have to change them less often, and are more likely to remember them when you do.

j. **Make a note**. Don't write your password down, but if you know you have trouble remembering, slip a piece of paper in your wallet with a clue that's significant only to you.

k. **Create a Passphrase instead of a Password** – A passphrase is similar to a password, except that it's based on a random collection of words, rather than just one. For example, **copy indicate trap bright**.

l. Because the length of a password is one of the primary factors in how strong it is, passphrases are much more secure than traditional passwords. At the same time, they are also much easier to remember and type.

m. They're not as strong as the kinds of passwords generated by password managers, but they're still a good option if you don't want to use a password manager. They're also the best way to generate the master password for a password manager or your operating system account, since those can't be automatically filled in by the password manager.

n. **Don't use the same password twice.** Many popular websites fail to adequately secure your password in their systems, and hackers routinely break into them and access hundreds of millions of accounts. If you reuse passwords from site to site, then someone who hacks into one site will be able to login to your account on other sites. At the very least, make sure that you have unique passwords for all sites that store financial or other sensitive data, or ones that could be used to hurt your reputation.

o. **Make sure your email password is also strong.** With many online services like WordPress.com, your email address serves as your identification. If a malicious user gains access to your email, they can easily reset your passwords and login to your account.

p. **Don’t share your passwords.** Even if you trust the person, it’s possible an attacker could intercept or eavesdrop on the transmission, or hack that person’s computer. If you suspect that someone else knows your password, you should change it immediately.

q. **Don’t send your password to anyone in an email.** E-mails are rarely encrypted, which makes them relatively easy for attackers to read. WordPress.com staff will never ask you for your password. If you must share a password, use a secure method of transmission like pwpush.com, and set the link to expire after the first view.

r. **Don’t save your passwords in a web browser.** They often fail to store the passwords in a secure manner, so use a password manager instead. See the section on password managers above for more information.

s. **Don’t save passwords or use “Remember Me” options on a public computer**. If you do, then the next person to use the computer will be able to access your account. Also make sure you log out or close your browser when you are done.

t. **Don’t write down your password.** If it’s written down somewhere and someone can find it, it’s not secure. Store passwords in a password manager instead, so that they’ll be encrypted. See the section on password managers above for more information. **The exception** to this rule is storing unrecoverable passwords (like the master password for a password manager, or your operating system account)

in a secure manner. One good way to secure them is to keep it in a safe deposit box, or locked in a safe.

u. **Don’t change your passwords, unless you suspect**

they've been compromised. As long as you have the type of strong password recommended in this article, changing it frequently will not do anything to minimize the risk of it being compromised. Because changing them can be a burden, it often tempts people to adopt bad practices in order to make the process easier, which increases their vulnerability to attacks. If you suspect someone has gained access to your account, though, then it's always a good precaution to change your password.

v. **Use a Password Manager**: A password manager is a software application on your computer or mobile device that generates very strong passwords and stores them in a secure database. You use a single passphrase to access the database, and then the manager will automatically enter your username and password into a website's login form for you. There are many different manager applications to choose from, so you'll need to pick which one you'd like to use, and then install it on your computer. These are the general steps, but you may want to check the documentation for your specific application for more details. Choose a password manager. Some popular ones are:

- 1Password (closed-source, commercial)
- LastPass (closed-source, free/commercial)
- Dashlane (closed-source, free/commercial)
- KeePass (open source, free)
- RoboForm (closed-source, commercial).

w. **Use Passphrase: Creating a passphrase follows similar rules to creating a traditional password, but it doesn't need to be as complex, because the length of the phrase will provide enough security to outweigh the simplicity.**

- Choose 4 random words. You can use the xkcd Passphrase Generator if you'd like, but it's better if you make up your own.
- Add spaces between the words if you prefer.
- Make a few of the letters upper-case.
- Add in a few number and symbols.

Things to avoid when using Passphrase

a. Don’t place the words in a **predictable pattern** or form a proper sentence; that would make it much easier to guess.
b. Don’t use **song lyrics, quotes or anything else that’s been published**. Attackers have massive databases of published works to build possible passwords from.
c. Don’t use any **personal information**. Even when combined with letters and numbers, someone who knows you, or can research you online, can easily guess a password with this information.

5.5 SUMMERY

1. **Authentication** is the act of confirming the truth of an attribute of a single piece of data (a datum) claimed true by an entity.
2. **Identification** which refers to the act of stating or otherwise indicating a claim purportedly attesting to a person or thing's identity, authentication is the process of actually confirming that identity.
3. **Electronic authentication**, also referred to as **e-authentication** is the process of establishing confidence in user identities electronically presented to an information system.
4. **Authorization** is the process of allocation of access rights to the used after the identity of the person is confirmed through authorization.
5. The three types of authentication factors are: Something you know (a password, a PIN, the answer to a “security question”, etc.), Something you have (an ATM card, a smart card, a one-time-password token, etc.}, Something you are (your fingerprint, retinal pattern, DNA)
6. When two or more access methods are included as part of the authentication process, it is known as multi-factor system.
7. Some of the popular authentication methods and protocols are: Kerberos, SSL, CHAP, PAP, MS-CHAP, MS-NTML,EAP, RADIUS, Certificate, Security tokens, etc.
8. Kerberos is a network authentication protocol. It is

designed to provide strong authentication for client-server applications by using secret-key cryptography.

9. The SSL protocol was originally invented by the Netscape Corporation as a way of providing secure browsing in their web browser.
10. Password authentication protocol (PAP) is an authentication protocol that uses a password.
11. Extensible Authentication Protocol, or EAP, is an authentication framework frequently used in wireless networks and point-to-point connections.
12. RADIUS is a networking protocol that provides centralized Authentication, Authorization, and Accounting (AAA) management for users who connect and use a network service.
13. This is another common form of authentication[29]. A server or certificate authority (CA) can issue a certificate that will be accepted by the challenging system.
14. Security tokens are similar to certificates. They contain the rights and access privileges of the token bearer as part of the token.
15. The weakest point in any security for your online accounts is usually your password.

16. The latest and most effective types of password attacks can attempt up to **350 billion guesses per second**, and that number will no doubt increase significantly over the next few years.

5.6 CHECK YOUR PROGRESS

1. Fill in the blanks:

I. __ ________ refers to the act of stating or otherwise indicating a claim
purportedly attesting to a person or thing's identity.

II. ____ __ _ is for verifying who the user is and the _______is for verifying the
user who he claims to be.

III. Massachusetts Institute of Technology (MIT) developed _____ to protect
network services provided by Project Athena

IV. Secure Sockets Layer is now known as ______ ______________ _.

V. The SSL protocol was originally invented by the _____ ____as a way of
providing secure browsing in their web browser

VI. CHAP stands for _____ ______________ _.

VII. __ _____ _can either be physical access devices, such as smart cards, or
electronic certificates that are used as part of the logon process.

2.State True or False

I. State true or False: You should use the same password on multiple websites.
II. State true or False: Withdrawing of money from a cash machine is an example of two-factor authentication.
III. State true or False: Kerberos is a network authentication protocol.
IV. State true or False: The symmetric-key encryption is considerably more efficient than public key encryption.
V. State true or False: Don’t save your passwords in a web browser.

ANSWERS TO CHECK YOUR PROGRESS

1. Fill in the blanks.

I. Identification
II. Login, password
III. Kerberos
IV. Transport Layer Security
V. Netscape Corporation
VI. Challenge-Handshake Authentication Protocol challenges
VII. Certificates

2. State True or False

I. False
II. True
III. True
IV. True
V. True

FURTHER READING

1. A mechanism for identity delegation at authentication level, N Ahmed, C Jensen - Identity and Privacy in the Internet Age - Springer 2009
2. " New NIST Publications Describe Standards for Identity Credentials and Authentication Systems" available at http://www.nist.gov/itl/csd/piv_090809.cfm
3. Forouzan (2007). Data Commn & Networking 4E Sie. McGraw-Hill Education (India) Pvt Limited. pp. 352–. ISBN 978-0-07-063414-5. Retrieved 24 November 2012.

4. Lloyd, Brian; Simpson, William Allen (October 1992). "Password Authentication Protocol". PPP Authentication Protocols. IETF. p. 2. RFC 1334. Retrieved 26 Sep. 2015 available at https://tools.ietf.org/html/rfc1334#page-2
5. "AAA and Network Security for Mobile Access. RADIUS, DIAMETER, EAP, PKI and IP mobility". M Nakhjiri. John Wiley and Sons, Ltd
6. Hassell, Jonathan (2002). RADIUS - Securing Public Access to Private Resources. O'Reilly & Associates. ISBN 0-596-00322-6.(Selecting a strong password, 2015)

MODEL QUESTIONS

1. What is Authentication? How it is different from electronic authentication?
2. Compare Authentication with authorization.
3. Explain different types of authentication factors in details.
4. What is multi-factor authentication?
5. Write a short note on Kerberos.
6. What are the core opertation specified by Security Support Provider Interface.
7. What are the two basic authentication steps of Password Authentication Protocol. Explain
8. Explain CHAP authentication process.
9. Define RADIUS.
10. What are the few bad password combination that people often use in daily life.
11. List some of the guidelines to create a strong password.

What is password manager? Give some examples of popular password managers.

UNIT VI: WIRELESS SECURITY

STRUCTURE

6.1 LEARNING OBJECTIVES

After going through this unit, you will able to:

- Define the terms cyber security
- Identify SSID
- Implementation of Encryption keys
- Understand the 802.11 IEEE wireless Standards
- Understand the wireless router
- Configure wireless router, WLAN

6.2 INTRODUCTION

Wireless security is the prevention of unauthorized access or damage to computers using wireless networks. The most common types of wireless security are Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access (WPA). WEP is a notoriously weak security standard. The password it uses can often be cracked in a few minutes with a basic laptop computer and widely available software tools. WEP is an old IEEE 802.11 standard from 1999, which was outdated in 2003 by WPA, or Wi-Fi Protected Access. WPA was a quick alternative to improve security over WEP. The current standard is WPA2; some hardware cannot support WPA2 without firmware upgrade or replacement. WPA2 uses an encryption device that encrypts the network with a 256-bit key; the longer key length improves security over WEP.

Many laptop computers have wireless cards pre-installed. The ability to enter a network whilemobile has great benefits. However, wireless networking is prone to some security issues. Hackers have found wireless networks relatively easy to break into, and even use wireless technology to hack into wired networks. As a result, it is very important that enterprises define effective wireless security policies that guard against unauthorized access to importantresources. Wireless Intrusion Prevention Systems (WIPS) or Wireless Intrusion Detection Systems (WIDS) are commonly used to enforce wireless security policies.

6.3 SERVICE SET IDENTIFICATION (SSID)

Service set identification (SSID) a series of 0 to 32 octets. It is used as a unique identifier fora wireless LAN. Since this identifier must often be entered into devices manually by a humanuser, it is often a human-readable string and thus commonly called the "network name".

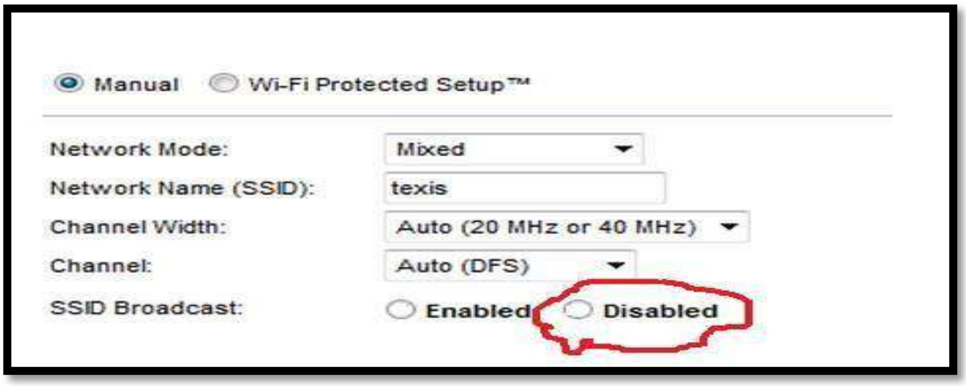
An SSID is the name of a wireless local area network (WLAN). All wireless devices on a WLAN must employ the same SSID in order to communicate with each other. A network administrator often uses a public SSID that is set on the access point and broadcast to all wireless devices in range. Some newer wireless access points disable the automatic SSID broadcast feature in an attempt to improve network security.

A common, albeit incorrect assumption, is that an SSID is a string of human-readable characters (such as ASCII), terminated by a NUL character (as in a C-string). SSIDs must be treated and handled as what they are, a series of 0 to 32 octets, some of which may not be human-readable. Note that the 2012 version of the 802.11 standard defines a primitive SSID Encoding, an Enumeration of UNSPECIFIED and UTF-8, indicating how the array of octets can be interpreted.

In an IBSS, the SSID is chosen by the client device that starts the network, and broadcasting of the SSID is performed in a pseudo-random order by all devices that are members of the network.

6.3.1 Security of SSID hiding

Every wireless router has the ability to broadcast its name, or SSID (Service Set Identifier). Disabling SSID broadcasting means that your wireless network won't appear in the list of "Available Wireless Networks" on any nearby computer. Theoretically, this makes your network more secure. Since your neighbors won't be able to see your network, how could they connect to it? Unfortunately, disabling the SSID does nothing to hide your wireless network from hackers using programs that scan the airwaves for wireless networks. In fact, itmakes it look like you've got something to hide, much like putting an expensive purchase in the back of a hatchback and covering it with a blanket does. Additionally, disabling SSID broadcasting makes it harder to troubleshoot connection problems, and also makes it difficultfor your guests to connect to your wireless network. Using WPA with a complex password means your neighbors can see your network, but they can't access it.



● Manual ○ Wi-Fi Protected Setup™

Network Mode: Mixed

Network Name (SSID): texis

Channel Width: Auto (20 MHz or 40 MHz)

Channel: Auto (DFS)

SSID Broadcast: ○ Enabled [✓ ○ Disabled]

Figure 18: SSID settings

6.4 ENCRYPTION METHODS

Encryption is used to hide or mask the data being sent through wireless transmission there areseveral popular and widely used encryption method used now a days including WEP (Wire Equivalent Privacy),WPA (Wi-Fi Protected Access) and WPA2(Wi-Fi Protected Access 2).

6.4.1 WEP (Wire Equivalent Privacy)

Wired Equivalent Privacy (**WEP**) is a security algorithm for IEEE 802.11 wireless networks. Introduced as part of the original 802.11 standard ratified in 1997,its intention was to provide data confidentiality comparable to that of a traditional wired network. WEP, recognizable by the key of 10 or 26 hexadecimal digits, was at one time widely in use and was often the first security choice presented to users by router configuration tools.

WEP is a notoriously weak security standard. The password it uses can often be cracked in a few minutes with a basic laptop computer and widely available software tools. WEP is an oldIEEE 802.11 standard from 1999, which was outdated in 2003 by WPA, or Wi-Fi Protected Access. WPA was a quick alternative to improve security over WEP.

6.4.2 Wi-Fi Protected Access (WPA) and Wi-Fi Protected Access II(WPA2)

Wi-Fi Protected Access (WPA) is a security standard that improves on older security standards by authenticating network users and providing more advanced encryption techniques. Wi-Fi Protected Access (WPA) and Wi-Fi Protected Access II (WPA2) are two most common security protocol and security certification programs developed by the Wi-Fi Alliance to secure wireless computer network. WPA was a quick alternative to improve security over WEP. The current standard is WPA2; some hardware cannot support WPA2 without firmware upgrade or replacement. WPA2 uses an encryption device that encrypts thenetwork with a 256-bit key; the longer key length improves security over WEP.

WPA become available in 2003 and was intended as an intermediate measure in anticipation of the availability of the more secure and complex WPA2. WPA2 become available in 2004.

A feature added to Wi-Fi, called Wi-Fi Protected Setup, allows WPA and WPA2 security to be bypassed and effectively broken in many situations. WPA and WPA2 security implemented without using the Wi-Fi Protected Setup feature are unaffected by the security vulnerability.

Wi-Fi Protected Access (WPA) is a specification of standard-based, interoperable security enhancements that increase the level of data protection and access control for existing and future wireless LAN system.

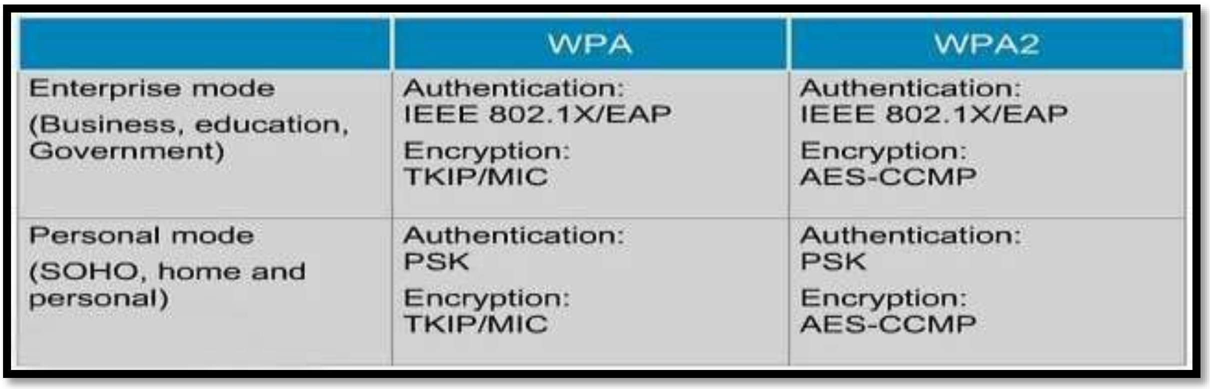
WPA offers the benefits Enhancement data security, robust key management, Data origin authentication and Data integrity protection.

Difference between WPA & WPA2

WPA (sometimes referred to as the draft IEEE 802.11i standard) became available in 2003. The Wi-Fi Alliance intended it as an intermediate measure in anticipation of the availability of the more secure and complex WPA2. WPA2 became available in 2004 and is a common shorthand for the full IEEE 802.11i (or IEEE 802.11i-2004) standard.

WPA and WPA2 Modes

WPA or Wi-Fi Protected Access leverages this authentication method of 802.1x plus EAP. Italso supports pre-shared keys or PSKs. This defines two levels of authentication, one more suitable for enterprise environments, business, education, government which uses per user persession and authentication keys obtained via the 802.1x EAP exchange. Another one which could be categorized as personal mode, more for home and personal use, which uses a Pre- Shared Key. It does not require the .1x EAP exchange and it can be set simply by configuringa pre-shared key manually.



	WPA	WPA2
Enterprise mode (Business, education, Government)	Authentication: IEEE 802.1X/EAP Encryption: TKIP/MIC	Authentication: IEEE 802.1X/EAP Encryption: AES-CCMP
Personal mode (SOHO, home and personal)	Authentication: PSK Encryption: TKIP/MIC	Authentication: PSK Encryption: AES-CCMP

Figure 19: Camparasion of WPA and WPA2

In WPA though, both modes would use WEP as the encryption algorithm and we know that is a vulnerable protocol and so some extra protection was built into WEP in the form of thosetwo protocols, the Temporal Key Integrity Protocol and the Message Integrity Check, which basically added longer rotating keys and some integrity measures. Similar modes exist in WPA2, which is an implementation of Wi-Fi Alliance of the IEEE 802.11i standard. The main difference here is that WEP is no longer the encryption algorithm. The framework is still 802.1x EAP or pre-shared keys, but again the encryption algorithm is now AES or Advanced Encryption Standard.

6.5 MAC Filtering

In computer networking, **MAC Filtering** refers to a security access control method whereby the 48-bit address assigned to each network card is used to determine access to the network. MAC addresses are uniquely assigned to each card, so using MAC filtering on a network permits and denies network access to specific devices through the use ofblacklists and whitelists. While the restriction of network access through the use of lists is straightforward, an individual person is not identified by a MAC address, rather a device only, so an authorized person will need to have a whitelist entry for each device that he or shewould use to access the network.

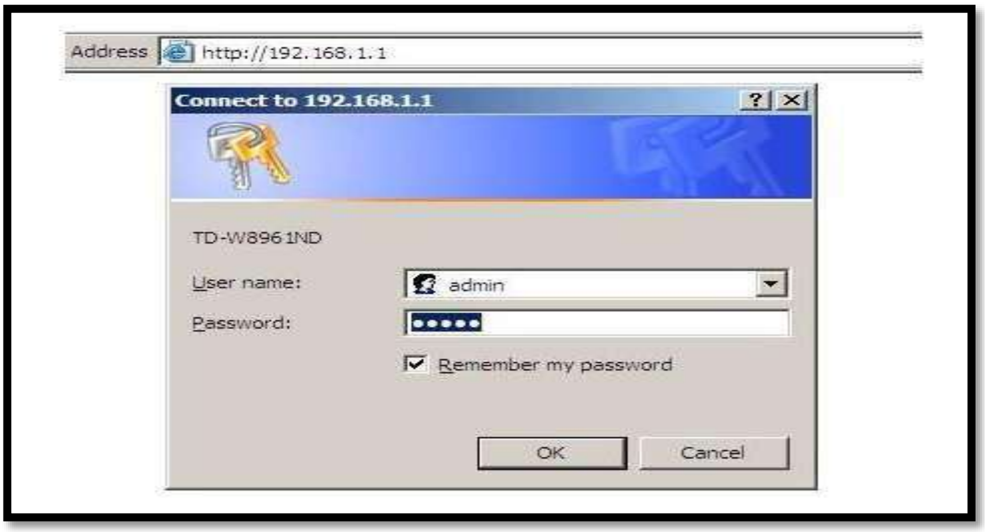
While giving a wireless network some additional protection, MAC filtering can be circumvented by scanning a valid MAC (via airodump-ng) and then spoofing one's own MAC into a validated one. This can be done in the Windows Registry or by using command line tools on a Linux platform. MAC Address filtering is often referred to as Security through obscurity. Unfortunately, using MAC Filtering may lead to a false sense of security. Also referencing to IP blocking.

MAC filtering is not an effective control in wireless networking as attackers can eavesdrop on wireless transmissions. However MAC filtering is more effective in wired networks, sinceit is more difficult for attackers to identify authorized MACs.

MAC filtering is also used on enterprise wireless networks with multiple access points to prevent clients from communicating with each other. The access point can be configured to only allow clients to talk to the default gateway, but not other wireless clients. It increases theefficiency of access to network.

6.5.1 Configuration of Wireless MAC address filter on wireless router

Step1: First step you need to open the web browser and then you have to typehttp://192.168.0.1or http://192.168.1.1 after that press enter. The user name and password areboth “admin”



Address http://192.168.1.1

Connect to 192.168.1.1

TD-W8961ND

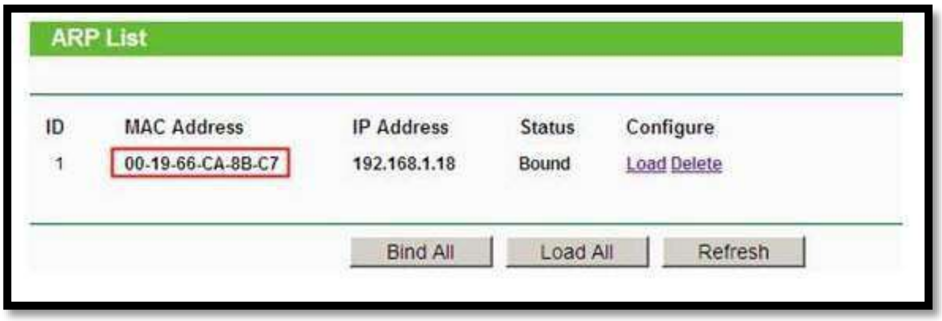
User name: admin

Password: •••••

☑ Remember my password

Figure 20: Configuring wireless router

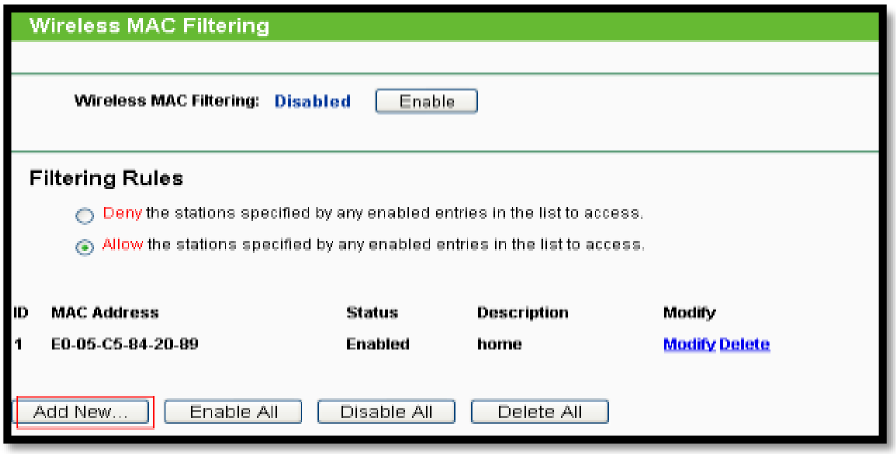
Step 2: In the second step Go to IP & MAC Binding->ARP List page, you can find theMAC address of the all the devices which are connected to the router.



ARP List

ID	MAC Address	IP Address	Status	Configure
1	00-19-66-CA-8B-C7	192.168.1.18	Bound	Load Delete

Step 3: In the third step Go to **Wireless->Wireless MAC Filtering** page, click the **Add New** button.



Wireless MAC Filtering

Wireless MAC Filtering: Disabled

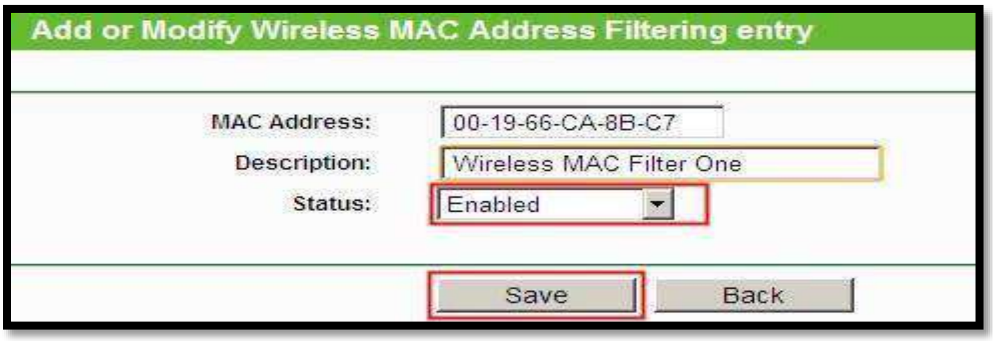
Filtering Rules

○ Deny the stations specified by any enabled entries in the list to access.

● Allow the stations specified by any enabled entries in the list to access.

ID	MAC Address	Status	Description	Modify
1	E0-05-C5-84-20-89	Enabled	home	Modify Delete

Step 4: In fourth step type in the MAC address you want to allow or deny to access the router, and give a description for this item. The status should be Enabled and at last, click theSave button. You need add items in this way one by one, 64 is the maximum number.



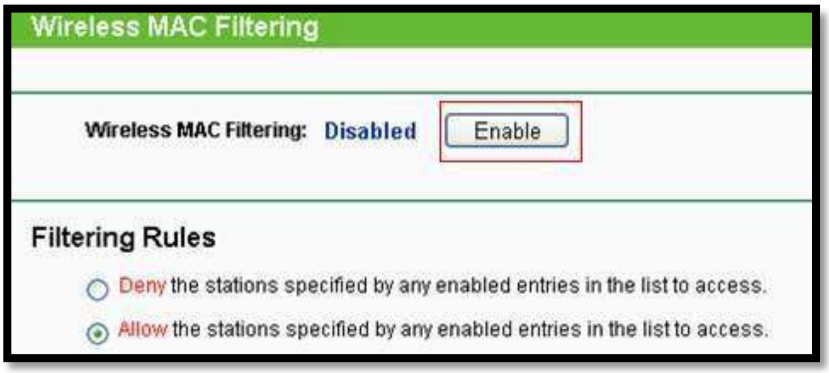
Add or Modify Wireless MAC Address Filtering entry

MAC Address: 00-19-66-CA-8B-C7

Description: Wireless MAC Filter One

Status: Enabled

Step 5: In fifth step at the end, about the Filtering Rules, please choose Allow/Deny andEnable the Wireless MAC Filtering function.



Wireless MAC Filtering

Wireless MAC Filtering: Disabled

Filtering Rules

○ Deny the stations specified by any enabled entries in the list to access.

● Allow the stations specified by any enabled entries in the list to access.

6.6 WIRELESS ROUTER

A wireless router is the most important piece of equipment that allows the internet to work. The most popular wireless routers are ones with built in DSL or cable modems. Wireless routers are the main part transferring emails and signals coming across your Internet connection into a wireless broadcast, sort of like a cordless phone base station. It makes sure information goes to the right place. Wireless routers can be connected to the internet through a hard wire connection or a wireless connection. They make the internet much easier to use.

Although wireless routers have a lot of positives, there are some negatives. They can be veryunsafe. People can have the ability to tap into your computer. You can prevent this by settingup security features. Firewalls, firmware, and other virus scanners are important to protect your wireless router.

Figure 21: Wireless Router

Radio Waves:

Wi-Fi uses radio waves like other wireless devices such as laptops and cellular phones, and radios. For the most part, to communicate with the wireless network, there is a two-way radiocontact between computers. Because of this, the process that happens when wireless devices communicate via Wi-Fi is simple. First, the computer's wireless adapter changes its data to radio frequency and uses an antenna to send it out. Then, the computer's wireless router gets the signal and translates it interprets the signal and sends the information to the internet usingan Ethernet connection. That is the process to send information. To receive information, the process works in reverse.

Modem:

A modem is a device that converts the digital signals from a computer into specific frequencies that can travel to television or telephone lines. It transfers the information from the internet to the wireless router. At the destination point the receiving modem transfers the data that was sent back into data information. The modem allows the computer to translate information from one computer to another.

The modem can be either internal or external to the computer. Regardless if your cable modem is outside or inside the computer all modems have a few key components:

- A tuner- receiver of data
- A demodulator-changes signal into a simple signal which is processed by the convertor.
- A modulator- convert data into radio-frequency signals for transmission
- A media access control (MAC) unit- interface between hardware and the software ofdifferent protocols.

6.7 HOW TO CREATE A WIRELESS NETWORK

The following are four easy steps to create a wireless network:

Step 1: Choose your wireless equipment. You should have a wireless router and network adapter. The router converts the signals coming across your internet connection into a wireless broadcast. Make sure it is a wireless router and not a wireless access point. Networkadapters wirelessly connect your computer to your wireless router. If you have a newer computer, you most likely already have a network adapter built in.

Computer networking is a great way to collaborate with other computer users in your home or office. While it is becoming increasingly easy for the basic computer user, it can still be a difficult, frustrating experience for many people.

Step 2: Second, connect your wireless router. Locate your cable modem or DSL modem and unplug it to turn it off. Next, connect your router to your modem. Next, plug in and turn on your cable or DSL modem. Wait a few minutes to give it time to connect to the Internet, and then plug in and turn on your wireless router. Your computer will wirelessly connect to your router, and the router will send communications through your modem to the Internet. After a minute, the WAN or WLAN light on your wireless router should light up, indicating that it has been connected successfully.

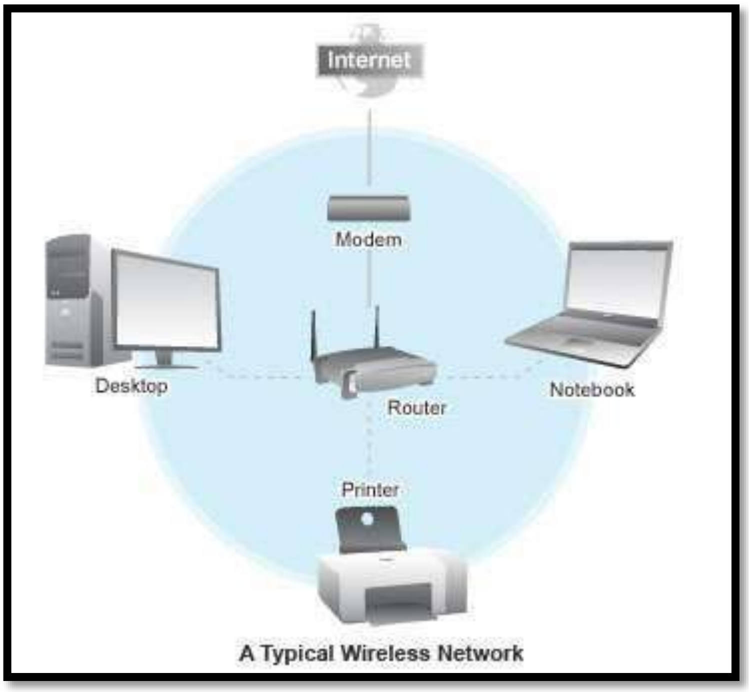


Figure 22: A typical Wireless Network

Step 3: Configure your wireless router. Using the network cable that came with your wirelessrouter, you should temporarily connect your computer to one of the open network ports on your wireless router. If you need to, turn your computer on and it should automatically connect to your router. Next, open Internet Explorer and type in the address to configure yourrouter.

Step 4: Connect your computers. If your computer does not have wireless network support built in, plug your network adapter into your USB port, and place the antenna on top of your computer (in the case of a desktop computer), or insert the network adapter into an empty PCcard slot (in the case of a laptop). Windows XP will automatically detect the new adapter, andmay require you to insert the CD that came with your adapter. The on-screen instructions willguide you through the configuration process.

Locate your cable modem or DSL modem and unplug it to turn it off. Next, connect your router to your modem. Next, plug in and turn on your cable or DSL modem. Wait a few minutes to give it time to connect to the Internet, and then plug in and turn on your wirelessrouter. Your computer will wirelessly connect to your router, and the router will send communications through your modem to the Internet. After a minute, the WAN or WLAN light on your wireless router should light up, indicating that it has been connected successfully.

6.8 CONFIGURATION OF WIRELESS ROUTER

Step 1: Plug the new wireless router into your internet connection point (filter/splitter if ADSL, directly into phone socket if DSL).

Step 2: If you have one, turn on your broadband connection and existing external modem FIRST (wait for all lights to return to normal).
Step 3: Plug the router into your PC with an Ethernet cable.
Step 4: Turn on your new wireless router Second.
Step 5: Go to your internet browser and type http://192.168.0.1/ (Belkin), http://192.168.1.1/(Linksys), http://192.168.2.1/ (Others) and enter your username and password for your router(often this is "admin" for username and "admin" or "password" for password) **Step 6:** Enable wireless capability (SSID) and enter your user name and password you got from your internet service provider.**Step 7:** Choose WPA* (or WEP if your card cannot handle WPA) security and enter a passkey and write it down.

6.9 WLAN

A wireless local area network (WLAN) is a wireless computer network that links two or moredevices using a wireless distribution method (often spread-spectrum or OFDM radio) within a limited area such as a home, school, computer laboratory, or office building. This gives usersthe ability to move around within a local coverage area and still be connected to the network,and can provide a connection to the wider Internet. Most modern WLANs are based on IEEE
802.11 standards.

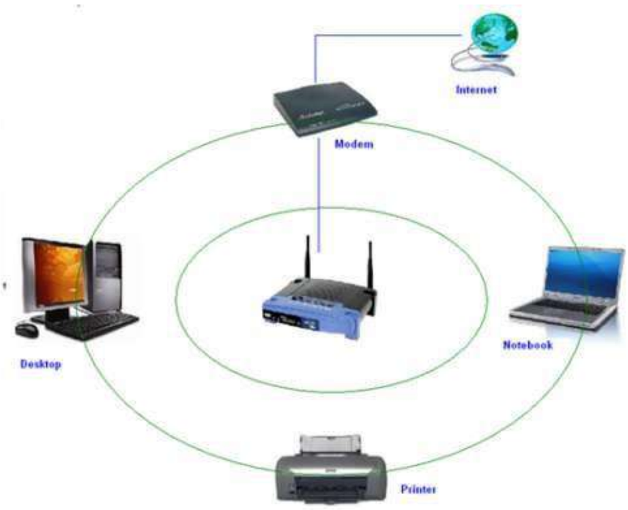


Figure 23: Wireless LAN

The other factors why WLANs are becoming more acceptable are:

1. No need to be connected physically with each other through any medium such as cables.You can roam around freely in office premises, home or around.
2. WLANs are cost effective. Cabling all the way in the offices, hotels etc. are not needed.So it’s cheap and provide same quality of service.
3. Unreachable spots where a cable is hardly accessible, WLAN signals can reach out such as big installations like airports. Also surfing outdoors is also convenient. Just install thedevice called Access Points (AP) and you are done.
4. Less interruption and easy trouble shooting in case of failures as compared to cablednetworks.
5. More secure as most of APs support best encryption methods which protect them fromsniffing and other attacks.

6.9.1 Major issues with WLAN

WLAN are also as prone to various attacks as their counterpart wired LANs are. Actually WLANs are easier to hack as compared to wired LANs, if not properly configured, due to itseasy accessibility around the installation. No need to be in contact of physical wires to hack, can be done from anywhere. Its convenience can turn into serious risk to the organization if not configured properly. Major attacks include such as, Sniffing, Key cracking, DoS (Denial of Service), De authentication attacks, War driving etc.

Wi-Fi at home

Wi-Fi at home is not a luxury anymore it has become a necessity. However, when the question of security comes into the scene, the first thought that would arise in my mind is how you can protect something which you cannot see, neither can you feel it?

Protecting a home wireless network is altogether a different side of the coin as compared to wired networks. Most of wireless network device vendor’s and Internet Service provider do not provide any security settings by default and leave the customer to find for herself. So make sure, your network is secured from being maliciously used. There is no silver bullet thatwill protect your wireless network infrastructure. These are, however, some countermeasureslisted below that should be used in conjunction with each other to secure your wireless network to the highest level:

1. **Use most secure possible encryptio**n: The first and most necessary step- use industry standard encryptions. The old (however generally used) WEP-Wired Equivalent Privacy, has been known to be broken. Even you use complex passwords it can be

   broken and decrypted within minutes or hours. WEP uses 40 bit or 128 bits RC4 ciphers to encrypt the channel. Instead use secure protocols such as WPA 2 – Wi-Fi Protected Access -2, which uses strong 128 bits AES ciphers and is typically considered more robust encryption strategy available.

2. **Use Firewall:** All the wireless routers come with built-in firewalls. Enable them withall the security features. You should block any anonymous ping requests and place restrictions on website browsing, if required. Define additional security policies and apply them.
3. **Have a monitoring system in place:** There's a saying- prevention is better than a cure. If you are able to detect some suspicious activities before it penetrates your network, you can block them or take precautionary measures. Deploy WIPS/WIDS for monitoring suspicious activities.
4. **Don't use default credentials:** Every wireless router comes with a set of default username/password. Sometimes, people don't change them and keep using them for long time. Username and passwords are used by computers or other devices to connect to wireless router. If any hacker is able to guess them, he can connect to yournetwork easily. Studies show that majority of users use the same combination of username/passwords as set by manufacturers. Some default username combinations are: admin/admin, admin/password or admin/ " ".
5. **Disable Auto-connect feature:** Some devices or the computers/laptops have 'Let thistool manage your wireless networks' or 'Connect automatically to available network'.Such users having this auto-connect feature enabled are prone to Phishing attack or Rogue AP attack. Attackers keep their APs alive and kicking for such kind ofunsuspecting users. They also use luring names as 'HotSpot', 'SecureConnect', 'GovtNetworks' etc. The user will never suspect them and keep surfing the wireless network happily. Also if you have not changed the default password of your router, the attacker will try to use this feature on their machine and automatically connect using the easily guessable default passwords.
6. **Don't use public Wi-Fi spots to surf sensitive websites:** Free and open wireless networks available on airports, cafes, railway stations are not very secure by nature. They do not use any encryption to secure the channel between your laptop to the router. So any information which is not by default going on HTTPS from your laptop/smart phone is susceptible to sniffing and even more your session could be hijacked because the unencrypted channel may leak the active session ID used by your website. All the attacker needs to do is to just install this tool in Firefox and start sniffing the communications on a public unencrypted Wi-Fi. Some applications like Facebook encrypts the login page [HTTPS] but internal pages are served on unencrypted [HTTP] channel so your session ID can be leaked
7. **Change the default SSID:** Although this will not prevent hackers breaking into a network, using a default SSID acts as an indication that the user is careless. So he maybe an obvious target to explore further to see if he still uses the default passwords as well?
8. **Restrict access by assigning static IP addresses and MAC filtering:** Disable automatic IP assigning feature and use private static IPs to the legitimate devices you

want to connect. This will help you in blocking unwanted devices from being connected to your network. Also, enable MAC filtering- router remembers MAC of each and every device connected to it and saves it as list. You can use this facility to restrict access. Only a set of trusted devices can be allowed to connect. However MAC spoofing is still possible but it raises an extra bar for your wireless network.

9. **Turn off your router when not in use:** Last but not least, a little obvious, but it will save your network from all the attacks for that time period.

6.10 SUMMERY

1. Wireless security is the prevention of unauthorized access or damage to computers using wireless networks.
2. Hackers have found wireless networks relatively easy to break into, and even use wireless technology to hack into wired networks.
3. Wireless Intrusion Prevention Systems (WIPS) or Wireless Intrusion Detection Systems (WIDS) are commonly used to enforce wireless security policies.
4. Service set identification (SSID) a series of 0 to 32 octets. It is used as a unique identifier for a wireless LAN. Since this identifier must often be entered into devices manually by a human user, it is often a human-readable string and thus commonly called the "network name".
5. Every wireless router has the ability to broadcast its name, or SSID (Service Set Identifier). Disabling SSID broadcasting means that your wireless network won’t appear in the list of “Available Wireless Networks” on any nearby computer.
6. Encryption is used to hide or mask the data being sent through wireless transmission.
7. WEP is a notoriously weak security standard. The password it uses can often be cracked in a few minutes with a basic laptop computer and widely available software tools.
8. WPA offers the benefits Enhancement data security, robust key management, Data origin authentication and Data integrity protection.
9. In computer networking, **MAC Filtering** refers to a

security access control method whereby the 48-bit address assigned to each network card is used to determine accessto the network.

10. MAC filtering is not an effective control in wireless networking as attackers can eavesdrop on wireless transmissions.
11. A wireless router is the most important piece of equipment that allows the internet to work.
12. Wi-Fi uses radio waves like other wireless devices such as laptops and cellular phones, and radios
13. A modem is a device that converts the digital signals from a computer into specific frequencies that can travel to television or telephone lines.
14. A wireless local area network (WLAN) is a wireless computer network that links twoor more devices using a wireless distribution method (often spread-spectrum or OFDM radio) within a limited area such as a home, school, computer laboratory, or office building.
15. WLAN are also as prone to various attacks as their counterpart wired LANs are.

6.11 CHECK YOUR PROGRESS

6.11.1 Fill in the blank

6.11.1.1 WPA2 uses an device that encrypts the network with a 256-bit key; the longer key length improves security over WEP.

6.11.1.2 WIPS stands for………………….

6.11.1.3 Wired Equivalent Privacy (WEP) is a security algorithm for………… wireless networks.

6.11.1.4 .... Service set identification (SSID) a series of 0 to .... octets.

6.11.1.5 MAC addresses are ...assigned to each card.

6.11.1.6 A wireless router is the

most important piece of equipment that allowsthe to work.

6.11.1.7 A modem is a device that converts the……………. from a computer into specific frequencies that can travel to television or telephone lines

6.12 <u>ANSWERS TO CHECK YOUR PROGRESS</u>

I. Encryption
II. Wireless Intrusion Prevention Systems
III. IEEE 802.11
IV. 32
V. Uniquely
VI. Internet
VII. Digital signals

6.13 <u>MODEL QUESTIONS</u>

6.13.1 What is wireless Security?

6.13.2 What is SSID? How do you disable SSID? Write step for that?

6.13.3 What is WEP, WPA and WPA2?

6.13.4 What is the difference between WPA and WPA2?

6.13.5 What are WPA and WPA2 Modes?

6.13.6 What is MAC filtering? How configure MAC address on wireless router.

6.14 What are 802.11 IEEE wireless Standards?

6.15 What is wireless router? How to create wireless network? Write steps for that.

6.16 How to configure wireless router? Write steps for that.

6.16.1 What is WLAN? What are major issues with WLAN?

6.16.2 What is firewall? Explain.

UNIT VII: INVESTIGATION AND DIGITAL FORENSIC

STRUCTURE

7.0 OBJECTIVES

After going through this unit, you will be able to:

- Define the meaning of word cyber and cyber crime
- Know the basics digital forensic
- Know various types of forensic investigation techniques
- Know the steps involved in forensic investigation

- Study about various forensic tools
- Define how the email works
- Differentiate between various protocols like SMTP, POP3,IMAP, etc
- Study different types of email abuses
- Perform fake emails and their senders investigation
- Recognize scams or hoax emails and websites
- Perform social media profile investigation

7.1 INTRODUCTION

Welcome to the Post-PC Era, the era dominated by truly portable devices with embedded GPS, near-field communication capabilities, microphones and HD video cameras and processing power equal to or greater than traditional computers[30]. We 'wear' our computers, we are empowered. Our dependency on cyber space grows by the day, our access points and networked devices are legion, and our cyber security systems are losing their foothold. Passwords, firewalls, security keys, encrypted memory sticks: traditional security measures no longer fully address the many sophisticated new challenges we face. To guarantee security and trustworthiness of data, industry, governments, businesses and citizens are to align their behaviors and interests to secure the growing digital-based economy and society. But what are the drivers behind this new cyber reality?

7.1.1 Drivers behind this new cyber reality?

a. The rise of bring your own device (BYOD): Employees are bringing their own tools to work – whether hardware, software or application. Information is on the move. The boundaries between "Internet at home" and "Internet at work" are quickly eroding with the advent of the ambient web: the Internet is always there and always on, connected to the devices and systems of your own choosing, whether you are aware of it or not. The overlap of professional and private lives raises new challenges for security: who secures your data, who is in control and from where?

b. An increasing reliance on cloud services: Cloud computing is still in an infancy stage, but a fundamental component of IT services orientation and consumerization,

c. heralding new avenues for producing, sharing and managing information. A truly globalized phenomenon, cloud computing is characterized by a services approach to satisfy the technology needs of organizations and individuals alike. Such is the new model in and for the IT-industry. Through this "hyper outsourcing", information and processes move beyond the company walls.

d. The Internet of things will accelerate operational dependency on the net: When all kinds of physical devices contain embedded networked software paving the way for 'the Internet of things' and so-called 'smart cities', cyber security becomes even more critical. The unique identification of citizens and objects in the virtual space may be some way off, yet we must still be prepared. The creation of an interconnected world and an ambient network will automatically create new possibilities for information breaches, system attacks and privacy violations.

e. The increasing sophistication of hackers, supported by advanced tools, will turn complex attacks into a simple click of a button: Amateurs turn professional, and the tools they use evolve accordingly – such is the common pattern to any popular and lucrative activity. The greater our dependency on the Internet and the more value we create from it, the smarter and more agile cyber attackers get. At the same time, hacking itself is becoming a commodity. For next to nothing you can buy the necessary tools for reading the SSID's of the wireless routers within 3km radius of you.

f. Regulatory constraints exercise a growing influence on information security: Protecting the interest of data stakeholders, be they clients, shareholders or private or public companies, is a raising issue for the regulators worldwide. For instance, communication about cyber attacks consequences will soon be legally binding and will lead organizations to ensure full transparent cyber security risk management. There is also increasing pressure on companies accepting credit cards payments to have their information system architecture certified by a third party, according to a standard (Payment Card Industry Data Security Standard) and to conduct penetration test campaigns with a frequency depending on payments volume.

7.1.2 Cyber Crime and Challenges Ahead

The word "cyber" is derived from the Greek term "cybernetic" meaning "skilled in steering or governing". Securing the virtual space and protecting networked assets, is of critical importance now that digital goods and services permeate all levels of the economy and civil society. The growing importance of this space for business and our personal lives requires that we take all necessary measures to secure it.

With the growing incidence of cyber crime, and the increased adoption of digital devices, digital forensics has gained significant importance in the recent past, augmenting what was conventionally limited to the recovery and analysis of biological and chemical evidence during criminal investigations[31].INVESTIGATION TECHNIQUES & COMPUTER FORENSICS

Forensics[32] is a discipline that dates back at least to the Roman era (and possibly event to ancient China), when people accused of crimes (and the accuser) presented evidence in front of a public audience (the Latin word forensics, means "of or before the forum"). In modern times it has come to mean the application of scientific processes to recover evidence related to crime or other legal action.

7.1.3 Digital Forensics System

Digital forensics, as a discipline, grew out of the explosion in personal computer use during the late 1970s and early 1980s. The first specific computer crimes were recognized in the 1978 Florida Computer Crimes Act, which included legislation against the unauthorized modification or deletion of data on a computer system. Over the next few years the range of computer crimes being committed increased and laws were passed to deal with issues of copyright, privacy/harassment and child pornography.

It was not until the 1980s that federal laws began to incorporate computer offences.

Canada was the first country to pass legislation in 1983. This was followed by the US Federal Computer Fraud and Abuse Act in 1986, Australian amendments to their crimes acts in 1989 and the British Computer Abuse Act in 1990.

Much of the forensic analysis during this period was performed on "live" systems, using traditional (and non-specialist) system administration tools. Very few standards or guidelines existed to help practitioners, and the evidence they produced was often rejected by courts.

Digital forensics is traditionally associated with criminal investigations and, as you would expect, most types of investigation centre on some form of computer crime. This sort of crime can take two forms; computer based crime and computer facilitated crime.

Computer based crime
This is criminal activity that is conducted purely on computers, for example cyber-bullying or spam. As well as crimes newly defined by the computing age it also includes traditional crime conducted purely on computers (for example, child pornography).

Computer facilitated crime
Crime conducted in the "real world" but facilitated by the use of computers. A classic example of this sort of crime is fraud: computers are commonly used to communicate with other fraudsters, to record/plan activities or to create fraudulent documents.

Not all digital forensics investigations focus on criminal behaviour; sometimes the techniques are used in corporate (or private) settings to recover lost information or to rebuild the activities of employees.

7.2 TYPES OF INVESTIGATION

There are four main types of investigation performed by digital forensics specialists. The first three are broadly similar in the activities involve, but differ in terms of the legal restrictions and guidelines imposed as well as the type of digital evidence and form of report.

7.2.1 Criminal forensics

The largest form of digital forensics and falling under the remit of law enforcement (or private contractors working for them). Criminal forensics is usually part of a wider investigation conducted by law enforcement and other specialists with reports being intended to facilitate that investigation and, ultimately, to be entered as expert evidence before the court. Focus is on forensically sound data extraction and producing report/evidence in simple terms that a lay man will understand.

7.2.2 Intelligence gathering

This type of investigation is often associated with crime, but in relation to providing intelligence to help track, stop or identify criminal activity. Unless the evidence is later to be used in court forensic soundness is less of a concern in this form of investigation, instead speed can be a common requirement.

7.2.3 Electronic discovery (eDiscovery)

Similar to "criminal forensics" but in relation to civil law. Although functionally identical to its criminal counter part, eDiscovery has specific legal limitations and restrictions, usually in relation to the scope of any investigation. Privacy laws (for example, the right of employees not to have personal conversation intercepted) and human rights legislation often affect electronic discovery.

7.2.4 Intrusion investigation

The final form of investigation is different from the previous three. Intrusion investigation is instigated as a response to a network intrusion, for example a hacker trying to steal corporate secrets. The investigation focuses on identifying the entry point for such attacks, the scope of access and mitigating the hackers activities. Intrusion investigation often occurs "live" (i.e. in real time) and leans heavily on the discipline of network forensics.

7.3 EVIDENCE AND ANALYSIS

Obviously the main aim of any investigation is to recover some form of digital evidence, objective data that is relevant to the examination. On top of that the investigator might be asked to make some form of analysis of that evidence; either to form an expert conclusion, or to explain the meaning of the evidence. Here are some examples of the kind of analysis an examiner might be asked to undertake:

Attribution

Meta data and other logs can be used to attribute actions to an individual. For example, personal documents on a computer drive might identify its owner.

Alibis and statements

Information provided by those involved can be cross checked with digital evidence.

Intent

Intent as well as finding objective evidence of a crime being committed, investigations can also be used to prove the intent (known by the legal term mens rea).

Evaluation of source

File artifacts and meta-data can be used to identify the origin of a particular piece of data. for example, older versions of Microsoft Word embedded a Global Unique Identifer into files which identified the computer it had been created on. Proving whether a file was produced on the digital device being examined or obtained from elsewhere (e.g., the Internet) can be very important.

Document authentication

Related to "Evaluation of Source", meta data associated with digital documents can be easily modified (for example, by changing the computer clock you can affect the created date of a file). Document authentication relates to detecting and identifying falsification of such details.

7.4 STEPS IN FORENSICS INVESTIGATION

A digital forensic investigation generally consists of five steps:

1. Identification: Identify the system to be investigated.

2. Data acquisition/data preservation: taking images of the drives/partition belonging to the identified system.
3. Data recovery: recover deleted data from the image file.
4. Analysis: of data for evidence: analyze digital artefacts inside the data for evidences.
5. Reporting: of the digital evidence found: reporting of evidences found during analysis phase.

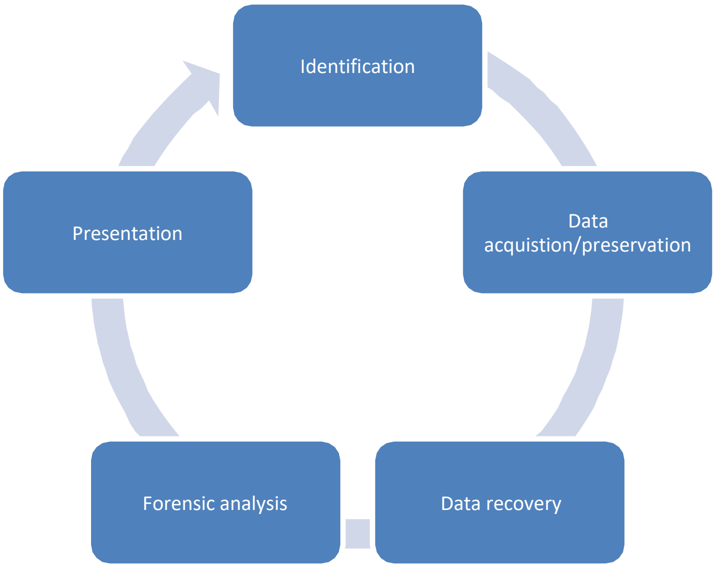


Figure 24: Digital forensic investigation process

7.5 FORENSICS TOOLS

In the early days of digital forensics analysts had to make do with existing system administration or information security tools. Plenty of these existed, but they were not particularly suited to the more formal approach of a forensic investigation. In particular much of the software required you to run it on the live system, which introduced all manner of problems with modifying evidence.

During the 1980s and 90s, however, increased funding and interest in the field encouraged the development of a variety of specialist commercial and freeware tools. These can generally be broken down into three categories:

- **General forensic tools:** Tools allowing a wide variety of investigation, particularly keyword searching, on digital media.
- **Specialist forensic tools:** Which focus on a specific piece of forensic material for investigation - perhaps images, or internet artifacts. Often relying on output from one of the general tools.
- **Case Management tools** These are used to track, audit and report on cases

In addition there is a "fourth" category of **useful software**, a normal piece of software which can usefully be adapted for use in a forensic investigation. The next section makes mention of several commercial tools. This is not an endorsement of the tools, they are intended to serve as examples to explore

General forensic tools

Many of these tools are complex, commercially produced, and come with enterprise price tags (in the region of thousands of dollars a year). The majority of commercial tools run on Windows whilst free tools tend to run on Linux. Later on we will discuss the ways digital media can be investigated in more depth, but for the moment it is important to understand that general forensic software is usually centred around the act of keyword searching across a piece of digital media. The two most common ways of performing such searches is "live search" (where the digital media is parsed for a set of keywords and bookmarks of hit locations is stored) and "indexing" (where a text index of the digital media is created, allowing searches to be performed quickly using the index). Both styles have advantages and disadvantages.

The "de facto" industry standard tool is usually considered to be EnCase, produced by Guidance Software. It is a general forensics tool tailored for windows systems and focuses on the live search method. It includes a scripting interface, dubbed EnScript, which is useful for developing custom tools to extract information. EnCase is closely followed by Access Data's Forensic Toolkit (or FTK). FTK focuses on indexing media and is often used in cases where large volumes of data are being investigated, or where a large list of keywords needs to be searched.

Specialist forensic tools

Specialist tools focus on a particular aspect of forensic investigation; for example categorising images or recovering internet artefacts. The range of tools and software is vast, including commercial and free offerings.

One of the better known is a free tool called "Categoriser 4 Pictures" which is a helper tool for classifying images and presenting your results. C4P is a class of tool that relies on output from EnCase, using an EnScript to parse and extract images for processing.

Another common theme for specialist tools is internet artefacts; this can range from recovering internet cache data (web pages and other fragments) to analysing internet history or recovering chat transcript. Internet artefacts often contain a large amount of useful evidence and it is a common focus for investigations. Some notable tools include:

- Netanalysis; commercial tool, parses internet history files (.dat) and allows searching/analysis of the data.
- Internet Evidence Finder; commercial, scans digital media for a variety of internet artefacts (i.e. chat, webmail and internet history)
- Virtual Forensic Computing; allows digital media containing an operating system to be mounted as a virtual machine.

Case management

We already touched on case management in "Documenting evidence", but it is included here for completeness. Very few (if any) software tools exist for complete case management (although some practitioners adapt case management tools from the law field). Several free cases note tools exist for creating audit-able notes; the primary example being CaseNotes. Many analysts still use paper documents, partly because this is an audit trail that courts understand and accept!

Useful software

A wide variety of tools exist that are adaptable for forensic investigation; system administration tools, for example, can often tell you a lot about a system. VMWare is a commercial/free tool that can be used to view digital media as virtual machines. VLC media player can be useful for handling a diverse collection of media.

7.6 INVESTIGATION

Today, nearly all abusive e-mail messages carry fake sender addresses[33]. The victims whose addresses are being abused often suffer from the consequences, because their reputation gets diminished and they have to disclaim liability for the abuse, or waste their time sorting out misdirected bounce messages.ou probably have experienced one kind of abuse or another of your e-mail address yourself in the past, e.g. when you received an error message saying that a message allegedly sent by you could not be delivered to the recipient, although you never sent a message to that address.

Sender address forgery is a threat to users and companies alike, and it even undermines the e- mail medium as a whole because it erodes people's confidence in its reliability. That is why your bank never sends you information about your account by e-mail and keeps making a point of that fact.

In the following sections we will touch upon how email works and techniques to find fake email. But before that, we are going to introduce some important terminologies which we are going to use throughout.

Some important terminologies

1. **Protocol:** Protocols are the set of rules and procedures governing the transmission of data over network.
2. **Internet Domains:** Internet domain names are the alphanumeric identifiers we use to refer to hosts on the Internet, like "LivingInternet.com". Internet domain names come in four main types[34]:

   - top-level domains
   - second-level domains
   - third-level domains, and
   - country domains.

   a. **Top-level domains**. Internet domain names are organized by their levels, with the higher levels on the right. For example, for the domain "mail.twenty.net" the top- level domain is "net", the second-level domain is "twenty.net", and the third-level domain is "www.twenty.net". Some of the popular top level domain

names are
.com, .net, .biz, .gov. etc/

b. **Second-level domains**. Top-level Internet domains like ".com" are shared by all the organizations in the domain. Second-level domain names like "yahoo.com" and "livinginternet.com" are registered by individuals and organizations. Second- level domains are the addresses commonly used to host Internet applications like web hosting and email addressing.

c. **Third-level domains**. Third-level Internet domain names are created by those that own second-level domains. Third-level domains can be used to set up individual domains for specific purposes, such as a domain for web access and one for mail, or a separate site for a special purpose:

- www.livinginternet.com
- mail.livinginternet.com
- rareorchids.livinginternet.com

d. **Fourth-level domains**: Fourth-level domains and even higher Internet domains like "www2.un.news.media.twenty.net" can be also be established. Three of four levels is usually sufficient for most purposes. Each country in the world has its own top-level Internet domain with a unique alphabetic designation, known as country domains. A few countries and example domains are: .ae for United Arab Emirates, .in for India, .au for Australia, .us for United States, etc.

3. **The Domain Name System (DNS):** DNS as a whole consists of a network of servers that map Internet domain names like www.livinginternet.com to a local IP addresses. The Domain Name System (DNS) servers distribute the job of mapping domain names to IP addresses among servers allocated to each domain. Each second-level domain must have at least one domain name server responsible for maintenance of information about that domain and all subsidiary domains, and response to queries about those domains from other computers on the Internet. For example, management of domain name information and queries for the LivingInternet.com domain is handled by a specific DNS server that takes care of the load required. This distributed architecture was designed to enable the Internet to grow, where as the number of domains grew, the number of DNS servers can grow to keep pace with the load. Today, everyone who registers a second-level domain name must at the same time designate two DNS servers to manage queries and return the current IP address for addresses in that domain. The primary domain name server is always consulted first, and the secondary domain name server is queried if the primary doesn't answer, providing a backup and important support to overall Internet reliability.

When your computer tries to access a domain like "www.livinginternet.com",
the domain name system works like this:

- Your computer asks your default DNS server if it knows the IP address for www.livinginternet.com. If the DNS server has been asked that question recently, then it will have the answer stored in its local cache, and can answer immediately.

- Otherwise, your DNS server queries the central zone files for the address of the primary domain name server for livinginternet.com, and is answered with something like "ns1.livinginternet.com".

- Your DNS server will ask the livinginternet.com DNS server for the IP address of www.livinginternet.com, which will then look up the answer and send it back.

- Your DNS server will store the IP address returned in its local cache, and make the address available to your computer.
- Your computer then contacts www.livinginternet.com with the standard Internet routing protocols by using the returned IP address.

4. **Email servers:** Each Internet domain has an associated email server that manages all email addresses at that domain. Each email address is expressed in the form "name@domain" and is unique at that domain, as in for example "jane@twenty.net".
5. **IP Address:** Every computer on the Internet has a unique numerical address, called an Internet Protocol (IP) address, used to route packets to it across the Internet. Just as your postal address enables the postal system to send mail to your house from anywhere around the world, your computer's IP address gives the Internet routing protocols the unique information they need to route packets of information to your desktop from anywhere across the Internet. If a machine needs to contact another by a domain name, it first looks up the corresponding IP address with the domain name service. The IP address is the geographical descriptor of the virtual world, and the addresses of both source and destination systems are stored in the header of every packet that flows across the Internet. You can find your IP address on a Windows computer by opening an MSDOS or Command window and typing one of "winipcfg" or "ipconfig". You can find your IP address on a Mac computer by checking your Network control panel. An IP address is made up of four bytes of information (totaling 32 bits) expressed as four numbers between 0 and 255 shown separated by periods. For example, your computer's IP address might be 238.17.159.4, which is shown below in human-readable decimal form and in the binary form used on the Internet.

Example IP Address	
Decimal:	238 . 17 . 159 . 4

Binary:	11101110 00010001 10011111 00000100

Each of the four numbers uses eight bits of storage, and so can represent any of the 256 numbers in the range between zero (binary 00000000) and 255 (binary 11111111). Therefore, there are more than 4 billion possible different IP addresses in all:

4,294,967,296 = 256 * 256 * 256 * 256

6. **Email client:** Your email client application communicates with an email server over the Internet to login, get mail status, and send and receive email. The most common email client are Internet Explorer, Morzila, Crome, etc.
7. **POP3:** Your email client talks to your email server to send it commands to login, get mail status, and send and receive email. The most common protocol used by email clients to communicate with email servers is the Post Office Protocol. POP3 has become the most common email client connection protocol. The POP3 protocol enables any email program anywhere on the Internet to connect to any email server to perform the usual email functions, such as reading and sending, as long as they have a valid account and password.
8. **The Internet Message Access Protocol (IMAP):** is a less common but more richly featured email protocol than POP3. IMAP is a more modern protocol than POP3, first invented at Stanford University in 1986. The current version is IMAP4, providing similar services to the POP3 protocol, but with additional features. The IMAP features can be useful in several situations, for example when you are travelling and don't want to download your email onto a laptop because then you won't have them on your home computer when you get back. It can also be useful for use on low-bandwidth devices like personal digital assistants, enabling you to select a few email from a list of subject headers before downloading just the ones you want.
9. **Messaging Application Programming Interface (MAPI)**: is a Microsoft Windows specific email interface.
10. **Simple Mail Transfer Protocol** (**SMTP**): is an Internet communication protocol used to send and relay an email message between email servers. It is not used to retrieve email messages from a server. Instead either IMAP or POP is used to retrieve email messages.

7.7 HOW EMAIL WORKS

Email is based around the use of electronic mailboxes[35]. When an email is sent, the message is routed from server to server, all the way to the recipient's email server. More precisely, the message is sent to the mail server tasked with transporting emails (called the **MTA**, for Mail Transport Agent) to the recipient's MTA. On the Internet, MTAs communicate with one another using the protocol SMTP, and so are logically called **SMTP servers** (or sometimes outgoing mail servers).

The recipient's MTA then delivers the email to the incoming mail server (called the **MDA**, for Mail Delivery Agent), which stores the email as it waits for the user to accept it. There are two main protocols used for retrieving email on an MDA:

- POP3 (Post Office Protocol), the older of the two, which is used for retrieving email and, in certain cases, leaving a copy of it on the server.
- IMAP (Internet Message Access Protocol), which is used for coordinating the status of emails (read, deleted, moved) across multiple email clients. With IMAP, a copy of every message is saved on the server, so that this synchronization task can be completed.

For this reason, incoming mail servers are called **POP servers** or **IMAP servers**, depending on which protocol is used.

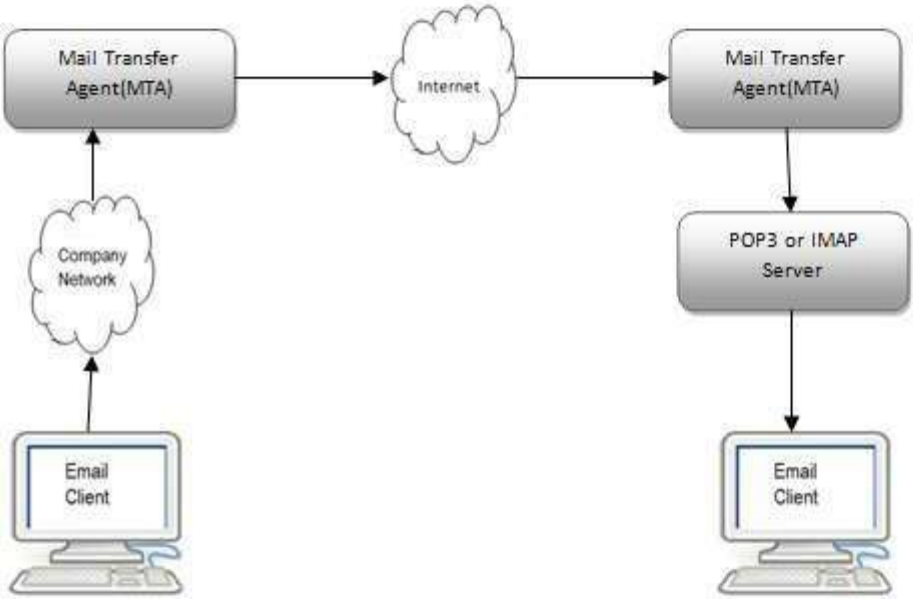


Figure 25: Working of an email

To use a real-world analogy, MTAs act as the post office (the sorting area and mail carrier, which handle message transportation), while MDAs act as mailboxes, which store messages (as much as their volume will allow) until the recipients check the box. This means that it is not necessary for recipients to be connected in order for them to be sent email.

To keep everyone from checking other users' emails, MDA is protected by a user name called a **login** and by a **password**.

Retrieving mail is done using a software program called an **MUA** (Mail User Agent).

When the MUA is a program installed on the user's system, it is called an **email client** (such as Mozilla Thunderbird, Microsoft Outlook, Eudora Mail, Incredimail or Lotus Notes).

7.8 COMMON TYPES OF E-MAIL ABUSE WHERE THE SENDER

ADDRESS IS FORGED

- **Spammers:** want to avoid receiving non-delivery notifications (bounces) to their real addresses.
- **Fraudsters:** want to cover their tracks and remain anonymous.
- **Computer worms:** want to cause confusion or just don't care about which sender addresses they use.
- **Phishers (password fishers):** want to impersonate well-known, trusted identities in order to steal passwords from users.

Sender Addresses in E-Mails

Like paper mail letters, e-mail messages have at least two kinds of sender addresses: one on the envelope and one in the letterhead.

- The **envelope sender address** (sometimes also called the return-path) is used during the transport of the message from mail server to mail server, e.g. to return the message to the sender in the case of a delivery failure. It is usually not displayed to the user by mail programs.

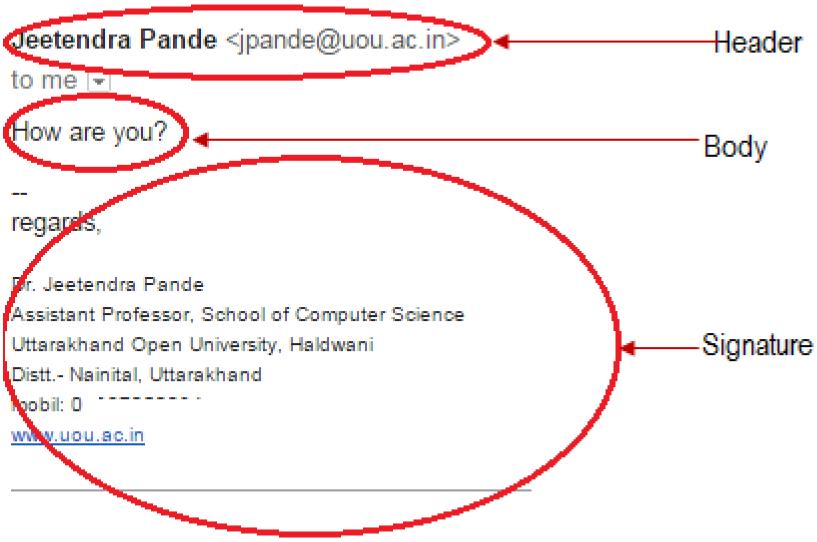
- The **header sender address** of an e-mail message is contained in the "From" or "Sender" header and is what is displayed to the user by mail programs. Generally, mail servers do not care about the header sender address when delivering a message.

Parts of an email

An email consists of three parts:

- Header
- Body
- Signature

An email structure is explained using the figure below.



Jeetendra Pande <jpande@uou.ac.in>
to me

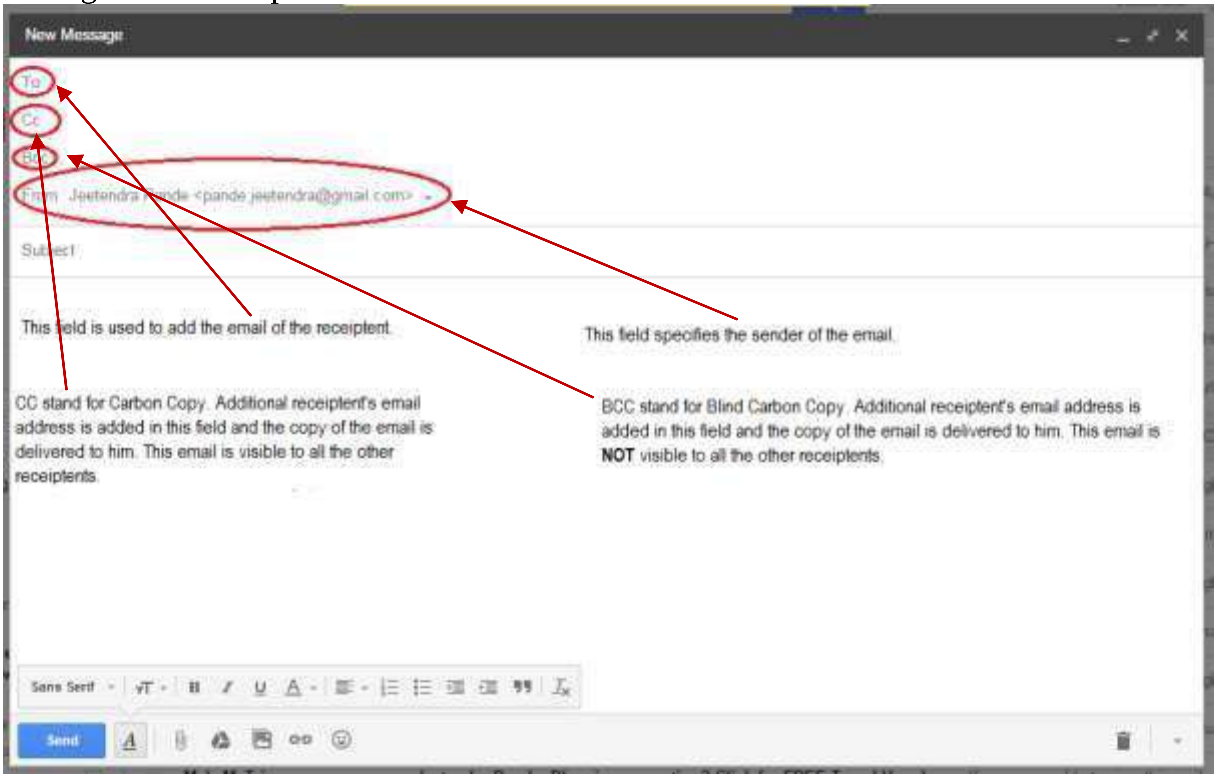
How are you?

--
regards,

Dr. Jeetendra Pande
Assistant Professor, School of Computer Science
Uttarakhand Open University, Haldwani
Distt.- Nainital, Uttarakhand
Mobil: 0
www.uou.ac.in

Figure 26: Different parts of an email

The figure below explains the different fields of an email.



This field is used to add the email of the receiptent

This field specifies the sender of the email.

CC stand for Carbon Copy. Additional receiptent's email address is added in this field and the copy of the email is delivered to him. This email is visible to all the other receiptents

BCC stand for Blind Carbon Copy. Additional receiptent's email address is added in this field and the copy of the email is delivered to him. This email is **NOT** visible to all the other receiptents

Figure 27: Fields of an email

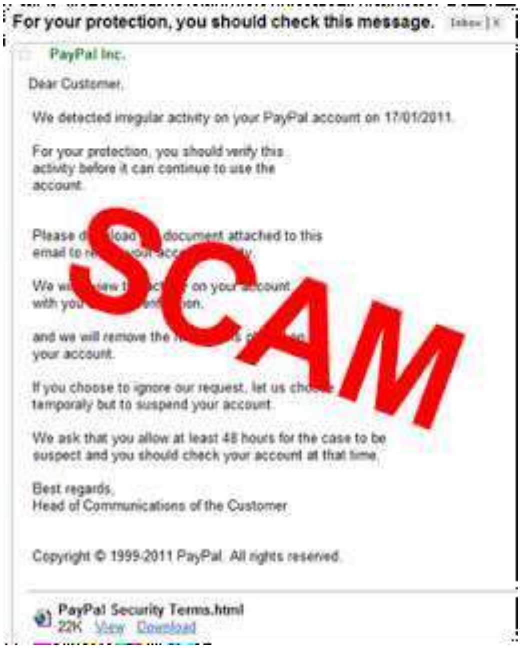
One of the most important and useful applications of the internet is undoubtedly the sending and receiving of emails[36]. Apart from their almost instant nature they are also effectively free. Unfortunately this also makes email an ideal medium of abuse, in the sense that because it is free to send an email, it is also free to send a million emails. The major ISP's estimate that of the billions of emails being sent and receiving, about 50% of them are spam (or UCE – Unsolicited Commercial Email). For many people who have the same email address for a period of years, the amount of spam can be counted in thousands a day.

Then next section will discuss how to identify fake mails and procedure to identify the origin of a fake email.

How to Identify Fake EMail And Trace Sender's Location[37]

Fake Emails

Fake emails or fake mails are those emails which pretend to come from a specific email address but are sent from some fake email senders. It is not hard to send fake email address. Anyone can use those free fake email sender tools available online. These tools ask for the name and email of the sender which you want to show in the email.



For your protection, you should check this message.

PayPal Inc.

Dear Customer,

We detected irregular activity on your PayPal account on 17/01/2011.

For your protection, you should verify this activity before it can continue to use the account.

Please d[illegible]load [illegible] document attached to this email to r[illegible]our acc[illegible]y.

We wi[illegible]iew t[illegible]c[illegible] on your account with you[illegible]on.

and we will remove the [illegible]s [illegible] your account.

If you choose to ignore our request, let us cho[illegible] temporaly but to suspend your account.

We ask that you allow at least 48 hours for the case to be suspect and you should check your account at that time.

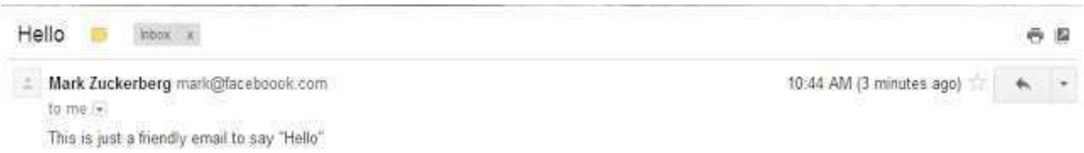
Best regards,
Head of Communications of the Customer

Copyright © 1999-2011 PayPal. All rights reserved.

PayPal Security Terms.html
22K View Download

Figure 28: A sample of fake email[8]

So you must know how to identify whether the email is fake or not. See the sample email snapshot:



Hello

Mark Zuckerberg mark@facebook.com
to me
This is just a friendly email to say "Hello"

10:44 AM (3 minutes ago)

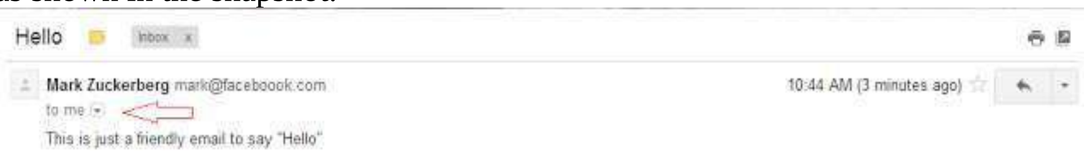
Figure 29:Sample spam email

See the sender's name and email. It's **Mark Zuckerberg** from email address **mark@facebook.com**.

This email is just an example. But these can be very harmful when sent by spammers. Suppose it pretends to be sent from your bank and asks for your banking username or password. There are so many examples which show why you must know about fake emails.

3.10.3.1 How to Identify Fake Email

It is really simple to identify a fake email. Click on the down arrow at the right side of the Me as shown in the snapshot.



Hello Inbox x

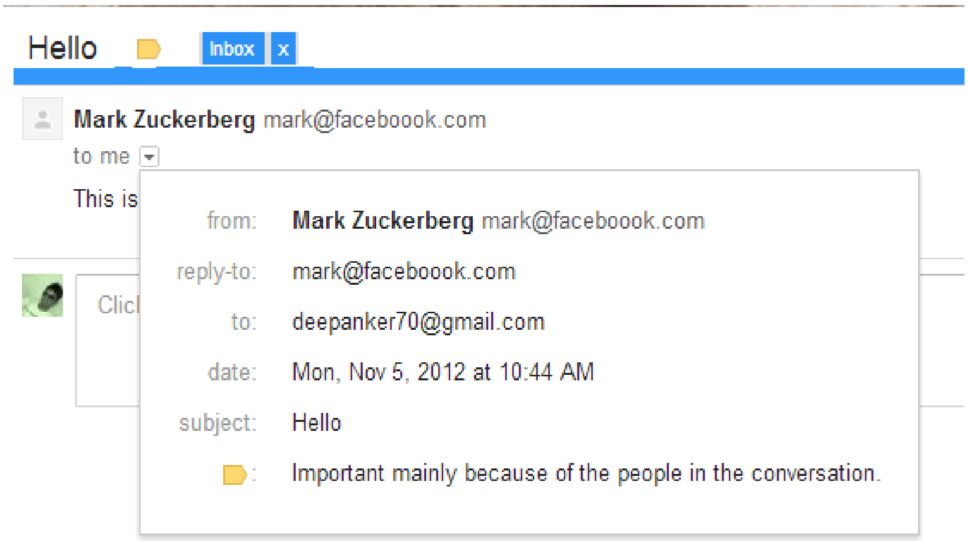
Mark Zuckerberg mark@faceboook.com 10:44 AM (3 minutes ago)

to me

This is just a friendly email to say "Hello"

Figure 30: Procedure to find out the details of the sender

You will see something like this:



Hello Inbox x

Mark Zuckerberg mark@faceboook.com

to me

This is

Clic

from: Mark Zuckerberg mark@faceboook.com

reply-to: mark@faceboook.com

to: deepanker70@gmail.com

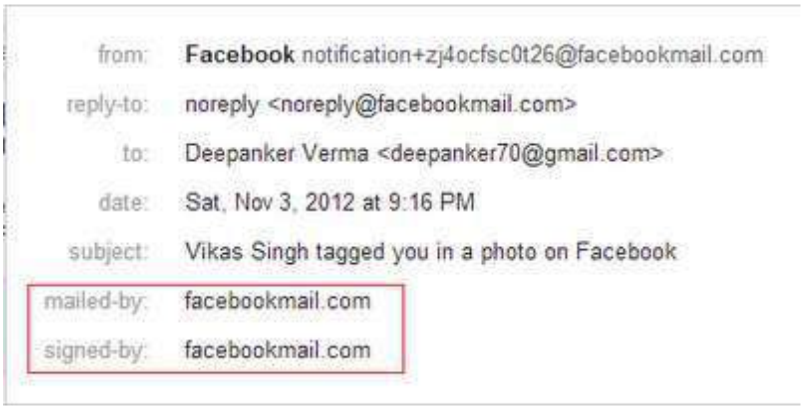
date: Mon, Nov 5, 2012 at 10:44 AM

subject: Hello

: Important mainly because of the people in the conversation.

Figure 31: Finding sender’s details

Here you will see some details about the email. If the email comes from a trusted source, you will be able to see two more fields, **Mailed By** and **Signed By**. See the snap below:



from: Facebook notification+zj4ocfsc0t26@facebookmail.com

reply-to: noreply <noreply@facebookmail.com>

to: Deepanker Verma <deepanker70@gmail.com>

date: Sat, Nov 3, 2012 at 9:16 PM

subject: Vikas Singh tagged you in a photo on Facebook

mailed-by: facebookmail.com

signed-by: facebookmail.com

Figure 32: Investigating mailed by and signed by fields

This method can be applied only for the emails coming from big companies like Google, Facebook, LinkedIn, Twitter and other companies which have their own servers. Email sent from Gmail will be mailed by and signed by Gmail. But there are so many small companies that does not have dedicated server. They use Gmail labs or their own hosting server. Email

coming from those may not show these two fields in mail. So we need to confirm this by one more way.

Now we will see the header of email. To see the header of email, click on down arrow at the right side of the reply icon and click on show original. Now it will open plain text email content with header information in a new tab.

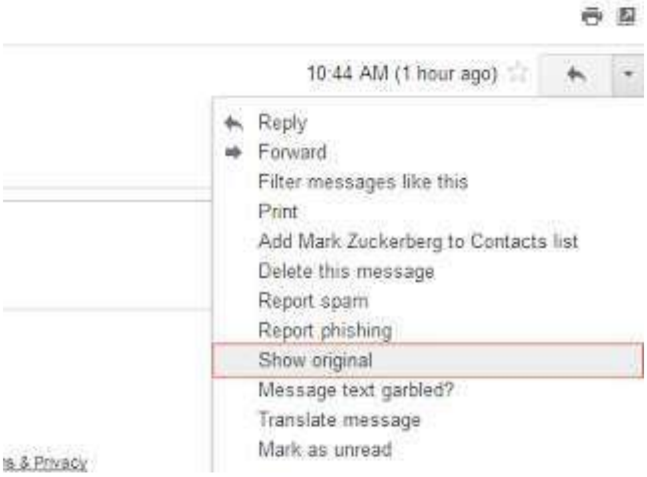
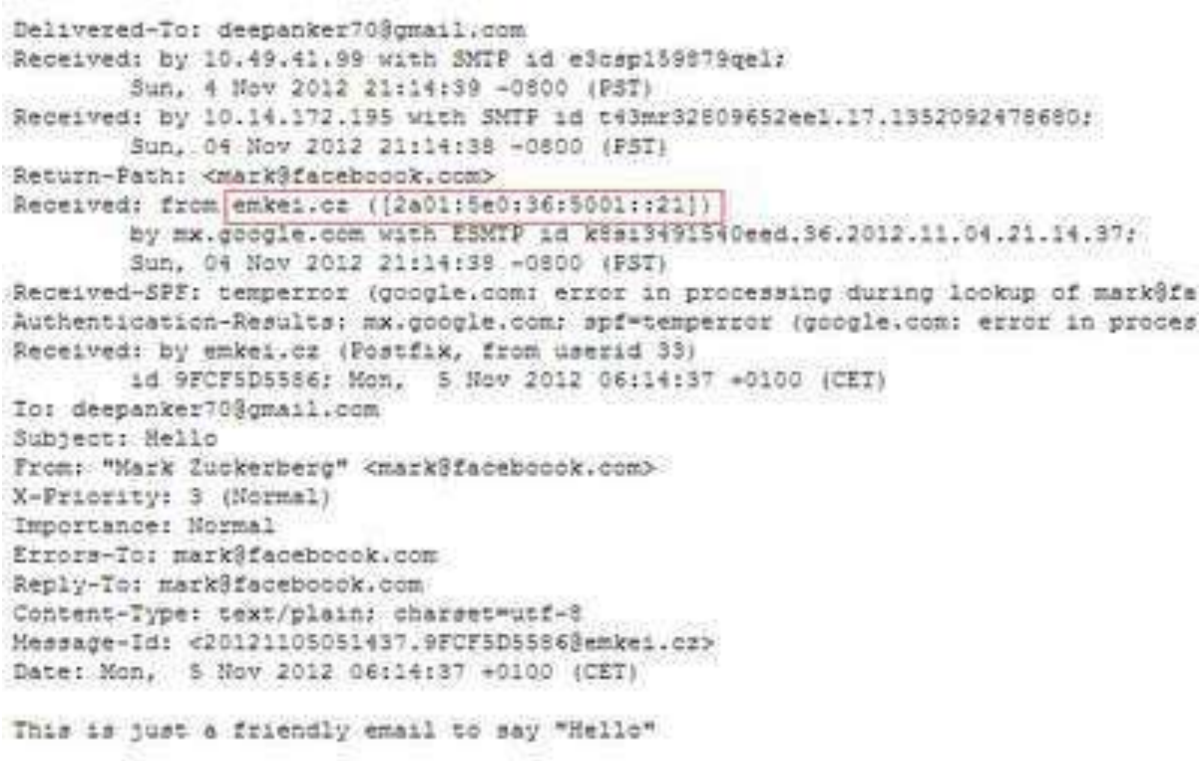


Figure 33: Procedure to find the headers of an email

Header information looks like this. Search for **Received: from** in this page. If there are more than one Received: from in the page, the go for the last one and see the domain there.



```
Delivered-To: deepanker70@gmail.com
Received: by 10.49.41.99 with SMTP id e3csp159879qel;
        Sun, 4 Nov 2012 21:14:39 -0800 (PST)
Received: by 10.14.172.195 with SMTP id t43mr32809652eel.17.1352092478680;
        Sun, 04 Nov 2012 21:14:38 -0800 (PST)
Return-Path: <mark@faceboook.com>
Received: from emkei.cz ([2a01:5e0:36:5001::21])
        by mx.google.com with ESMTP id k8si3491540eed.36.2012.11.04.21.14.37;
        Sun, 04 Nov 2012 21:14:38 -0800 (PST)
Received-SPF: temperror (google.com: error in processing during lookup of mark@fa
Authentication-Results: mx.google.com; spf=temperror (google.com: error in proces
Received: by emkei.cz (Postfix, from userid 33)
        id 9FCF5D5586; Mon,  5 Nov 2012 06:14:37 +0100 (CET)
To: deepanker70@gmail.com
Subject: Hello
From: "Mark Zuckerberg" <mark@faceboook.com>
X-Priority: 3 (Normal)
Importance: Normal
Errors-To: mark@faceboook.com
Reply-To: mark@faceboook.com
Content-Type: text/plain; charset=utf-8
Message-Id: <20121105051437.9FCF5D5586@emkei.cz>
Date: Mon,  5 Nov 2012 06:14:37 +0100 (CET)

This is just a friendly email to say "Hello"
```

Figure 34: Headers of an email

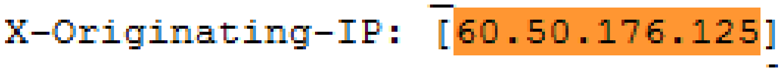
It shows emkei.cz in the fake mail sent by me. Now see the website **emkei.cz** and you will know that the domain belongs to a fake mail sender website.

If you see the header of some other emails, you will see that header comes with too many information which are not present in the header of this fake mail.

7.9 HOW TO TRACE LOCATION OF EMAIL SENDER

Email address can be traced or not. It depends on the mail server it has been sent. If the email is sent from Gmail using GMail web, you will never get the original IP address of the sender. Some other email servers (It may be a fake mail sender) also do not reveal the IP address of the sender in the email header. But in most of the cases (Other than sender is gmail) you can easily get the IP address of the person.

To get the IP address of the sender in the email header, search for **X-originating-IP:** and you will get the IP address of the sender.



```
X-Originating-IP: [60.50.176.125]
```

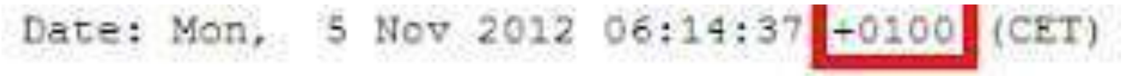
Now see the header of fake mail added above as snap by me, you will not find this field. It means it does not reveal the IP address of sender.
If you get the IP address, now you can use any IP tracer online tool to get the IP address. Use WhatIsMyAddress Ip Lookup tool available at https://www.whatismyip.com/ip-

address- lookup/

Note: If a person is using dialup connection with dynamic IP, IP Tracing will only trace up to the IP address of the ISP. For getting exact location, you need to contact ISP which is impossible without the permission of cyber police. If a person has purchased a dedicated IP connection, you will get the exact location of it.

What to do if IP is not there or Email is sent from Gmail

If you are not able to get the IP address of the person, At least you can know the country of the email. Search for Date: and at the end of line, see the time zone:



```
Date: Mon, 5 Nov 2012 06:14:37 +0100 (CET)
```

Here the time zone is +0100. Treat it as +01:00. Although, there will be so many countries belongs to a timezone, but you may get an approx idea.

7.10 RECOGNISE SCAM OR HIAX EMAIL AND WEBSITES

Scam and hoax websites and emails are designed to:

- trick you into disclosing personal information such as bank account details, passwords or credit card numbers.
- con you into paying money for fake get-rich-quick offers, prizes or lottery wins, or fraudulent or poor quality goods.

Be suspicious of emails from people or businesses you don't know, particularly if they promise you money, good health or a solution to all your problems.

Be suspicious of unexpected emails from your bank or financial institution. Remember banks don't do business via email and never ask for confidential information via email. Scammers put a lot of time and money into making hoax emails and bogus websites look real. Don't be fooled.

Scams

There is a huge range of scams on the internet, from promises of non-existent free products in return for clicking through to a website (which make advertising profits), to sophisticated targeted scams which can run for years and cost victims hundreds of thousands of dollars.

Table 4: Some common scams

Scam	Delivery	Characteristics
419 or 'Nigerian letter' advance fee These messages are sent to thousands of people on the probability that some will believe the story and forward the advance fee.	Email (or letter) claiming to be from a solicitor, barrister, public official or some other authoritative person.	The writer is in another (usually underdeveloped) country. They own of a huge sum of money, but need the help of a foreign partner (you) in order to access it. Help them access the money and you will receive a share of it. You 'wire' them a small advance fee for some contrived reason (for example clearance fees, tax).

Dating and romance They attempt to enter into an online relationship with the victim in order to persuade them to forward money or divulge personal details.	Fake profiles on dating sites or responding to profiles with fake personas A friend request or message in social media or a communications service such as Skype.	Their profile pictures look professional and might have been cut and pasted from a website or magazine. They attempt to become intimate quickly. They may say they live close by, but are unable to arrange a meeting or suddenly have to travel overseas. There are signs the correspondence might be cut and pasted or taken from template, for example wrong names, inconsistent or disjointed grammar. Requests for money accompanied by elaborate scenarios and effusive language.

Generally, any offer that promises a large reward for a small fee is almost certainly a scam.

Steps to avoid online scams and hoaxes

Delete suspicious emails and leave websites that:

- ask you to provide your banking details or personal information
- promise you money
- present hard luck or exotic stories telling you that you can share in hidden millions of dollars
- offer jobs where you need no qualifications, but just ask for a bank account for money transfers.

Points to Remember

- Never provide personal details via emails or links from emails. If you are unsure, double check by telephone with the company or institution.
- Never follow the links in spam emails; these could lead to downloading unwanted

viruses, spyware or malware.

- Ensure that you have up-to-date anti-virus and anti-spyware software installed on your computer.
- Install a firewall on your computer and make sure it is activated.

7.11 FAKE SOCIAL MEDIA PROFILE INVESTIGATION

There are social media platforms like facebook, twiter, etc. which are used to launch social engineering attacks, identity thefts, cyber stalking etc. So it is very important to identify fake social media profile to save yourself being a victim of such attacks. We will discuss investigating a fake social media profile in one of the most common social media platform, facebook in the next section.

How to spot fake facebook account[39]

1. **Know why it is important to spot a fake account:** First and foremost, somebody with a fake account is- almost by definition- a con artist. Unless you run with that crowd, you probably don't want them in your life.

- While they may present themselves as a friend, or even a romantic interest, their sole purpose in friending you may be as harmless as a mind game, or they may be after much more, such as your money, goods and property.
- The impostor might also be setting you up to steal your identity or valuable information from you that they can use to manipulate someone else.

2. **Don't talk to strangers.** At the least, think twice about accepting friend requests from people you don't know and who are not connected to you through legitimate, verifiable means. If you're not sure, do the following:

- Ask them questions: What makes you want to be your friend? How did they find out about you? Who do you know in common? By clicking on their name, you can see if you have any mutual friends. If you do, contact your friend. If not—that's a big red flag.

3. **Do a little detective work.** At the very least, it can be fun. You might also find out that your would-be "friend" is really bad news.

4. **Read the profile carefully.** Does what is being said add up or are there some really hard- to-believe statements being made? For example, maybe there is a photo of a very young person next to claims of being a professor or a CEO. Does the embellishment seem more than the usual "making oneself look good" and come across as simply implausible? Trust your own senses on this one. You could even ask for proof of some of the things the person has stated—they're approaching you, after all. You have every right to make sure they're legitimate.

5. **Check out their profile picture.** Is there only one? Is it way too perfect or does it seem touched up in any way? Maybe you've seen it before? A good photo — or a touched up

one
— may not be a negative sign, but it could be that they've simple scoured Google for an attractive photo, thinking nobody would ever find out. Try this:

- Click and drag their profile picture to your desktop.
  - Launch Google Chrome or Firefox, and navigate to Google Images

.

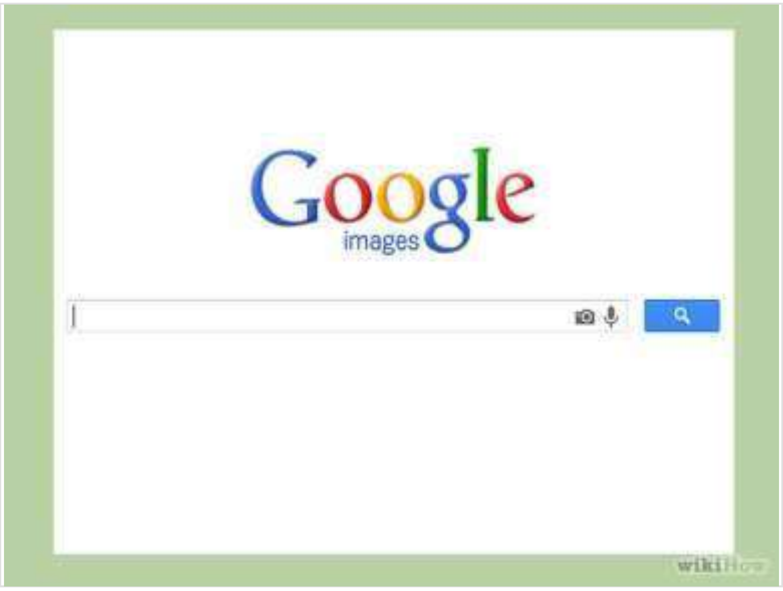


Figure 35: Google Image search

- Drag and drop the profile pic into the search field: it will expand, as shown:

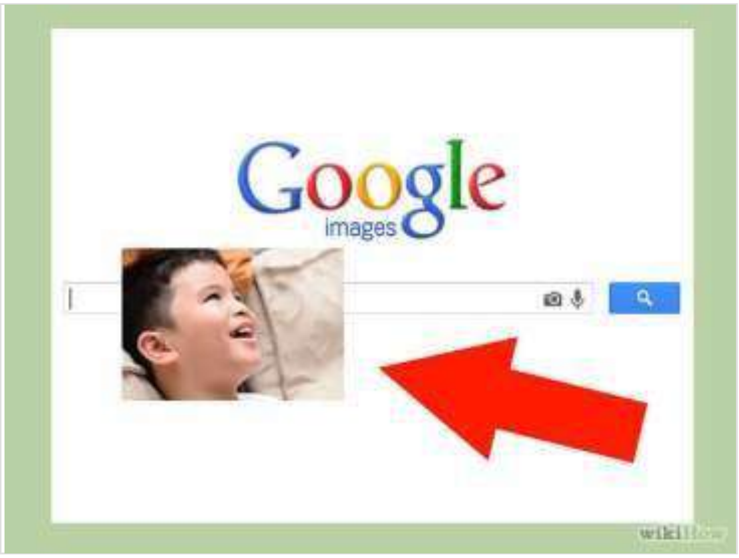


Figure 36: Searching image in google

- Google will either return an exact match (with information like names), or pictures similar to the original.

6. **Search their name online to see if it returns.** This won't be so useful if the name is a

common one, but for a more unusual one there might be some interesting returns.

- If they have a common name, add other information such as their location, approximate age, or any other information you can glean from their profile.
- Have they been tagged? A real person is generally tagged here and there as part of the Facebook sharing experience.

7. **Check out their friends.** Are their friends global or local? The more local the friends, the more likely the person is to be real. The more global their friendship list, with very few or no local friends, start getting suspicious. The lack of local friends suggests that this is not a real person you're dealing with but a fake account. This is often used by people pretending to be attractive young women. They will often contact you with a line like "I saw your picture and you looked nice."

8. **Block the request.** If you don't have a good feeling about somebody, there's a simple solution: don't just turn down the request for friendship, block them completely.

- Click on their Facebook name, and go to their Timeline. On the right, under the Cover Photo, click on the Message settings.
- You can block them from contacting you, or report them to Facebook if you feel they are a threat or involved in illicit or illegal activities.

9. **Create a "probationary period."** If you're in the (not-so-great) habit of accepting friend requests from friends of friends' friends, or friend people because they seem to have similar tastes to yours in music, cooking, dancing, or whatever, then you leave yourself open to the occasional fake.

    - While you can make wonderful connections in this way, try to always have someone you do know vouch for this person first. And if that's not possible, be alert to signs of weird behavior, such as suddenly bombarding you with likes, comments, photos, etc. on a daily basis.
    - If you hardly know this person, they should be taking things slowly and politely, not invading your space immediately.
    - If, after a week or two, you're not comfortable with your new friend, unfriend them!

10. **Beware interconnected faking.** At one time it was probably reasonable to think that if someone had a group of friends interacting with them and vouching for each other, that that person must be real. Not anymore!
    - There are increasing cases of one person running numerous fake Facebook accounts, pretending to be an array of different people, all vouching for one another and all trying to be friends with someone real!
    - An excellent example is the case of Natalia Burgess, who wove a web of deceit and caused many young males to fall for her various aliases — all because she felt inadequately loved. Sadly, impostors of this sort go to incredible lengths to create an array of fake accounts including other social media accounts and websites to give the impression that their fake personas are "real".

11. **Look for and record inconsistencies.** If you're being targeted by an elaborate web of lies, eventually these start to unravel. This is most evident in someone who is trying to maintain several fake Facebook accounts at once and eventually, they will drop the ball and mix up their stories.
    - If you start noticing this in response to questions, or in their comments, take note and remain alert for more inconsistencies.
12. **Do a double take if the person says anything weird or "out-of-character".**For example: if an adult is pretending to be a teen, they may say something that date them by referring to a historic event or person that teens wouldn't really know much about. Or they may prove to know way too much about a topic that someone they're claiming to be would not.
    - Take note of what the suspicious person says, as everyone slips up! No one is perfect, and they're bound to eventually say something that will give you a hint that your hunch is correct.
13. **Be really wary of undying declarations of love, affection, and romance.** If someone you've never met, who lives thousands of miles away from you, and who has barely revealed themselves gets amorous with you, be suspicious. Sometimes the faker does this because they love the feeling of playing with the life and feelings of someone else; sometimes it's because they're in love with online love but are too afraid to reveal their true selves (or they're in a relationship in real life); and other times it could be that they're after something, like money, sex, or drugs.
    - Question your own feelings and motivations if you start to feel something for a person who declares they love you online. Is it too sudden? Too weird? Too freaky? A little bit icky? Trust those feelings and delete this fake friend from your account.
    - If they ask you for sexy pictures, immediately be suspicious. A fake account is a good shill for getting free pornographic material that then gets passed around online.
14. **Unfriend them!** If you're suspicious, unsure, or uncomfortable with having them as part of your Facebook friends, pull the plug. It's not like they're your real friends or family, and they could cause you a lot of future problems.
    - Warn other friends of yours on Facebook if you know they have also friended the fake account; one of the tactics of an impostor is to befriend others in your circle of friends to try to make the friendship seem more "real

Tips

Warnings

a. Be careful what you put online and what you tell people you don't really know. Some people act very caring until they have enough information about you and then they turn around and blackmail you with it. If you don't know the person, no matter how friendly you've become in the online context, keep back your private details and keep everything very general.
b. Look for evidence of offline interaction with their Facebook friends. However, keep in mind that even this can be faked if they're running multiple Facebook accounts.
c. Check any links they've provided to personal websites, social media pages, etc., to help you to see if things add up.

a. Read **How to Reveal a fake Facebook account** if an impostor has stolen your own identity.

b. Keep an eye on your teens. Young people are the most vulnerable to building online relationships with people who don't exist. They fall in love with an image of the perfect person and the faker is happy to oblige for their own gratification or other reasons.

7.12 SUMMERY

1. The word "cyber" is derived from the Greek term "cybernetic" meaning "skilled in steering or governing".
2. Forensics is a discipline that dates back at least to the Roman era (and possibly event to ancient China), when people accused of crimes (and the accuser) presented evidence in front of a public audience (the Latin word forensis, means "of or before the forum").
3. The first specific computer crimes were recognized in the 1978 Florida Computer Crimes Act, which included legislation against the unauthorized modification or deletion of data on a computer system.
4. Canada was the first country to pass legislation in 1983.
5. Computer based crime is a criminal activity that is conducted purely on computers, for example cyber-bullying or spam.
6. Computer facilitated crime is a crime conducted in the "real world" but facilitated by the use of computers.
7. There are four main types of investigation performed by digital forensics specialists. They are criminal forensic, intelligence gathering, electronic discovery and intrusion investigation.
8. A digital forensic investigation generally consists of five steps viz. identification, data acquisition, data recovery, analysis and reporting.
9. Protocol are the set of rules and procedures governing the transmission of data over network.
10. You can find your IP address on a Windows computer by opening an MSDOS or Command window and typing one of "winipcfg" or "ipconfig".
11. The POP3 protocol enables any email program

anywhere on the Internet to connect to any email server to perform the usual email functions, such as reading and sending, as long as they have a valid account and password.

12. IMAP is a more modern protocol than POP3.
13. The IMAP features can be useful in several situations, for example when you are travelling and don't want to download your email onto a laptop because then you won't have them on your home computer when you get back.
14. Spammers want to avoid receiving non-delivery notifications (bounces) to their real addresses.
15. Fraudsters want to cover their tracks and remain anonymous.
16. Computer worms want to cause confusion or just don’t care about which sender addresses they use.
17. Phishers want to impersonate well-known, trusted identities in order to steal passwords from users.
18. The envelope sender address is used during the transport of the message from mail server to mail server, e.g. to return the message to the sender in the case of a delivery failure.
19. The header sender address of an e-mail message is contained in the "From" or "Sender" header and is what is displayed to the user by mail programs.
20. Fake emails or fake mails are those emails which pretend to come from a specific email address but are sent from some fake email senders.
21. Use WhatIsMyAddress Ip Lookup tool available at https://www.whatismyip.com/ip- address-lookup/ to lookup the origin of an IP address.

7.13 ACTIVITY: DO IT YOURSELF

1. Google the term identity theft and find out more on the topic.

2. Google the term social engineering attacks and find out how hackers use this techniques to find out your personal information.

3. Find out more about “Internet of things(IOT)”. Study about the lattest application of IOT.

4. Find all the country level domains.

5. Search the procedure to find out the email headers of yahoomail.

CHECK YOUR PROGRESS

1. State True and False

I. US was the first country to pass legislation for computer offence in 1983.

II. Child pornography is an example of computer facilitated crime.

III. Focus is criminal forensic on forensically sound data extraction and producing report/evidence in simple terms that a lay man will understand.

IV. Meta data and other logs can be used to attribute actions to an individual.

V. Data acquisition/data preservation is taking images of the drives/partition belonging to the identified system.

VI. Specialist forensic tools are used to track, audit and report on cases.

VII. Live search is a place where a text index of the digital media is created, allowing searches to be performed quickly using the index.

VIII. Netanalysis scans digital media for a variety of internet artefacts.

IX. Spammers want to impersonate well-known, trusted identities in order to steal passwords from users.

X. The envelope sender address is used during the transport of the message from mail server to mail server.

2. Fill in the blanks

i. BYOD stands for __________ _.

ii. PCIDSS stands for _______ _ _..

iii ____________ is instigated as a response to a network intrusion.

iv. File ____________ and meta-data can be used to identify the origin of a particular
piece of data.

v. C4P stands for __________ _.

vi. MTA stands for __________ _.

ANSWERS TO CHECK YOUR PROGRESS

1. State true or False

   i. False
   ii. False
   iii. True
   iv. True
   v. True
   vi. False
   vii. False
   viii. False
   ix. False
   x. True

2. Fill in the blanks

   i. Bring Your Own Device.
   ii. Payment Card Industry Data Security Standard.
   iii. Intrusion investigation
   iv. Artefacts
   v. Categoriser 4 Pictures.
   vi. Mail Transport Agent.

MODEL QUESTIONS

1. What are the drivers behind this new cyber reality?
2. What do you mean by cyber?
3. What are the two forms of cyber crime?
4. There are the main types of investigation performed by digital forensics specialists
5. Explain some examples of the kind of analysis an examiner might be asked to undertake.
6. Explain the various steps involved in digital forensic investigation.
7. What are the three broad categories into which the forensic tools are categorised into?
8. What is “ live search”?
9. What are internet domains?
10. What is the difference between POP3 protocol and IMAP protocol. Which is better?
11. What is an IP address.
12. How email works? Explain with a help of a diagram.
13. Explain the different types of email abuse.
14. Explain the steps to trace the location of an email sender.
15. Explain the motivation behind designing scam or hoax websites.

UNIT-VIII CRYPTOGRAPHY

STRUCTURE

8.0 OBJECTIVES

After going through this unit, you will be able to:

- Know the basic terminologies of cryptography
- Define the meaning of word cryptography
- Know the objectives of cryptography
- Differentiate various cryptographic techniques
- Define encryption
- Describe the importance and working of encryption
- Describe the working of Secret key cryptography
- Recognize application of Secret key cryptography
- Describe the working of Public key cryptography
- Recognize applications of public key cryptography

8.1 INTRODUCTION

Cryptography is "Art of writing or hiding secret". It is a science of protecting the information from theft or unauthorized access. To do so, important or confidential information is hided as or converted to some other form of gibberish data. Now original information can be recovered only by the right person or application[40].

Cryptography was developed to create secure communication while there was a third-party present also known as adverseries[41]. In the beginning, the cryptographic codes were written by hand to convert plaintext to cipher-text and vise-versa. Early cryptography was messages written in a language the other party could not read or words that were written in reverse order. Cryptography was used mainly by military officers and spies to secure the confidentiality of the messages.

Cryptography is the art of writing and solving code. It is used to secure files and is similar to a puzzle in which the message is scrambled using algorithms and unscrambled using another set of algorithms. Encryption is used in protecting passwords, securing classified messages and also used to protect personal and sensitive data.

The example of cryptography can be seen everywhere in our daily lives. For example the e- mails we send through g-mail or other mail servers are encrypted using algorithm to make sure no one else can intercept the messages being send and read them. The way the computer hides the password when we log in is also a form of cryptography.

Cryptography Objectives

Cryptography is needed in various scenarios varying from simple encryption of a small file to the complicated usages of smart cards used for windows authentications. Fundamentally, it is used in below mentioned scenarios

- Data at Motion - Cryptography is required when communicating over any non trusted medium. This medium can be internet, mobile phones, bank automatic teller machines, wireless intercom systems, Bluetooth devices, wireless microphones and portable storage disks. These days, organizations invest heavily to secure all the business communications like emails using cryptographic techniques and products. This is to ensure that no one else other than the trusted recipient can read the message.
- Data at Rest - Cryptography is must in securely storing all the sensitive and vital data. This is a basic provision mentioned in most of the compliances which an organization must meet. A simple example for this is Encrypting File system (EFS) which is a file system introduced in Windows operating system to provide file system level protection.
- Data integrity - Cryptography not only protects the information, but also verifies the integrity of data. This is necessary to ensure that the transferred data has not been tampered by a hacker.

Before discussing cryptography in detail, let us first gear up with some common terminologies used frequently in cryptography.

Cryptography Glossary

1. **Key-** In the world of cryptography, “Key” refers to a digital data or file which mathematically determines the output of a cryptographic algorithm when applied to an input message.
2. **Encryption-** Encryption is a process of transforming information, using mathematical algorithms, to some sort of “nonsense” data. To encrypt a message or plain text, one needs to select an Encryption algorithm and a key (or a key – pair, based on encryption

algorithm)

3. **Decryption-** Decryption is the reverse process of encryption, in which the encrypted message is processed and transformed back to the original message. Decryption can succeed if and only if, the correct algorithm (the one used during encryption process) and authentic keys are used.
4. **Digital certificates-** Digital certificates are file used for proving the authenticity of the user or sender. Digital certificates have information about the authority, which has issued the certificate and also, to whom the certificate is issued. Now, there are worldwide trusted certifying authorities (CA) like VeriSign, etc. So, any certificate issued by a Trusted CA, can be trusted as authentic and any information (generally cryptographic keys) contained in the certificate can be safely assumed to be from a trusted source.

8.2 TYPES OF CRYPTOGRAPHY

Cryptography[42] is essentially the science of writing in secret code. In data and telecommunications, cryptography has specific security requirements, such as authentication,

privacy or confidentiality, integrity, and non-repudiation. To meet these security requirements, we employ secret key (or symmetric) cryptography, public-key (or asymmetric) cryptography, and hash functions.

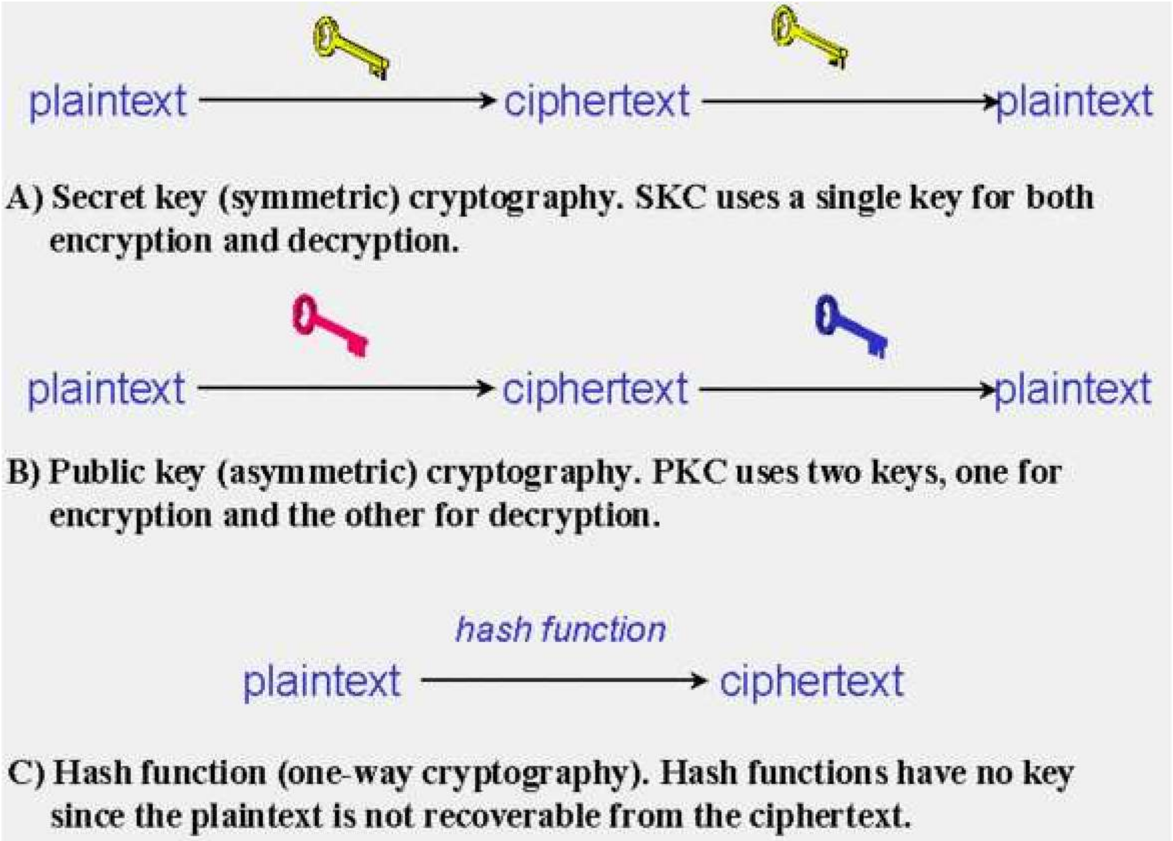

plaintext
ciphertext
plaintext
A) Secret key (symmetric) cryptography. SKC uses a single key for both encryption and decryption.
plaintext
ciphertext
plaintext
B) Public key (asymmetric) cryptography. PKC uses two keys, one for encryption and the other for decryption.
hash function
plaintext
ciphertext
C) Hash function (one-way cryptography). Hash functions have no key since the plaintext is not recoverable from the ciphertext.

Figure 37: Types of cryptography[1]

There are three main types of cryptography:

1. Secret key cryptography
2. Public key cryptography
3. Hash function
   1. Secret key cryptography – In this type of cryptography, the information is encrypted using a “secret” key. For decrypting the information, the user must possess the secret key. This type of encryption scheme is also known as Symmetric key encryption. In case of confidential data being transferred after encrypted with Symmetric key algorithms, both the sender and receiver must share the secret key. This encryption scheme is preferred over Public key cryptography when a large amount of data is to be encrypted, because it takes lesser time to encrypt or decrypt the data. An example of symmetric key cryptography is the whole disk encryption used by EFS of Windows operating system (EFS).
   2. Public key cryptography – This scheme of cryptography involves two keys or Key-pair, one is a Public key and the other one is private key. Public and private keys are mathematically related and it is impossible to calculate the private or public half of the pair given one key (private or public) of the key pair.

      The public key is meant to be distributed publicly whereas the corresponding private key must be kept much secured, ideally in a HSM (Hardware Security Module) device.
      If some information is encrypted by a public key, it can be decrypted only by the corresponding private key. Thus, in this scheme, it is not necessary for the sending and receiving users to share the common secret. The recipient distributes his public key. Sender encrypts the data using this public key. Now the data can be decrypted only by the receiver because he only has the correct private key.
   3. Hash functions - Hash functions are one-way cryptographic schemes. In this method, a plain text is processed by the hash algorithm and the output is the hashed value of the original text. From this hashed value, it is impossible to recover the original information. Now a days, Hashing function and algorithms are used in the Authentication module of almost every application including the Windows authentication mechanism.

8.3 WHY OS ENCRYPTION IMPORTANT

Do you know who may be reading your E-Mail? It is transmitted in plain text over unknown pathways and resides for various periods of time on computer files over which you have no control. Whether you're planning a political campaign, discussing your finances, having an affair, completing a business deal, or engaging in some totally innocuous activity, your messages have less privacy than if you sent all of your written correspondence on postcards.

Why should Encryption be used?

Encryption is important because of the nature of the Internet and the electronic medium. It allows effective scanning of message contents using sophisticated filtering software.

Electronic mail is gradually replacing conventional paper mail and messages can be easily and automatically intercepted and scanned for interesting keywords. Another problem with E- Mail is that it is very easy to forge the identity of the sender. The solution to these problems is to use cryptography. However, there are restrictions on the export and use of strong cryptography, particularly in the USA, but now gaining momentum in other countries. Furthermore, some governments, and again the USA is the most prominant, want decryption keys lodged with escrow agents, so that law enforcement agencies can, with appropriate authorisation, intercept and decrypt private messages. It is often claimed that this facility is no different from powers that the government has always possessed to wiretap telephones. There is however, a vital difference. Citizens are now being asked to take action to make themselves available for surveillance.

Cryptography today involves more than encryption and decryption of messages. It also provides mechanisms for authenticating documents using a digital signature, which binds a document to the possessor of a particular key, while a digital timestamp binds a document to its creation at a particular time. These are important functions which must take the place of equivalent manual authentication procedures as we move into the digital age. Cryptography also plays an important part in the developing field of digital cash and electronic funds transfer.

The major applications for encryption may then be summarised as:

- To protect privacy and confidentiality.
- To transmit secure information (e.g. credit card details)
- To provide authentication of the sender of a message.
- To provide authentication of the time a message was sent.

How does it work?

Up until the mid 1970's cryptography was an arcane science practised largely by government and military security experts. That situation changed dramatically following the development of public key cryptography by Hellman and Diffie in 1975. This development solved a major problem with most cryptographic systems - that of exchanging keys, and preceded a rapid escalation in civil involvement in this field of endeavour.

Public key cryptography systems work with public and secret (or private) keys. You generate these yourself as a once-only task. You distribute your public key to anyone who may need to send you encrypted information, or you can place it on one of the many public-key repositories around the globe. Your public key is then used by others to encrypt messages sent to you. Only you can decrypt such a message since the secret key is needed to perform this task. In practice, because public key encryption is a time-consuming process, many cryptosystems only use the public key to encrypt a random session key, which is then used to encrypt the actual message.

As an example, to exchange secure communications with someone the procedure would be as follows. Here we will introduce Alice and Bob, the renowned "first couple" of cryptography.

- Alice and Bob exchange their respective public keys or obtained them from a public key repository.
- Alice transmits the message encrypted with the Bob's public key.
- Bob decrypts it with his secret key.
- Only Bob can decipher the message. Even Alice will be unable to decipher the message once encrypted, unless she has included herself as a recipient (using multiple keys to encrypt the message).
- Cryptography can also be used to produce a digital signature which proves that the transmission is unchanged and can authenticate the sender. In this case the Bob would use Alice's public key to read the signature created by Alice's secret key.

8.4 PUBLIC KEY CRYPTOGRAPHY

Public keys and private keys

When using public key cryptography, Alice and Bob both have their own key pairs. A key pair consists of a public key and a private key. If the public key is used to encrypt something, then it can be decrypted only using the private key. And similarly, if the private key is used to encrypt something, then it can be decrypted only using the public key. It is not possible to figure out what the private key is given only the public key, or vice versa. This makes it possible for Alice and Bob to simply send their public keys to one another, even if the channel they are using to do so is insecure. It is no problem that Eve now gets a copy of the public keys. If Alice wants to send a secret message to Bob, she encrypts the message using Bob's public key. Bob then takes his private key to decrypt the message. Since Eve does not have a copy of Bob's private key, she cannot decrypt the message. Of course this means that Bob has to carefully guard his private key. With public key cryptography it is thus possible for two people who have never met to securely exchange messages.

Combining public key and secret key cryptography

A problem with public key cryptography is that it is very slow. Encrypting a message that is several megabytes long takes a very long time, much longer than when using secret key cryptography. For this reason few people use only public key cryptography. What Alice usually does is encrypt the message using a secret key encryption algorithm and a key she made up on the spot. She then encrypts this key (called the session key) using Bob's public key. Bob can then obtain the session key by decrypting it using his private key. And with the session key he can decrypt the message. This way a long message is encrypted very quickly and Alice can still send it to Bob without needing a secure way of agreeing on the key.

How public key cryptography works

Public key cryptography systems are usually based on the assumption that a particular mathematical operation is easy to do, but difficult to undo unless you know some particular secret. This particular secret that serves as the private key. The two most famous public key cryptography systems are Diffie-Hellman and the RSA system, named after its inventors Rivest, Shamir and Adleman. More recently public key cryptography based on so-called elliptic curves has gotten a lot of attention. Using public key cryptography it is possible to easily encrypt a message for multiple recipients. It is even possible to later authorize recipients to read the message. The message is simply encrypted with a session key. The session key is encrypted multiple times, once for every recipient using his public key. If later another recipient needs to be added, the session key is simply then also encrypted using his public key. Every recipient can now decrypt the session key independently from every other recipient.

8.5 APPLICATIONS OF PUBLIC KEY CRYPTOGRAPHY

Secure Web communication

One important application of public key cryptography is encrypted communication with a Web server. This enables Alice to securely place an order and to transmit her credit card details. Alice's Web browser generates a random number which will be used to encrypt all communication with the server using secret key cryptography. The browser obtains a copy of the public key of the server and uses this public key to encrypt this random number (called the session key). The result is sent to the server. After that, browser and server can encrypt all information so that Eve cannot read it.

Alternatively, it is the server that generates the session key. This requires that the browser sends its own public key to the server so that the session key can be transmitted securely. This makes it easier to upgrade the hardware or software that generates the session key.

Secure content distribution

Content such as music or video can be distributed in encrypted form. To be able to play it back, the recipient needs the decryption key. To avoid having to encrypt the entire movie or song again for every recipient, the content is encrypted with a session key. Every recipient receives the encrypted content together with the session key. The session key is of course encrypted using the recipient's public key. This way only a very small amount of data needs to be encrypted again for every recipient.

It is even possible to distribute the encrypted content in advance. If the recipient wants to play back the content, he can at any time contact the distributor and obtain (purchase) a copy of the session key encrypted using his public key. This is sometimes called 'superdistribution'.

8.6 SECRET KEY CRYPTOGRAPHY

Encryption and decryption using a secret key

To secretly communicate with Bob, Alice encrypts her messages before sending them. There are many techniques (cryptographic algorithms) that she can use. All these algorithms have in common that they can transform a message using a key into something that resembles random noise. This is called encrypting the message. Only the persons who know the key can transform the random noise back into the original message, or in other words, decrypt the message. This means that those persons must keep this key a secret, hence the name secret key cryptography.

How to get the key to the recipient

A fundamental problem with secret key encryption is that somehow the secret key has to be delivered to the recipient of the message in a secure way. Once that key has been securely delivered, other keys can be delivered by simply encrypting them with that first key. One way to solve this problem is to have Alice and Bob meet in person so they can agree on a key. They must make sure that Eve is not listening in on them, otherwise Eve also learns the key. This applies especially if Alice and Bob agree on a key via telephone or e-mail. Of course Bob must also be able to distinguish Alice and Eve if they meet for the first time (for Alice it shouldn't be a problem to tell Bob from Eve).

If Alice and Bob cannot meet in private to agree on the key, it is very difficult for them to use secret key cryptography. If they simply agree on a key by e-mail for example, Eve could be listening in on their e-mail conversation and thus also learn what the key is. If Alice and Bob

had a secure channel that Eve could not tap, they could use that channel to agree on a secret key. However, then they could also use the secure channel to simply transmit their messages. This problem is solved by using public key cryptography, which is discussed in the next section.

How secret key cryptography works

Secret key cryptography transforms (scrambles) a message into something resembling random noise. The precise transformation is determined by the key. Mathematically seen, a cryptographic algorithm is a function that maps a message onto a ciphertext (an encrypted message). By using keys, it is possible to encrypt many different messages using one particular cryptographic algorithm in different ways. And keeping the key a secret is much easier than keeping a complete algorithm a secret.

Some cryptographic algorithms operate on single characters of the message. These are called stream ciphers. Others operate on entire blocks, and therefore are called block ciphers. Stream ciphers are easier to implement in hardware than block ciphers, and they are also generally faster. Block ciphers tend to be harder to crack. We will discuss an example of a secret key cryptographic system to further elaborate the concept.

A very simple technique to encrypt messages is to replace every letter of the message with one that is a certain number of positions further in the alphabet. The key then is the number of positions. For example, the message "This is an example" can be encrypted using the key "1 position" into the encrypted message "Uijt jt bo fybnqmf". Taking the letter that is 1 position previous in the alphabet results in the original message again.

This system is of course not very secure. There are only twenty-six possible keys. Eve can simply try out all the keys to see which one results in a readable message. Furthermore, it is a well-known fact that certain letters occur more often in messages than others. The letter "e" is the most frequently used letter in the English language, for example. Using this fact

Eve can simply count which letter occurs the most often in the encrypted message and replace that one with the letter "e". She then knows how many positions she has to rotate to get from "e" to the encrypted version of "e" and thus she immediately knows the key.

In principle, all cryptographic systems can be broken. At the very least, Eve can try out all different keys until she finds one that successfully decrypts the message. Eve might also be able to break one of the mathematical principles behind the cryptographic algorithm that Alice and Bob use. For example, some cryptographic systems assume that it is very difficult to divide a number into its prime factors. Eve might find a quick way to do this. This then enables Eve to read Alice and Bob's messages or to recover their keys.

There is one cryptographic algorithm that cannot be broken. This algorithm is called the one- time pad (OTP). According to this algorithm, Alice generates a very large sequence of random numbers. The numbers in the sequence serve as the key. The sequence is called the "pad". Alice communicates the sequence to Bob in a secure way, so that Eve cannot obtain a copy of the key.

Every character in the message that Alice wants to send to Bob is encrypted with a different number in the sequence. In practice this means that the first character of the message will be encrypted with the first number in the sequence, the second character with the second number, and so on. When Bob receives the encrypted message, he takes out his copy of the sequence and simply decrypts the first character with the first number in the sequence, the second character with the second number, and so on.

Because every character of the message is encrypted with a different key, there is nothing Eve can do to guess the key. Even if she knew that the first words of the message were "Dear Bob", she could not use this information to recover the key of other words in the message. Every number is chosen randomly, so Eve has no way to know which number is the right one, even if Eve knew how to decrypt all other characters.

It is absolutely essential that every number in the sequence is chosen randomly and is only used once. If Eve can recover some of the numbers in the sequence and use those to predict other numbers, she can eventually reconstruct the entire sequence and thereby decrypt the message. For this reason it is not a good idea to use a random number generator implemented in software. Those generators are unable to generate really random numbers. They use a mathematical function that generates a set of numbers that appears to be random. But if you know the mathematical function and the number that it last generated, you can immediately compute the next "random" number.

To achieve this unbreakability, Alice and Bob must have very large sequences that contain only really random numbers. This makes an OTP very difficult to manage. It is said to have been used for the "hotline" between Washington and Moscow during the Cold War. In a case like that, it is practical to send couriers carrying suitcases chained to their arms to securely transmit the pad.

8.7 APPLICATIONS OF SECRET KEY CRYPTOGRAPHY

Secret key encryption is most often used to encrypt data to be stored on a particular location. If the encrypted data has to be transmitted, there always is the problem of how to get the secret key to the recipient in a safe way. Usually the key is encrypted using public key encryption so it can be transmitted safely.

Hiding spoilers

Even though it is not secure, the simple alphabet shifting system is still in use on the Internet. It is used to hide "spoilers" (revealing plot twists in movies or books) and potentially offensive messages from unsuspecting readers. Such messages are encrypted using the key "13 positions". Anyone can thus decrypt the message by simply taking the letter that is 13 positions previous in the alphabet. However, this requires some active step by the reader, and so he should then not be surprised or upset if the decrypted message reveals something about the plot of a movie he wanted to see. This system is commonly known as "ROT-13".

Encrypting the contents of hard disks

Using secret key encryption Alice can encrypt her entire hard disk so the data on it is safe if the disk (or laptop containing it) is ever stolen. Disk encryption programs exist that can encrypt and decrypt data as it is being written and read to and from the hard disk. This way Alice does not notice that her data is stored encrypted, except for the fact that disk access might be a bit slower. Once she turns off her computer, it is not possible anymore for Eve to read the data.

Protecting pay TV transmissions

Secret key encryption and smart cards are used for example in pay TV applications. Sometimes this is referred to as "conditional access" television. Television programs (usually premium movies, football or soccer matches and adult content) are encrypted using a secret key. To make it difficult for Eve to obtain this key, the secret key is changed every few minutes or sometimes even every few seconds. This way, even if Eve can successfully use a brute force attack to guess the key, she only has a very small portion of the television program. Alice has a set-top box and a smart card that allows her to decrypt the television programs. The set-top box passes the decrypted television program on to the television. Originally these boxes were designed to be placed on top of the television set, hence the name.

Special messages, called Entitlement Control Messages (ECMs), are sent along with the program. These messages contain the secret keys. Of course the ECMs themselves are also encrypted, this time using a key stored on the smart card. Alice's set-top box receives the ECMs and passes them on to the smart card. The smart card decrypts the ECMs and extracts the secret keys contained therein. This allows the set-top box to decrypt the television program.

The keys needed to decrypt the ECMs can be programmed on the smart card in advance. By regularly changing these keys, Alice is forced to purchase a new smart card every month or so. If Eve manages to make a copy of the smart card, or to extract the keys from it, she will only be able to watch the programs for the rest of that particular month.

Another option is to regularly send out so-called Entitlement Management Messages (EMMs) that contain the keys needed to decrypt the ECMs. The EMMs themselves are then encrypted with keys stored on the smart card. The service provider then every month simply sends out a new EMM. This provides much greater flexibility, and Alice does not have to go to the store every month. Every smart card can now have a different key. The service provider sends out different EMMs for all the smart cards in the system. Every EMM thus is readable only by one smart card. If the service provider thinks a particular smart card has been copied illegally, he simply does not send out a new EMM for that particular smart card.

8.8 SUMMERY

1. Cryptography is art of writing or hiding secret.
2. Encryption is used in protecting passwords, securing classified messages and also used to protect personal and sensitive data.
3. Key refers to a digital data or file which mathematically determines the output of a cryptographic algorithm when applied to an input message.
4. Digital certificates are file used for proving the authenticity of the user or sender.
5. In secret key cryptography the information is encrypted using a “secret” key. For decrypting the information, the user must possess the secret key.
6. In public key cryptography, two keys or Key-pair are invloved, one is a Public key and the other one is private key. Public and private keys are mathematically related and it is impossible to calculate the private or public half of the pair given one key (private or public) of the key pair.
7. In Hash functions method, a plain text is processed by the hash algorithm and the output is the hashed value of the original text. From this hashed value, it is impossible to recover the original information.

CHECK YOUR PROGRESS

1. Fill in the blanks.

i. Cryptography was developed to create secure communication while there was a third- party present also known as_______.

ii. ________________is used in protecting passwords, securing classified messages and also used to protect personal and sensitive data.

iii. Cryptography not only protects the information, but also verifies the ___________ of data.

iv. ________________is the reverse process of encryption.

v. EMM stands for_______________.

vi. ________________are file used for proving the authenticity of the user or sender

vii. OTP stands for_______________.

2. State True or False

i. Hash functions are two-way cryptographic schemes.

ii. Encryption is also used to provide authentication of the time a message was sent.

iii. Secret key cryptography transforms (scrambles) a message into something resembling random noise.

iv. Your private key is used by others to encrypt messages sent to you.

ANSWERS TO CHECK YOUR PROGRESS

1. Fill in the blanks.

i. adverseries
ii. Encryption
iii. integrity
iv. Decryption
v. Entitlement Management Messages
vi. Digital certificates
vii. One Time Pad

2. State True or False

i. False
ii. True
iii. True
iv. False

MODEL QUESTIONS

1. What is cryptography? What are the objectives of cryptography?
2. What are the various types of cryptographic techniques?
3. Define:
   a. Encryption
   b. Decryption
   c. Key
   d. Digital Certificate
4. Explain the importance of using encryption.
5. Define public key cryptography in details.
6. Explain various public key cryptography examples.
7. How secret key cryptography works.
8. Explain various secrets key cryptography examples.

References, Article Source & Contributors

[1]. (n.d.). Retrieved Dec. 07, 2015, from Istqbexamcertification.com

[2]. (n.d.). Retrieved Dec. 07, 2015, from Softwaretestinghelp.com

[3]. (n.d.). Retrieved Dec. 07, 2015, from www.onestoptesting.com

[4]. Babu B., S., & Venkataram, P. Wireless and Mobile Security 1st Edition. Mcgraw Hill Education.

[5]. Bose, R. (2008). Mcgraw Hill Education. Mcgraw Hill Education.

[6]. Bunting, S., & Wei, W. (2006). The Official EnCE: EnCase Certified ExaminorStudy Guide. Wiley Publishing Inc.

[7]. Cryptography. (n.d.). Retrieved Sep. 29, 2015, from Wikispaces: http://icsproject.wikispaces.com/Cryptography

[8]. Cryptography. (n.d.). Retrieved Oct. 01, 2015, from ICSProjects.Wikispaces.com: http://icsproject.wikispaces.com/Cryptography

[9]. CYBER SECURITY MANIFESTO 2.0. (2012, Oct. 01). Retrieved Sep. 26, 2015, from cybersecuritymanifesto: http://cybersecuritymanifesto.com/

[10]. CYBER SECURITY MANIFESTO 2.0. (2012, Oct. 01). Retrieved Sep. 28, 2015, from cybersecuritymanifesto.com: http://cybersecuritymanifesto.com/

[11]. Edson, J. (2011, July 25). A Brief History Of Forensic Science. Retrieved Oct. 04, 2015, from riaus.org.au: http://riaus.org.au/articles/a-brief-history-of-forensic-science/

[12]. Gallagher, S. (2013, Oct. 02). We are not who we are. Retrieved Sep. 26, 2015, from Security Blog: https://securityblog.redhat.com/tag/two-factor-authentication/

[13]. Gardner, M. (1972). Codes, ciphers, and secret writing.

[14]. Glass, E. (2003). The NTLM Authentication Protocol and Security Support Provider. Retrieved Sep. 26, 2015, from Sourceforge: http://davenport.sourceforge.net/ntlm.html

[15]. (1998). How Email Works. In P. Grall, How Internet Works (p. 85). Que

Corporation. [16]. Gupta, A. (2011, March 01). Digital Forensic Analysis Using

BackTrack, Part 1.
Retrieved Sep. 26, 2015, from opensourceforu: http://opensourceforu.efytimes.com/2011/03/digital-forensic-analysis-using-backtrack-part- 1/

[17]. Gupta, A. (2011, March 01). Digital Forensic Analysis Using BackTrack, Part 1. Retrieved Oct. 03, 2015, from OpenSourceForU: http://opensourceforu.efytimes.com/2011/03/digital-forensic-analysis-using-

backtrack-part- 1/

[18]. Havercan, P. (2015, July 17). A plain person's guide to Secure Sockets Layer. Retrieved Sep. 26, 2015, from http://peter.havercan.net/computing/plain-persons-guide-to- secure-sockets-layer.html

[19]. How it works. (2010, Jan. 17). Retrieved Sep. 26, 2015, from Wikidot: http://pychatter.wikidot.com/how-it-works

[20]. How to Reveal a Fake Facebook Account. (n.d.). Retrieved Sep. 27, 2015, from www.wikihow.com: http://www.wikihow.com/Reveal-a-Fake-Facebook-Account

[21]. Introduction to computer forensics. (n.d.). Retrieved Oct. 03, 2015, from Forensic Control: https://forensiccontrol.com/resources/beginners-guide-computer-forensics/

[22]. Introduction to computer forensics. (n.d.). Retrieved Oct. 03, 2015, from Forensic Control: https://forensiccontrol.com/resources/beginners-guide-computer-forensics/

[23]. Introduction to Cryptography. (2001, Feb. 24). Retrieved Oct. 01, 2015, from efa.org.au: https://www.efa.org.au/Issues/Crypto/crypto1.html

[24]. Introduction to Digital Forensics. (2011, Nov. 16). Retrieved Sep. 28, 2015, from Wikibooks: https://en.wikibooks.org/wiki/Introduction_to_Digital_Forensics

[25]. Kerberos Authentication. (n.d.). Retrieved Sep. 26, 2015, from Interactiva: http://computers.interactiva.org/Security/Authentication/Kerberos/

[26]. Kumar, N. Wireless Security – Best Practices.

http://www.chmag.in/. [27].learncisco. Understanding WLAN Security.

[28]. learncisco. (2015). Wireless Technology. Haldwani.

[29]. Maxim, M., & Pollino, D. (2002). Wireless Security. Osborne/McGraw-Hill.

[30]. Mehnle, J. (2010, April 17). Sender Policy Framework. Retrieved Sep. 28, 2015, from Openspf: http://www.openspf.org/Introduction

[31]. Morton, T. (2013, Sep. 13). Types of investigations. Retrieved Oct. 04, 2015, from Introduction to Digital Forensics:

https://en.wikibooks.org/wiki/Introduction_to_Digital_Forensics/Types

[32]. Nelson, B., Phillips, A., & Steuart, C. (2009). Guide to Computer Forensics and Investigations. Cengage Learning.

[33]. Nichols, R. K., & Lekkas, P. C. Wireless Security: Models, Threats, and Solutions. McGraw Hill TELECOM.

[34]. Nolan, R., O'Sullivan, C., Branson, J., & Waits, C. (2005). First Responders guide to Computer Forensic. CERT Training and Education.

[35]. Pachghare, V. K. (2008). Cryptography and Information Security: Second

Edition. PHI. [36]. Password Authentication Protocol. (2015, July 17). Retrieved Sep.

26, 2015, from WIKIPEDIA: https://en.wikipedia.org/wiki/Password_Authentication_Protocol

[37]. Peterson, D. (2015, July 06). Computer Forensics Miami. Retrieved Oct. 03, 2015, from computer-forensics.wikidot: http://computer-forensics.wikidot.com/

[38]. Public key cryptography. (2005, Oct. 01). Retrieved Oct. 01, 2015, from iusmentis.com: http://www.iusmentis.com/technology/encryption/crashcourse/publickeycrypto/

[39]. Quirk, S. (2014, Mar. 13). Concordia Password Security Policy. Retrieved Sep. 26, 2015, from http://kb.cu-portland.edu/Password+Security

[40]. Recognise scam or hoax emails and websites. (n.d.). Retrieved Sep. 27, 2015, from https://www.communications.gov.au: https://www.communications.gov.au/what-we- do/internet/stay-smart-online/your-identity/recognise-scam-or-hoax-emails-and-websites

[41]. Rowlingson, R. (2005). An Introduction to Forensic Readiness Planning. available under Open Government Licence http://www.nationalarchives.gov.uk/doc/open- government-licence/version/2/, Centre for the Protection of National Infrastructure.

[42]. Saylor, A. (2015). CRYPTOGRAPHY. Retrieved Oct. 01, 2015, from learn.saylor.org: https://learn.saylor.org/course/cs409

[43]. Sbh. (2011, Jan. 14). An Introduction to Cryptography. Retrieved 10 01, 2015, from go4experts: http://www.go4expert.com/articles/introduction-cryptography-t24529/

[44]. Schneier, B. (1994). Applied Cryptography. Wiley.

[45]. Secret key cryptography. (2005, Oct. 01). Retrieved Oct. 01, 2015, from iusmentis.com: http://www.iusmentis.com/technology/encryption/crashcourse/secretkeycrypto/

[46]. Selecting a strong password. (2015, Sep. 10). Retrieved Sep. 26, 2015, from Wordpress: https://en.support.wordpress.com/selecting-a-strong-password/

[47]. Singh, S. (1999). The Code Book: The Evolution Of Secrecy From Mary, Queen Of Scots To Quantum Cryptography. New york: Doubleday.

[48]. Stewart, W. (2000, Jan. 07). How Email Works. Retrieved Sep. 28, 2015, from http://www.livinginternet.com/: http://www.livinginternet.com/e/ew.htm

[49]. The National Archives. (2011). Digital Continuity to Support Forensic Readiness. Retrieved Oct. 04, 2015, from nationalarchives: http://www.nationalarchives.gov.uk/documents/information-management/forensic- readiness.pdf

[50]. Tutorials Point. (n.d.). Retrieved Dec. 07, 2015, from www.tutorialspoint.com

[51]. Understanding Authentication. (2008, Feb. 14). Retrieved Sep. 26, 2015, from Go4Experts: http://www.go4expert.com/articles/understanding-authentication-t8842/

[52]. Verma, D. (2012, Nov. 05). How To Identify Fake EMail And Trace Sender's Location. Retrieved Sep. 27, 2015, from http://www.usethistip.com: http://www.usethistip.com/2012/11/how-to-identify-fake-email-and-trace.html

[53]. Wheelbarger, S. (2009, Aug. 27). CyberForensics. Retrieved Oct. 04, 2015, from Wikidot: http://colbycriminaljustice.wikidot.com/cyberforensics

[54]. wikipedia. (2015). Wireless LAN. Haldwani:

wikipedia. [55]. Wikispaces. (2015). Wireless

Networking. Wikispaces.

[56]. Xiao, Y., Shen, X., & Du, D. (2011). Wireless Network Security. Springer.

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

(Established by Act No. 19 of 2019 of the Legislature of State of Punjab)

The Motto of the University
(SEWA)

SKILL ENHANCEMENT EMPLOYABILITY WISDOM ACCESSIBILITY

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ, ਪਟਿਆਲਾ
JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA

M.SC. (COMPUTER SCIENCE)
SEMESTER-II
Course: SOFTWARE ENGINEERING (MSCS-2-04T)

ADDRESS: C/28, THE LOWER MALL, PATIALA-147001
WEBSITE: www.psou.ac.in

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

M.Sc. (Computer Science) Programme Coordinator:
Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
JGND PSOU, Patiala

Faculty of School of Science and Emerging Technologies:
Dr. Baljit Singh Khera (Head)
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Kanwalvir Singh Dhindsa
Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Amitoj Singh
Associate Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Dr. Monika Pathak
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Business Management & Commerce:
Dr. Pooja Aggarwal
Assistant Professor, School of Business & Commerce
Jagat Guru Nanak Dev Punjab State Open University, Patiala
Faculty of School of Social Sciences and Liberal Arts:
Dr. Pinky Sra
Assistant Professor, School of Social Sciences and Liberal Arts
Jagat Guru Nanak Dev Punjab State Open University, Patiala

ਜਗਤ ਗੁਰੂ ਨਾਨਕ ਦੇਵ
ਪੰਜਾਬ ਸਟੇਟ ਓਪਨ ਯੂਨੀਵਰਸਿਟੀ
ਪਟਿਆਲਾ

JAGAT GURU NANAK DEV
PUNJAB STATE OPEN UNIVERSITY, PATIALA
A State University Established by Govt. of Punjab vide Act No. 19
of 2019 and Approved Under section 2(f) of UGC

PROGRAMME COORDINATOR

Dr. Karan Sukhija (Assistant Professor)
School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

COURSE COORDINATOR & CONTENT WRITER

Dr. Monika Pathak (Assistant Professor)
School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

Course: Software Engineering	
Course Code: MSCS-2-04T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Understand the software development life cycle which increases the growth opportunity.
CO2	Learn the detail knowledge of software requirement analysis.
CO3	Understands the detailed knowledge of software design and coding.
CO4	Understand the software testing that is relevant to the industry.
CO5	Acquire the detail knowledge of the fundamentals, including terminology; the nature and need for maintenance; maintenance costs and software evolution

JAGAT GURU NANAK DEV

PUNJAB STATE OPEN UNIVERSITY PATIALA

(Established by Act No.19 of 2019 of Legislature of the State of Punjab)

PREFACE

Jagat Guru Nanak Dev Punjab State Open University, Patiala was established in Decembas 2019 by Act 19 of the Legislature of State of Punjab. It is the first and only Open Universit of the State, entrusted with the responsibility of making higher education accessible to all especially to those sections of society who do not have the means, time or opportunity to pursue regular education.

In keeping with the nature of an Open University, this University provides a flexible education system to suit every need. The time given to complete a programme is double the duration of a regular mode programme. Well-designed study material has been prepared in consultation with experts in their respective fields.

The University offers programmes which have been designed to provide relevant, skill-based and employability-enhancing education. The study material provided in this booklet is self instructional, with self-assessment exercises, and recommendations for further readings. The syllabus has been divided in sections, and provided as units for simplification.

The Learner Support Centres/Study Centres are located in the Government and Government aided colleges of Punjab, to enable students to make use of reading facilities, and for curriculum-based counselling and practicals. We, at the University, welcome you to be a part of this institution of knowledge.

Prof. G. S. Batra,
Dean Academic Affairs

Name of Programme: M.Sc. (Computer Science)
Name of Course: Software Engineering
Course Code: MSCS-2-04T
Semester: 2nd Semester

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 4
Pass Percentage: 40%

SECTION – A

Unit I: Introduction of Software Engineering– The Problem Domain, Software Engineering, Challenges, Software Engineering Approach. Software development life cycle and its phases,Software development process models: Waterfall, Prototyping, Iterative.

Unit II: Software Process- Characteristics of software process, Project management process, Software configuration management process.

Unit III: Project Planning – activities, COCOMO model.Software Metrics – Definition, Importance, Categories of metrics. Software Quality – Attributes, Cyclomatic complexity metric.

Unit IV: Software Requirements Analysis – Need for SRS, Data flow diagrams, Data Dictionary, entity relationship diagram, Characteristics and components of SRS, validation, metrics.

SECTION-B

Unit V: Software Design – Design principles, Module-level concepts, Structure Chart and Structured Design methodology, verification, metrics : network metrics, information flow metrics.

Unit VI: Coding – Programming Principles and Guidelines, Verification- code inspections, static analysis.

Unit VII: Software Testing – testing fundamentals, Black Box Testing: Equivalence class partitioning, Boundary value analysis, cause-effect graphing; White Box Testing : Control flow and Data flow based testing, mutation testing; levels of testing, test plan, test case specification, test case execution and analysis.

Unit VIII: Software maintenance – Categories of maintenance. Software Reliability – Definition, uses of reliability studies

Reference Books:

- An Integrated approach to Software Engineering, Third Edition 2005, PankajJalote, Narosa Publications.
- Software Engineering, Revised Second Edition , K.K. Aggarwal, Yogesh Singh, New Age International Publishers.
- Software Engineering – A Practitioner’s Approach, Fifth Edition, Roger. S. Pressman, McGraw Hill
- Fundamentals of Software Engineering, Fifth Edition, Rajib Mall, PHI Learning

M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 1: INTRODUCTION OF SOFTWARE ENGINEERING

1.1 **INTRODUCTION**

Today, computers are everywhere .Almost all organizations are dependent on the computer. It helps in marketing, accounting, sales etc. Computer is a combination of hardware and software. Computer cannot work without software. Software helps hardware to perform various tasks. Software has changes the society and their work. There are many different software packages in the market. People choose software according to their need and facility. It helps people in many different ways, including saving time and money. For example, CRM (customer relationship management) software helps business to manage their customer relationships.

1.2 **PROGRAM**

1. A program is normally complete in itself and small in size.
2. It is used by the single user. Program lacks in proper documentation because single user is the author or single developer of the program. User makes little documentation according to his/her needs without using any format of documentation.
3. In a program, author is the only user of the program. If any problem occurs in the program and program crashes, then the author can fix the problem and can start using program again without any disturbance.
4. Program is the result of adhoc development where issues like portability and reliability are not considered.

1.3 SOFTWARE

Computers cannot perform any task/work without getting instructions from user through software. The combination of hardware and software is necessary if we want to get some task to be performed by the computer:

1. Software gives instructions to the computer what to do and how to do it.
2. Software is used for managing controlling and integrating the hardware components of the computer.
3. Software is not only the collection of computer programs. There is a difference between a program and software.
4. IEEE defines:***"Software is a collection of computer programs, procedures, rules and associated documentation and data."***

- ***Program:*** It is a list of organized instructions when executed behave in a predetermined manner.
- ***Procedures*****:** It is a step by step instruction to achieve a desired result. It explains how to prepare something.
- ***Rules:*** Rules are basically standards for actions/activities. Rule is basically a collection of guidelines to achieve something.
- ***Documentation:*** It is a set of documents provided in a particular format. It is written on paper, online or digital media.
- ***Data:*** Data is a raw material which is used in software. It is a set of values.

5. Software is basically developed by a team of developers who ensure the proper documentation.
6. Software developers ensure the well-designed interface because software are used by many people from different backgrounds.
7. Software is usually large in size and used by many people according to their needs.
8. Many resources and very many efforts are required for developing software.
9. Cost of developing software is very high because the software technology is still labor intensive and expensive.
10. Software projects are very large, involving many people and span over many years.
11. Software development is a systematic process where key issues like portability and reliability are considered.

1.4 DIFFERENCE BETWEEN PROGRAM AND SOFTWARE

Sr. No.	Concept	Program	Software
1.	Size	Size of program is small.	Size of software is large.
2.	Number of users	Single user	Large number of users.
3.	User interface	Program lacks in adequate user interface.	Software provide well designed and adequate user interface.
4.	Development	Program is the result of adhoc development.	Software is the result of systematic development.
5.	Number of developers	Single developer.	Team of developers.
6.	Documentation	User is the developer who makes little documentation according to his/her needs without using any format of documentation.	Software is the result of systematic work of team of developers. They provide proper documentation.
7.	Concentration on Issues	Issues like portability and reliability are not considered/concentrated.	Issues like portability and reliability are given proper concentration.
8.	Effort	Less effort is required.	More effort is required in the development of software as compared to program.
9.	Use of Resources	Less resources are required.	More resources are required as compared to program.
10.	Cost	Development cost of programme is less.	Development cost of software is approximately ten times more as compared to program.
11.	Complexity	Less complex.	More complex as compared to program.

Table 1.1 Difference Between Program and Software

1.5TYPES OF SOFTWARE

Software is basically of three types:

1. System software
2. Application software
3. Service or utility software

1. ***System Software:*** System software is a set of one more programs designed to control the operation and extend the processing capabilities of a computer system.

The efficient system software allows application software to be run on the computer with less time and effort. System software makes the operation of a computer system more effective and efficient.

System software helps the hardware components to work together and provides support for the development and execution of application software.

Some commonly known system software are as follows:

(i) **Operating System:** An operating system is software that runs on computer and manages the computer hardware. Operating system performs basic tasks such as recognizing input from keyboard, sending output to the display screen, keeping tracks of files and controlling peripheral devices like printers and responsible for security ensuring that unauthorized users do not access the system. Operating system acts as an interface between hardware and software.

(ii) **Language Processors (Translators):** Language processors or translators are system software which transfer (convert) the instructions written in a programming language, into a form which can be interpreted and executed by a computer system.

It is of three types:

- **Compilers:** A compiler is a software that accept the total program code (high level language) as input and then converts it into machine code (low level language). Compilers also detect and indicate certain types of errors in source programs automatically.

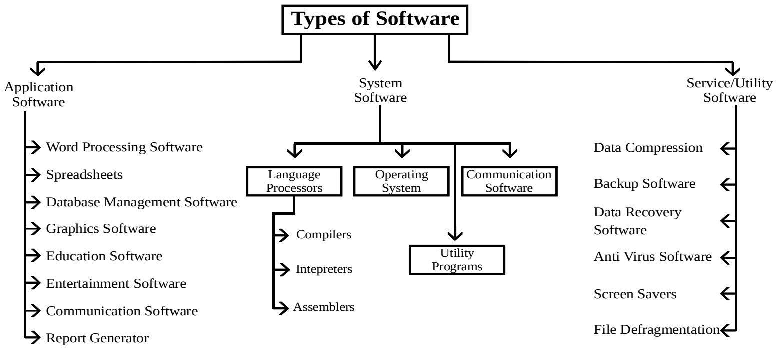


Fig. 1.1 Types of Software

- **Interpreters:** The interpreter takes one source program instruction, translates it into object code and executes it, then takes up the next instruction translates it and so on. Interpreters are slower than compilers.
- **Assemblers:** An Assemblers is a system software that takes as an input a program written in a assembly language and as an output generates the program written in machine level which can be directly executed on the computer.

(iii) **Communication Software:** Communication software enables to transfer the data and programs from one computer system to another computer system in a network environment where multiple computers are inter-connected together by communication networks.

(iv) **Utility Programs:** Utility programs is a set of programs which helps users in system maintenance and in performing tasks of routine nature like formatting of hard disks, taking backup of files, sorting records etc.

2. **Application Software:** Application software is a set of one or more programs, designed to solve a specific problem. There are wide ranges of applications ranging from simple applications like word processing, banking etc. to complex scientific/engineering applications like weather forecasting, design of aircrafts etc.

Some commonly known application software are as follows:

(i) **Word processing software:** Word processing software helps in day to day documentation work. It helps in creating texts, manipulating, formatting and printing of the text. It is considered as a computerized version of typewriter. It allows users to make changes and corrections in the text easily.

(ii) **Spreadsheets (Electronic Spreads Sheets):** Electronic spreadsheets are sheets of paper with rows and columns. Data is entered in a tabular form. The area where row and column meet is known as cell. User put text, number or formulae into cell. It also provides flexibility to user to make changes in spreadsheets. It provides various built in functions for the convenience of user.

(iii) **Database Management Software:** Database management system is a software package that allows a user to perform various operations like creating, deleting, updating, adding, modifying data in databases. Whereas database is an organized collection of logically related data. It makes retrieval of information easy and effective.

(iv) **Graphics Software:** Graphics software is a collection of programs which manipulate the images to any extent. These programs are referred to as point or draw programs in which user can draw illustrations from the scratch using an electronic pointing devices such as mouse, light pen or brush. Graphics programs are also known as point programs.

(v) **Education Software:** Education softwares are known as teaching and learning tool. These softwares are used to teach students. These softwares are very effective in mathematics, recognizing of alphabets, read whole words and sentences etc.

(vi) **Entertainment Software:** Entertainment software allows a computer system to be used as an entertainment tool. A good example of such an application is computer video games.

(vii) **Communication Software:** Communication software is used to share information and resources by connecting the computers. Communication software is also known as network software. A user can share information with another user with communication software.

(viii)**Report Generators:** Report generator helps a user to design his own report. This method is very easy and less time consumed. Manually design of these reports is very difficult task. Commonly available report generating software is oracle reports.

3. **Service or Utility Software:** Utility software is used to provide services to facilities the working of the operating system. It helps user to take backup of data, scanning virus and retrieval of deleted files.

Some commonly known utility software are as follows:

(i) **Data Compression:** Data compression is a ability to reduce the storage requirements of a file using mathematical algorithms. It helps to store more data on a disk. Commonly used data compression software areeasyip and WinZip.

(ii) **Backup Software:** The backup software helps the user to copy large group of files from hard disk to other storage media like floppy etc.

(iii) **Data Recovery Software:** It is also known as unerase software. It is required when user delete a file accidentally from a disk. After deleting a file, user realize that file is still needed and very important. This software gets back those deleted files which were deleted by the user.

(iv) **Antivirus Utilities:** The antivirus software track the virus, eradicate and prevent their spread. The antivirus scans the disk, identify any virus and attempts to remove them. Examples are McAfee, Norton antivirus etc.

(v) **Screen Savers:**Screen saver displays moving images on the screen if no input is received from the user for several minutes. A variety of screen savers are available ranging from flying windows to hollywood personalities and desktop themes.

(vi) **File defragmentation:** A file defragmentation is used to defragment the files to speed up the working of the hard disk. It is now part of the window operating system.

Comparison between system software and application software

Sr.No	System Software	Application Software
1	It is general purpose software.	It is specific purpose software.
2	System software is a superset of application software.	Application software is a subset of system software.
3	Examples: operating system, language processor, communication software.	Examples: word processing, software database management system, spreadsheets etc.
4	System software gets installed when the operating system is installed on the computer.	Application software gets installed according to the requirement of the user.
5	System software can run independently without the presence of the application software.	Application software cannot run without the presence of the system software.
6	It is essential for a computer.	It is not essential for a computer.
7	The number of system software is less than application software.	The number of application software is much more than system software.

Table 1.2 Comparison between system software and application software

1.6 **COMPONENTS/PARTS OF SOFTWARE**

The components of a software means those parts which complement a software product. Software needs other existing softwares for communication. These parts are necessary for the software development. The combination of these parts make a complete software.

The following are components of a software:

1. **Operating Environment:** Operating environment describe the suitable environment where software will operate on particular hardware platform in a peaceful manner.

2. **User classes and characteristics:** Various user classes are used to anticipate the software product. User classes are based on many aspects like frequency of use, technical expertise, security levels, educational level etc. The characteristics of user classes are also described in this section.

3. **Documentation:** Documentation is the text in the written form, which explains all the operating mechanisms/methods, functions and uses of the software. It provides the full-fledged information about the software. It explains complete information of the software in a written form. It is of three types:

(i) **Architecture/Design Documentation:** This document gives an overview of the software. This type of document gives the justification of the list things which are used in the development of the software. The design documentation also explains why such things are used in the software.

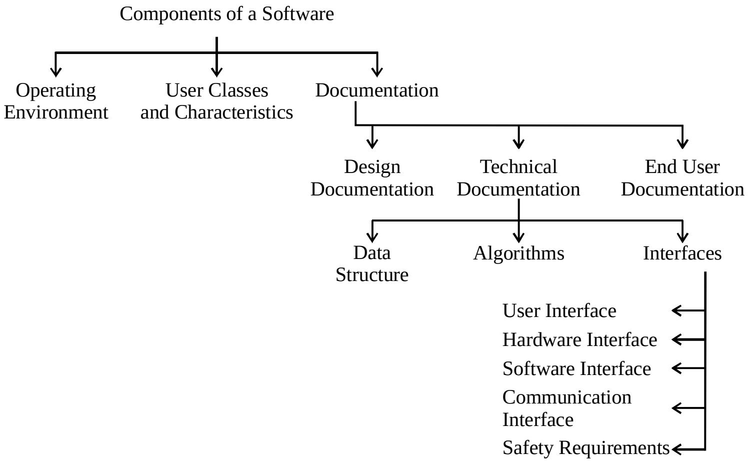


Fig. 1.2 Components of a Software

(ii) **Technical Documentation:** The technical documentation provides the coding part and various aspects of operations of the software. This documentation also explains the data structures, algorithms and interfaces:

- **Data Structure:** A data structure is a systematic way of organizing and accessing data. It explains how data are stored in a computer.
- **Algorithms:** Analgorithm is a procedure to solve a problem. The algorithm is the basic technique used to get the job done.
- **Interfaces:** It is a boundary across in which two independent system meet and act on to communicate with each other. The following are the types of interface:

(a) **User Interface:** It explains the characteristics of interface between the software product and users. It include the screen images, GUI standards, standard buttons and functions, keyboard shortcuts, error messages etc.

(b) **Hardware Interface:** It explains the characteristics of the software product and hardware components of the system. It include the supported device types, nature of data, communication protocols, control interactions between the software and hardware.

(c) **Software Interface:** It explains the connection between the software product with other specific software components, databases, tools, operating systems, messages coming into the system and going out. It also explains the communications nature and services needed.

(d) **Communication Interface:** It explains the communication functions of any product like email, communication protocols, electronic forms etc. It also describes the used communication standards like FTP or HTTP.

(e) **Safety requirements:** Safety requirements are concerned with damage, harm or loss which occurs during the usage of the product. It also explains some actions that prevents from these damages.

(iii) **End-User documentation:** This documentation helps the end users, system administrators and support staff in understanding the software. It provides all the features and steps required in the software. In short, it is a type of guide tutorials or a reference guide for a beginner. It should be very simple, manual and understandable so that end-user can easily understand it and use it.

1.7 CHARACTERISTICS/FEATURES OF SOFTWARE

The uses of computers are growing very rapidly. The computer industries are also growing very rapidly. The computer industries are very different from manufacturing industries in many aspects. Manufacturing industries are those industries who make/manufacture T.V.s, buildings, bikes etc. Computer industries develop the software not manufactured. Software is a logical entity not a physical entity. User cannot touch the software, cannot feel the quality of software.

The following are the characteristics of a software:

1. **Software is not manufactured or created, it is developed:** Software is first designed and then developed. Software is a logical entity, so it is developed not manufactured.Industries manufactured only those products which are in physical from, which can be touched. Software is not in a touchable form that is why it is developed.

Developers have number of automated case tools but quality of software depends upon the skills and creative abilities of the individual developers. Developers needs managing and controlling abilities to ensure the quality of the software product.

2. **Software is Highly Malleable:** Software is different and highly malleable as compared to other products. For example, we buy a new cabinet for old T.V. but old T.V. does not fit into this new cabinet. What we do? We call the carpenter to make necessary changes in the cabinet so that T.V. can fit into the new cabinet. On the other hand, if we want to change something in the software. We can directly modify the software itself rather than changing in the design of the software.

3. **Software does not wear out but deteriorates:** Software does not wear out means software is not influenced by the environmental changes. Software is a logical entity.

Software functions exactly the same way years after years without changing requirements over a period of time. If changes are necessary in the software to meet the user's new requirements, then there is a possibility of occurring of some defects which deteriorates the quality of the software.

Manufacturing companies replace the defected parts with the spare parts and maintain the quality of the product. But this cannot happens in the software industries. Software cannot replace with the spare parts.

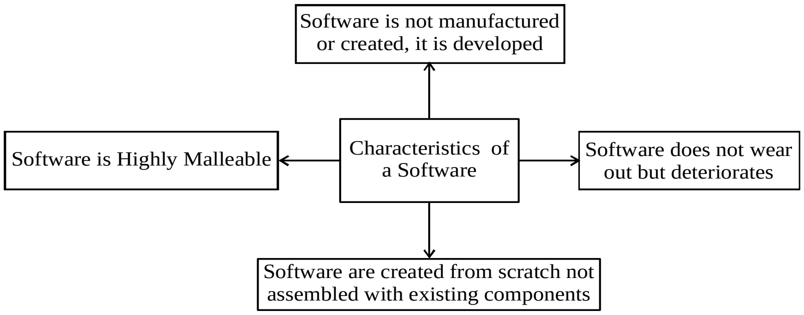

Software is not manufactured or created, it is developed
Software is Highly Malleable
Characteristics of a Software
Software does not wear out but deteriorates
Software are created from scratch not assembled with existing components

Fig. 1.3 Characteristics of a Software

4. **Software are created from scratch not assembled with existing components:** In manufacturing companies, various identifying components are put together to form the original masterpiece. Every component is flexible and independent in nature. But this approach is not applicable in software development. In software development approach, each module is inter-linked with another module. Each module is further divided into sub-modules. Different development teams are required for

different modules and sub-modules. Here modules are not assembled, modules are developed and inter-linked with each other to make a complete software.

1.8 APPLICATIONS OF A SOFTWARE

Software are used in many areas such as business, entertainment, medical, reservations etc. There is no area where software is not used.

The following are some areas where software are used:

1. **Business Softwares:** Softwares are used in business for fast, efficient and intelligent functions/tasks. With the help of softwares, companies make charts, excel sheets etc. which help in effective decision making. Companies also use software for visuals and advanced formatting.

2. **Entertainment Softwares:**Softwares are also used for entertainment purposes. There are number of entertainment softwares in the market. These softwares are extremely popular over the past decades. These softwaresincludes games, graphics, multimedia etc.

3. **Scientific Investigation and Engineering Software:** These softwares satisfy the requirements of scientists and engineers. These software are specially developed and designed for scientific and engineering purposes. These softwares are capable to perform complicated algorithms, programs etc. Astronomy and satellite launching etc. are related to scientific investigation and engineering softwares.

4. **Education Softwares:** Education softwares are known as teaching and learning tool. These software are used to teach students. These softwares are very effective in mathematics, recognizing of alphabets, read whole words and sentences etc. The primary purpose of education software is teaching or self-learning.

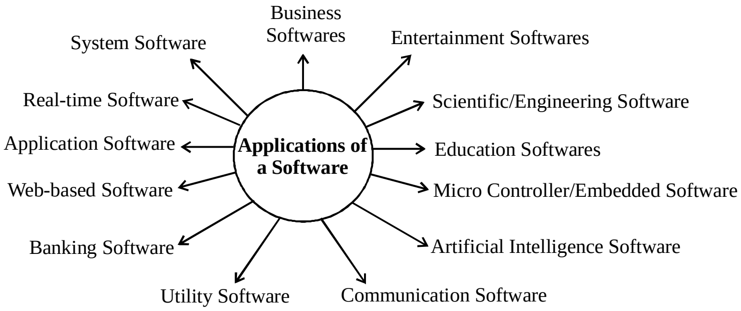


Fig. 1.4 Applications of a Software

5. **Micro Controller/Embedded Software:** Embedded software is designed for a particular kind of application service. It performs one or more predefined tasks. It is a combination of computer hardware and software. Embedded software refers to computer programs that directly interact with hardware. Example — Cars, Microwave, Watches etc.

6. **Artificial Intelligence Software:** Artificial intelligence software is an advance technique software. There softwares can understand a natural language, find out the reasons and their solutions automatically around the world. Artificial intelligence software requires human type of intelligence. Example—Expert system, Pattern recognition etc.

7. **Communication Software:** Communication software is used to share information and resources by connecting the computers. A user can share information with another user with communication software. With the help of modem and telephone line, a person can make contact with another person located at any place in the world.

8. **Utility Software:** Utility software is designed to help in management and tuning of computer hardware, operating system and application software. It is normally used to solve the common problems of software and hardware. The examples of Utility software are as follows:

- Disk defragmenters
- System profilers
- Virus scanners

- Application launchers
- Network managers
- Encryption utilities.

9. **Banking Software:** Banking software is an enterprise application solution that enables Banks to provide banking services and content to customers with efficieney and reliability within their bank premises. The banking software provides following features:

- Customer Management
- Transaction Management
- General Ledger & Accounting
- Accounting Transaction /Maintenance
- Rule Implementation / Maintenance
- Control / Configuration Management
- User-orientation
- Reporting
- Security
- Control / Configuration Management

10. **Web-based Software:** Web based software is an application package that can be accessed through the web browser. The software and database reside on a central server. Web based application gives us an opportunity to access the business information from anywhere in the world at anytime. It also facilitates us to save time, money and improve the interactivity with the customers and partners. It is also widely used for information exchange, entertainment and e-commerce.

11. **Application Software:** Application software is a computer software that makes use of the capabilities of a computer directly and thoroughly to a task that the user wishes to perform. Typical examples of software applications are word processors, spreadsheets and media players. Multiple applications bundled together as a package are sometimes referred to as an application suite.

12. **Real-time Software:** The term "real-time" refers to the ability to do or perform tasks right at the very moment they are said to be executed. Real-time software enables the user to execute various tasks and activities at the same time, as long as the programs are kept open. In computer systems, real-time operating systems contain a multitude of programs to run and operate even if the user is focused only on just one application. These software include media, building tools, computing and analysis applications, and system maintenance

13. **System Software:** System software is a type of computer program that is designed to run computer hardware and application programs. The operating system (OS) is the best-known example of system software. The OS manages all the other programs in a computer. Other examples of system software are BIOS (basic input/output system) program, the boot program, assembler, device driver, etc.

1.9**ATTRIBUTES/PROPERTIES QUALITIES OF A GOOD SOFTWARE**

Quality of a software depends upon different parameters. No single property fully describes the quality of a product. A software is called a good quality software if it possesses the following attributes:

1. **High Quality:** A good software product should possess the high quality means software should have minimum number of errors. The minimum number of errors make software more reliable.
2. **User Friendly:** Software should be user friendly. It should be usable by all types of user. Software should be easy in use. Software should provide all appropriate and adequate user interface along with the sufficient documentation.
3. **Secured and Safe:** Software should not cause any economic/physical damage. It should be secured and safe to work without any failure.
4. **Fit for intended job:** Software should fulfills all the requirements of the user. It should work according to the needs of the user and should be fit for the intended job. A quality software always fulfils the purpose of the user.
5. **Within the budget:**Software should be within the budget for the user. The cost of software should be within the packet of the user. Software should give profit to the computer industry.

6. **High Adherence (tolerance) power:** Software should respond to each and every type of input of the user. Software should behave reasonably even in the anticipated situations.
7. **Maintainable/Modifiable:**A good software should be maintainable and modifiable according to the situation. It should be changed to meet the new requirements of a user/customer.
8. **Efficiency:** Software should utilize resources in such a way that there should be no wastage of any resource. For example: memory and processor should be utilized up to the optimum level.

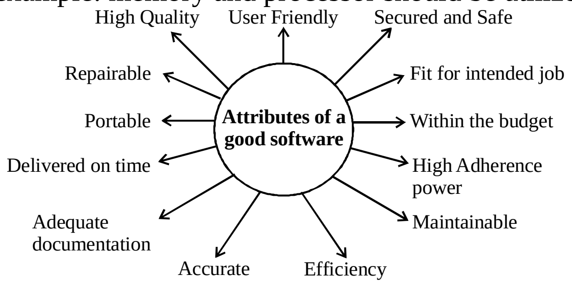


Fig. 1.5 Attributes of a good software

9. **Adequate documentation:** A software should have adequate documentation so that it can solve its purpose very well.
10. **Accurate:** Software should perform all tasks very accurately. Software should perform all operations according to the predefined specifications of the user and give accurate output to the user.
11. **Delivered on time:**Software should be developed and delivered within the specified time.
12. **Portable:**Software should be portable and have ability to work in different platforms.
13. **Repairable:**Software should be easily repairable. It should have privilege to correct the errors in a limited time period with the small amount of effort.

1.10 SOFTWARE **CRISIS**

Development of large software product with many essential features is the manual, error prone and complex process. The term software crisis is used in development of software. The term software crisis refers to the difficulty in developing a software like writing correct, understandable and verifiable computer programs. The main reasons of term software crisis are expectations, level of complexity and changeability. For example: every user demands large number of features in a required time and in a minimum budget. These expectations take the software into the software crisis.

"Software crisis means difficulty in writing correct, understandable and verifiable computer program."

Software crisis are the problems encountered in the development of the software. This is due to many issues like projects are not managed properly, late delivery of software, exceeds the budget for the development of software etc.

The following are the main reasons of software crisis:

1. Lack of communication between developers and users.
2. Lack of understanding of the problem.
3. Lack of adequate skills in the development staff.
4. Large and complex problems.
5. Increase in cost of software as compared to hardware.
6. Lack of adequate environment for developing the software.

The following are the some software crisis which are faced by many software industries:

1. Software is expensive.
2. Late delivery.
3. Software is not as per user requirements.
4. Difficult to modify/change and rework.

5. More chances of failures.
6. Brittleness.
7. Inadequate documentation.
8. Low quality of software.

In short, many software projects failed in late 1960's because these projects were late, over budget, complex, difficult to maintain and did not satisfy the client's requirements. This was known as software crisis.

Companies has been made many improvements and changes to solve these problems but problems are not decreasing. After using various methods and processes, companies are still facing new challenges and problems with the same old problems. Companies come with the conclusion that no single approach prevents the project/software over runs and failures in all cases.

In general, large complicated and poorly documented software projects have large, unanticipated problems. Whereas small projects have small problems. Researchers are still searching for the solution of the crisis.

1.11 **SOFTWARE MYTHS**

The term myth relates with mis-information or confusion. There are so many myths associated with software development. These myths affect the development of software indirectly. It means because of these myths, problems are arise in the initial stage of software development. There is no truth behind these myths. Some myths related to software development are as follows:

1. Software is good if it has many features.
2. Many software engineers cover up the delay.
3. Quality of software can be checked only after execution.
4. User requirements can be changed during any stage of software development process.
5. Reusing software increases safety.
6. Testing software can remove all the errors/bugs.
7. Software is easy to change.

1.12**SOFTWARE ENGINEERING**

1. Software engineering has different meaning and definition for many people. People explain software engineering according to their interest whereas experts give different definitions.

According to IEEE

"Software engineering is a process of systematic and disciplined approach for the development, operations and maintenance of the software."

According to Barry Bohem:

"Software engineering is the application of science and mathematics by which the capabilities of computer equipment we made useful to man via computer programs, procedures and associated documentation."

2. Software engineering ensures the development of good quality software in a minimum budget. It is the disciplined process which is concerned with all aspects of software production.

3. Software engineering follows quality standards like ISO 9001 and capability maturity model to maintain the quality in software. It focuses on various methods which are used in many software projects.

4. **Need of Software Engineering:** The goal of software engineering is to develop methods and procedures fear software development in a consistent manner to produce high quality software at low budget. Software engineering follows quality standards like ISO 9001 and capability maturity model to maintain quality in software. To achieve this goal, software development should be done in phases. Software processes should be designed and controlled properly to maintain consistency in development of software. The aim of the software engineering is to take the development nearer to the science and to concentrate on methods helpful in software development.

5. **Issues in software engineering:** The following are some issues/problems related to software engineering.

- **Technology and Management:** Developing a large software is very difficult as compared to small software. Large software needs adequate technology (methods, procedures and tools) and proper management so that software can be developed in a systematic manner in a required time without facing any problem.

Large projects are tightly managed to control the cost schedule and quality. Formal technology requirements of large software projects are high.

Whereas, small software projects can be developed with informal methods and technologies. For example: if we count the people in a room. It requires easy method and less time. But if we want to take a census of a country, then it requires proper formal methods and procedures for management and organization.

- **Consistency in terms of performance and quality:** Consistency of performance and quality is very important for all computer industries. The aim of computer industries is to develop a software with low budget and high quality in a consistent manner. Computer industries want consistent quality with improved productivity but it is very difficult task. A good quality software in a low cost is a challenge for computer industries.
- **Expensive:** Software development needs resources like hardware, software, manpower etc. in a low budget. But it is a challenge because software development is a labor intensive, time consuming and costly method.

1.13 EVOLUTION OF SOFTWARE ENGINEERING

Evolution of software engineering means take a look to check how software engineering been over the years. With the passage of years, scope and meaning of software engineering enhanced with number of issues.

During 1960, software was developed by a single programmer or by a small team of programmers. In this period, programmer can draw flow diagrams using language tools, write programming codesaccording to his/her choice. In 1970, team of programmers became larger on the basis of type of software project. Cost estimation was not considered but programming languages like FORTRAN and quality assurance got introduced. During the period of 1980’s and 1990’s, standards and methodologies were introduced for ensuring quality and reducing cost in software product.

Computer aided software engineering (CASE) companies use no of tools (Knowledge Based Systems and Total Quality Management) to improve the software development process.

1.14 SOFTWARE ENGINEERING — A LAYERED TECHNOLOGY

Software engineering follows the layered technology which are as follows:

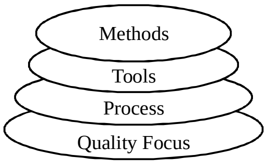


Fig. 1.6 Software Engineering — A Layered Technology

1. **Methods:**It provides technical support for building software. It encompasses many tasks including communication, requirement analysis, design modeling, program construction, testing and support.
2. **Tools:**Tools provide automated or semi-automated support for the process and methods.
3. **Process:** It is a framework with activities for effective delivery of software engineering technology. A process is a collection of activities, actions and tasks that are performed when some work product is to be created. The purpose of process is to deliver software in a timely manner and with sufficient quality to satisfy those who have sponsored its creation and those who will use it.

4. **Quality Focus:** Any engineering approach must rest on organizational commitment to quality which fosters a continuous process improvement culture.

1.15 ROLE AND RESPONSIBILITIES OF SOFTWARE ENGINEER

An engineer is someone who uses advanced knowledge of science, mathematics, and technology to build objects for use by others. Whereas software engineer is responsible for the complete life cycle of a new/modified software product from design to implementation. The roles and responsibilities of software engineer are as follows:

1. Software engineer is a good programmer and have a good knowledge of data structures and algorithms. Software engineer is expert in more than one programming language and have creative or logical thinking mind.
2. Software engineer is familiar with many design approaches and have ability to move among several levels of abstraction of different stages of the project, to a specific design for system and at the end to the detailed coding level.
3. Software engineer consult with systems analysts, programmers and others to collect the limitations, capabilities and performance requirements of software product.
4. Software engineer modify the existing software to correct errors, allow it to adapt to new hardware, or to improve its performance.
5. Software engineer analyze the user needs and software requirements to determine feasibility of design within time and cost constraints.
6. Software engineer coordinate software system installation and monitor functions to ensure specifications are met.
7. Software engineer analyze information to determine, recommend, and plan computer specifications and layouts, and peripheral equipment modifications.
8. Software engineer supervise the work of programmers, technologists and technicians.
9. Software engineer utilizes software engineering tools such as configuration management systems, build processes, and debuggers in the software development process.
10. Software engineer serve as a mentor to less experienced software engineers.
11. Software engineer also provides training to new users and handles support and feedback.
12. Software engineer produce well-organized, optimized, and documented source code. Contribute to technical design documentation.
13. Software engineer continuously learn and improve skills and can work independently when required.

1.16 SOFTWARE DEVELOPMENT LIFE CYCLE (SDLC)

Software development life cycle is also known as product development life cycle (PDLC). It is also called software development process (SDP). Software development life cycle has many phases and activities for software development. Each phase ends with a defined output which is the input for the next phase.

The goal of SDLC is to develop a high quality product in minimum cost by checking the progress at the end of each phase. In all the phases, complexity of tasks and project tracking etc. should be managed to maintain the quality of the software product. It should also help to solve the software crisis.

SDLC varies with large projects and small projects. Small projects have different phases, activities and issues as compared to large software projects. Large software projects should require proper documentation with user specifications whereas small softwares can be developed without written record of activities.

Key Points:

- In the SDLC, basically the major activities related to software development are performed.

- SDLC is a sequence of steps/phases to perform the well-defined activities to fulfill the desired goals. In SDLC output of one step/phase is the input of next step/phase.
- There is a criteria for each input/output phase. Input criteria specifies the conditions when to terminate the activities of the phase output criteria specifies the conditions when to terminate the activities of the phase.
- The goal of the SDLC is to detect the errors/defects in the starting of their origin in the phase. Validation and verification should be performed at the end of the phase to detect the errors.
- Due to importance of development process, various models have been proposed for developing a software which are being discussed in the this chapter. [2.3]
- Clear objective is defined in the starting of the development process to make it more effective and less expensive.

There are seven phase of SDLC:

1. Feasibility Analysis
2. Requirement Analysis
3. Design
4. Coding
5. Testing
6. Implementation
7. Maintenance

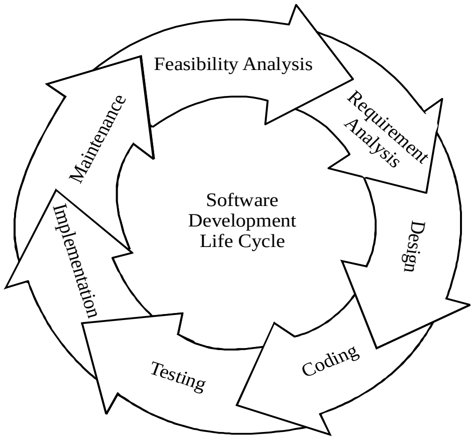


Fig. 1.7 Phases of SDLC

1. **Feasibility Analysis/Feasibility Study:**

- **Requirements Gathering:** The Business Analyst (BA) or a representator of a company collects all the predefined information of the company like what type of softwares a company can develop and in how many days etc. and goes to the client. Then BA collects all the basic requirements and specifications of the client and prepare a document known as BDD (business development design), BRS (business requirement specification), URS (user requirement specification), and CRS (customer requirement specification) etc. All these documents are same, only name is different. These documents are the output of the feasibility study phase and input of the requirement analysis phase.

- **Types of Feasibility Study:**

(a) **Technical feasibility study:** In the technical feasibility study, possibility of project development is measured. It is checked that whether the software project is technically sound or not.

(b) **Economic feasibility study:** In the economic feasibility study financial matters are discussed. For example : Is the software project is under budget or not? A software project should be under budget.

(c) **Schedule feasibility study:** In the schedule feasibility study, time duration is decided. The company finds the probability to complete the project in fixed allowed time period.

(d) **Motivational feasibility study:** In the motivational feasibility study, the sufficient motivation of organization is checked with necessary user participations, resources etc.

2. **Requirement Analysis:** The output of the feasibility study phase CRS (customer requirement specification) is used as input in the requirement analysis phase. Generally project manager analyze the requirements of the user and prepare the project plan.

Requirement analysis phase has 4 steps which are as follows:

(a) **Requirement Analysis:** All the requirements of client (mentioned in CRS) are studied and analyzed in detail. To understand the requirements of clients for large projects with many features, different tasks are analyzed. But it is very difficult because of communication gap between client and developer. This type of difficulty is based on basically two aspects. In one aspect, developer tries to develop a software which meets client's requirements without understanding the client's problem domain. In second aspect, client does not aware about the issues/problems in the development of the software.

(b) **Deciding Technology:** In this step, suitable technology is being declared for the development of software project. Companies select suitable technology based on many factors like requirements of clients, type of software project etc.

(c) **Estimating the Resources:** In this step, resources like manpower, time, money etc. are estimated for the development software project. All resources should be examined and improved in advance to reduce the difficulties in the software development.

(d) **Preparing SRS (Software requirement specification):**

- SRS (Software Requirement Specification) is a last and fourth step of requirement analysis phase. It is a document with the specific language to specify all the inputs, outputs, functional requirements, performance requirements with some constraints like economic issues, security etc.
- Definition: SRS is the specification for particular software product that perform certain functions in a specific environment.
- SRS is a complete description of the software which is going to be developed. It describes the complete scope of the product.
- SRS is prepared by Senior Analyst (SR).
- It is a type of agreement between the client and the company on what the software product is going to do.
- All functions/tasks of software development are based on the software requirement specification.

3. **Designing:** In the design phase, a plan is proposed for software development according to the software requirement specification. The quality of a software depends upon the design document. Design document is like a blue-print or plan for the software development which is used in later stages.

The design activity of a software development is divided into two levels.

(a) **High Level Designing/System Design**

- It is also known as top-level design. It is done by Technical Manager (TM).
- In this designing, software is divided into number of modules. Modules are identified along with their specification and interaction among them. In this type of designing, focus is on identifying the modules.

(b) **Low level designing/detailed design:**

- It is also known as detailed design. It is done by Team Lead (TL).
- In this designing, modules are further divided into number of sub-modules.
- Internal logic of sub-modules are specifies in the high level descriptive language.
- The output of this phase is detail design document (DDD) which is further used in later phases to develop a high quality software.

4. **Coding:**

- Once the designing of software is complete, the coding phase comes, in which developers follow some coding standards for writing the programs.
- In this phase, design of a software product translates into simple and understandable code in a particular language.
- Developers prepare the source code which is easy to write, read and understand.
- Coding should be done with simplicity, clarity and well-written format. Effective coding can reduce the testing and maintenance effort.
- Coding part is very time consuming and difficult task. Coding should be done in structured programming which linearizes the control flow in the programs.

5. **Testing:**

- The quality of a software product is measured and controlled by the testing phase.
- Testing is a process of executing a program with the intent of detecting or finding errors. Proper planning is required for the testing.
- Testing is very difficult and time-consumingprocess.

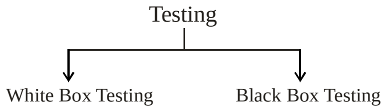


Fig. 1.8 Types of Testing

- Before starting this process, test engineers study the CRS (customer requirement specification) document prepared in the first phase and prepare a review report.
- In the review report, test engineers mentioned those points which are not clear and understandable. Test engineer send this review report to the Bussing Analyst (BA).
- Test Engineer prepare test plan before testing. Test plan specifies the scope, approach, resources and schedule of all testing activities.
- Test plan covers all the activities like items to be tested, types of testing to be performed, features to be tested, persons/resources responsible for testing, time scheduling issues/risks associated with the plan etc.
- Testing starts with the unit testing, in which a module is tested separately then integration testing is performed, which detect design errors and in last, system testing is performed to check the system performance according to the software requirement specification (SRS).
- The output of the testing phase is known as defect profile document (DPD). Test Engineers sends this document to the team of developers for removing the errors and controlling the quality of software.

6. **Implementation:**

- In this phase, after completing the software project, a mail is sent to the client. This mail is known as software delivery note.
- It after receiving this mail, client test the software project known as user acceptance testing. After this successful testing, software is installed in the client's environment.

7. **Maintenance:** During installation of software product in the client's environment, if any problem occurs or if client wants to make some changes in the software, then the maintenance people make changes to solve the problem and prepare the deployment document (DD) for further use.

1.17 SOFTWARE DEVELOPMENT PROCESS MODELS

1. A software development life cycle models describe the necessary sequence of different activities helpful in software development.
2. Software development starts with the request of client and undergoes many stages until software product is developed and delivered to the client.

3. Thses models show series of identifiable activities of software development for the successful delivery of the software product.
4. In the SDLC models, several activities and several documents are carried out. Documents are normally made to collect all the input/output information. Different phases of software development are shown graphically with the textual description for easy understanding.
5. In short, these models give descriptive and diagrammatic representation of software development phases.
6. There are so many software development life cycle models like waterfall model, spiral model etc. The common aim of these models is to develop a good quality software.
7. No matter which life cycle model is followed, all models have basic activities, carrying process of these models may be in different orders, but goal will remain same.
8. These models are helpful in developing a high quality software with low budget and time constraints.

There are so many life cycle models. The following are some life cycle models commonly used by computer organizations:

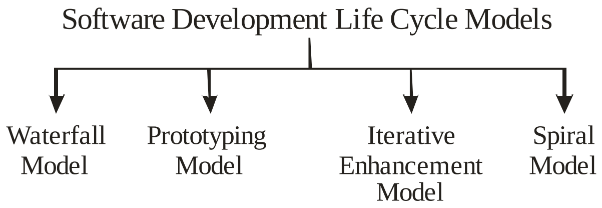

Software Development Life Cycle Models
Waterfall Model
Prototyping Model
Iterative Enhancement Model
Spiral Model

Fig. 1.9 Models of SDLC

1.18 **WATERFALL MODEL/LINEAR SEQUENTIAL LIFE CYCLE MODEL**

1. Waterfall model is the simplest and oldest process model.
2. It is represented by sequence of different stages.
3. The output of one stage flows into the second stage, the output of second stage flows into third stage and so on.
4. It is a definite structure of software development life cycle. Each phase is distinct and has a well-defined entry.
5. There is a provision for verification, validation and error correction at the end of every phase.
6. The part of waterfall model between the feasibility study and system testing is known as development part. After the development part, the software product is delivered to the customer. The maintenance phase comes after the delivery of the developed product to the customer.
7. This model is also known as the linear sequential model or the software life cycle model.
8. The waterfall model consists of following six stages:

(i) **Feasibility Study:** The feasibility study determines whether the developing model is financially and technically feasible or not. It is necessary to analyze the problem. More we understand the problem and the better we can identify alternative solutions. The feasibility study is usually done within limited time bounds. The outcome of feasibility study is a document that should contain at least the following factors:

- A definition of the problem.
- Determination of technical and economic viability.
- Alternative solutions and their expected benefits.
- Requirement of resources and detail about delivery

This report is called as feasibility study which is prepared by a group of software engineers. The client or the customer is also consulted through questionnaires. The report determines whether the project is feasible or not.

(ii) **Requirement Analysis and Specification:** In this phase, we exactly analyze the requirements and needs of the project. The purpose of a requirement analysis is to identify the qualities required for the application, in terms of functionality, performance, ease of use portability, and so on. The result of this phase is known as the software requirement specification (SRS) document.

An SRS document must contain the following items:

- Detailed statement of problem.
- Possible alternative solution to problem.
- Functional requirements of the software system.
- Constraints on the software system.

The SRS document must be precise, consistent, and complete and covers features like problem statement, introduction to the problem, functional requirements of the system, non-functional requirements of the system, behavioral descriptions and validation criteria.

(iii) **Designing:** The objective of the design phase is to convert the requirements specified in the SRS document into a structure that is suitable for execution in some programming language. Two different design approaches are available: the traditional design approach and the object-oriented design approach.

- In Traditional Design Approach, two activities are carried out, first a structured analysis of the requirements specification and second structured design. During structured design, the results of structured analysis are transformed into the software design.
- In Object-Oriented Design Approach, the various objects and their relationships within the system are identified.

(iv) **Coding and Unit Testing:** Coding and unit testing is the phase in which we actually write programs using a programming language. The output of this phase is an implemented and tested collection of units. The unit testing is the testing of code to check correctness and remove the bugs. The debugging is a related activity performed in this phase.

(v) **Integration and System Testing:** During the integration and system testing phase, the modules are integrated in a planned manner. The different modules are integrated to develop a software product. During each integration step, the partially integrated system is tested finally, when all the modules have been successfully integrated and tested, system testing is carried out. The objective of system testing is to determine whether the software system performs per the requirements mentioned in the SRS document.

The system testing is done in three phases: Alpha, Beta, and Acceptance Testing.

- **Alpha Testing** is conducted by the software-development team at the developer's site.
- **Beta Testing** is conducted by a group of friendly customers in the presence of the software-development team.
- **Acceptance Testing** is performed by the customers themselves. If the software is successful in acceptance testing, the software product is installed at the customer's site.

(vi) **Delivery:** The delivery of software is done in two phases. In the first phase, the application is deployed on few customers' sites. The purpose of this procedure is to check the product performance at client site, on the basis of feedback from users, whether any changes is necessary or not. In the second phase, finally the product is distributed to the customer's officially.

(vii) **Maintenance:** The maintenance as the set of activities that are performed after the system is delivered to the customer. Its purpose is to monitor the performance of the software, and removing error if nay faced by the customer in site.

Feasibility Study

↓

Requirement Analysis and Specification

↓

Designing

↓

Coding and Unit Testing

↓

Integration and System Testing

↓

Delivery

↓

Maintenance

Fig. 1.20 Steps of Waterfall Model

Advantages of Waterfall Model

The various advantages of the waterfall model include:

1. It is a linear model.
2. It is a segmental model
3. It is systematic and sequential.
4. It is a simple one.
5. It has proper documentation.

Disadvantages of Waterfall Model

The various disadvantages of the waterfall model include:

1. It is difficult to define all requirements at the beginning of a project.
2. The model is not suitable for accommodating any change in later stage.
3. It does not good for large scale projects.
4. It involves heavy documentations.
5. We cannot go backward in the SDLC.
6. There is no prototyping model for clearly realizing the customer's needs.
7. There is no risk analysis.
8. If there is any mistake in any of seven phases then we cannot develop good software.
9. It is a document driven process that requires formal documents at the end of each phase.

1.19 **PROTOTYPING MODEL**

1. Prototyping model is an intuitive approach of the waterfall. It overcomes the limitations of waterfall model.
2. Multiple development cycles take place in prototyping, making it multi-waterfall cycle. Cycles are further divided into smaller and easily manageable iterations.
3. The interactions in the prototype enable the developer to better understand what needs to be done and to satisfy users.
4. The prototype model is helpful in understanding the currently known requirements. In the waterfall model, client has to specify his/her requirements before the designing and coding phase. Client's requirements are freezed in the requirement phase. But in the prototyping model, client can specify his/her requirements in more detailed manner.
5. In this model, the client can interact with the prototype and can get an actual feel of the system.
6. Prototyping model is suitable for those projects where it is very difficult to determine the specifications of complex and large projects like client's feedback regarding software product, what is correct and missing, what needs modification etc.

7. Developers make changes in the software product on client's suggestions. Clients use the new changed software product again and allowed to give his/her suggestions to the developers. This cycle repeats until client satisfies with the software product.
8. Prototyping ensures that end users constantly work with the system. Prototyping model is excellent for designing good human computer interface systems.
9. The steps of prototyping model are as follows:

(i) **Requirements gathering and analysis:** The model begins with requirement gathering and analysis in detail. In this phase, requirements are gathered and analyzed by the organization.

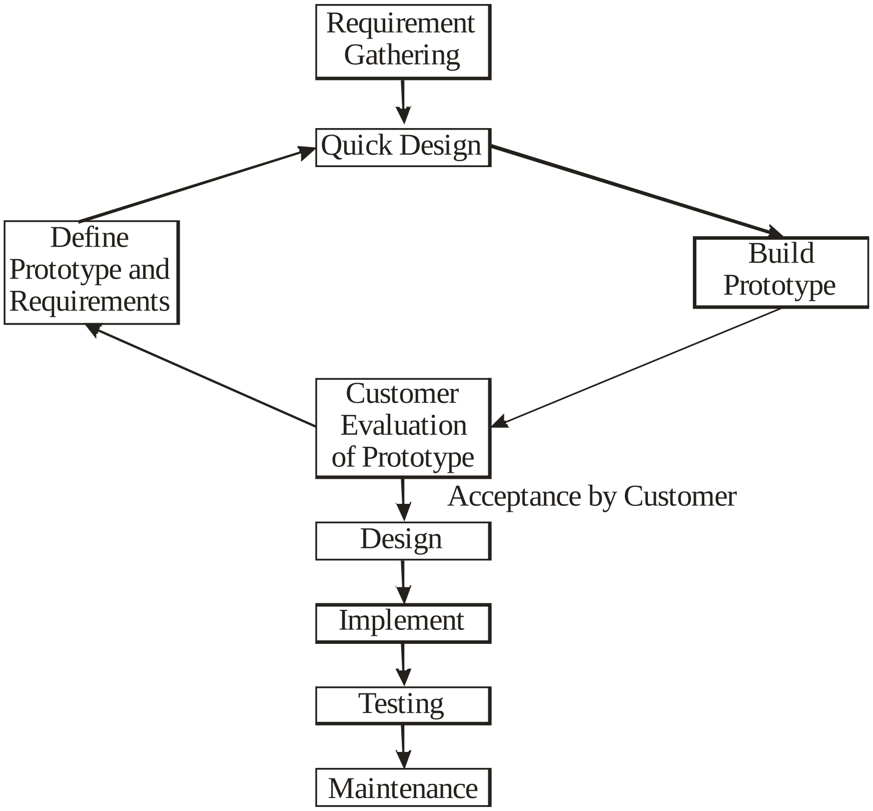
(ii) **Quick Design:** After analysis of requirements, quick design is created. It is a rough design not a detailed design. This quick design helps in developing a prototype and provides an idea of the software product to the user.

(iii) **Build Prototype:** With the help of quick design, a prototype is build which represents the working model of the required software product.

(iv) **User Evaluation:** A proposed prototype is evaluated by the user. User studies the strengths and weaknesses of this prototype. User provides his/her comments and suggestions like what is to be added or removed etc. to the developer.

(v) **Refine Prototype:** If after the evaluation of the prototype, user does not satisfy, then prototype is refined according to the requirements. A new prototyped is developed and being evaluated by the user. After the satisfaction of user, final system is developed.

(vi) **Engineer Product:** Once the user accepts the software product is maintained on regular basis.



Prototyping Model

Fig. 1.21 Prototyping Model

Advantages of Prototyping Model:

The various advantages of the prototyping model are as follows:

1. **Quick development:** Developer's approach is to generate software product quickly according to the client's requirements.
2. **Flexible:** Prototyping models are more flexible as compared to waterfall model. Clients can give suggestions to developers for the changes in the software product. In this way, model reduces the communication gap between client and developers.
3. **Easy to test and debug:** Developers can easily test and debug a smaller iteration rather than the whole software product.
4. **Manage risk:** During the small iteration, risky pieces are identified and handled easily.
5. **User Involvement:** Users are actively involved in the software development process.

Disadvantages of Prototyping Model

The various disadvantages of the prototyping model are as follows:

1. Each phase of iteration is rigid and do not overlap each other.
2. **High Cost:** Experienced developers and changing requirements are two main reasons of higher cost in prototyping model.
3. **Lower Quality:** During the development of software product according to prototyping model, the developer's focus is on quick development without thinking about the quality of software product. Development approach is to satisfy client's needs without bothering about the quality of software product.
4. **Compromises in Implementation:** Developer's aim is to get prototyping quickly and to meet client's desires. To achieve this aim, developers make many implementation compromises like inappropriate operating system, an inefficient algorithm etc.
5. **Time Consuming Process:** It is a slow process. The aim of prototyping model is to satisfy the customer. But requirements of customer keeps changing according to external environment. Hence,it is time consuming process.

1.20 ITERATIVE ENHANCEMENT MODEL

1. Iterative enhancement model provides the benefits of both waterfall model and prototyping model.
2. According to the iterative enhancement model, each increment adds some functional capability to the system until the full system is implemented.

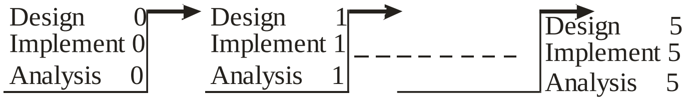


Fig. 1.22 Iterative Enhancement Model

3. The basic idea behind this model is to develop a system through repeated cycles (iterative) and in smaller portions at a time.
4. This model is used when requirements of system are clearly defined and understood.
5. Iterative enhancement model supports changing requirements.
6. Parallel development can be planned and progress can be determined during iterative enhancement model.

Advantages of Iterative Enhancement Model

1. Iterative enhancement model is better in testing. It is because testing increment is very easy as compared to testing the entire system. It overcomes the limitation of waterfall model.
2. Like prototyping, this model also provides feedback to the client. This feedback is very useful for determining the final requirements of the system.

3. This model also provide results periodically and early. It allows the users to get some early experience of the software.
4. Risks are identified and managed during iteration. During smaller iteration, testing and debugging is easy.
5. This model is suitable for large and mission critical projects.

Disadvantages of Iterative Enhancement Model:

1. This model is not suitable for small projects.
2. During iterative enhancement model, management complexity is more and highly skilled resources or required for risk analysis.
3. Sometimes, the iteration may never end and the user may never really get the final product.

1.21 **SPIRAL MODEL**

1. Spiral model was developed by Barry Bohem in 1988. In this model, software is developed in spiral with many loops.
2. The number of loops in a spiral model are not fixed and can vary from project to project. Each loop is the phase of the software project.
3. At each loop, risks are identified and resolved by prototyping. At every spiral, risk analysis is done to evaluate the development efforts/labour and risks.
4. This model is very flexible because number of phases are not fixed for the software development.
5. Inner cycles/loops represent the requirement analysis alongwith the prototype, whereas outer cycles represent the waterfall model.
6. Generally, spiral model is combination of strengths of various models. It uses stepwise approach of waterfall model and prototyping for resolving the risks occurred during software development.
7. Spiral model is also called as metal model because both waterfall model and prototype model are used in it. It reduce risk as well as follow systematic approach.
8. **Phases of Spiral Model:** There are four phases in the spiral model. In the starting of this model, requirements are gathered to analyze the risk. Spiral model makes communication effective between customer and developer. In this model, angular component represent the progress and reduces the represent cost.

The four phases of spiral model are discussed as follows:

(i) **Planning:** In the planning phase, requirements are gathered. All methods and strategies are decided in this phase. The objectives, constraints and alternatives are also discussed in this phase.

(ii) **Risk Analysis:** A process is undertaken to identify the risks. In this phase, alternate solutions are also identified to resolve the risk. Prototyping is used at the end of this phase to resolve the risks which are identified and analyzed. In this phase, risks associated with these new alternative solutions are also analyzed. In this phase, keeping new risks in mind, strong decisions are taken to resolve the risks with prototype.

(iii) **Engineering:** It involves the actual development of software project. After finding and resolving all the risks through prototyping, software is being developed in the engineering phase.

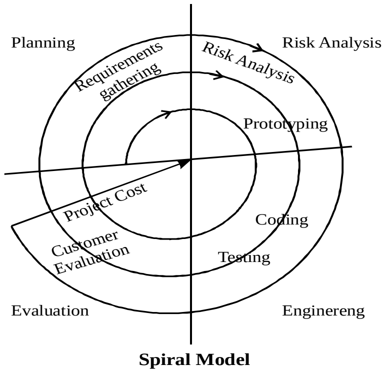


Fig. 1.23 Spiral Model

(iv) **Evaluation:** This phase allows the customer to evaluate the output of the project to date before the project continues to the next spiral.

Advantages of Spiral Model

1. Spiral model evaluates the risk analysis. It enhances the avoidance of risk.
2. It is good for large and mission critical objects.
3. Software is produced early in the software life cycle.
4. Strong approval and documentation control.

Disadvantages of Spiral Model

1. Sometimes, it is costly in use.
2. It is not suitable for small projects.
3. The success of spiral model depends upon the risk analysis phase.
4. Risk analysis requires highly specific expertise.
5. It is not suitable for low risk projects.
6. Spiral may continue indefinitely.

M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 2: SOFTWARE PROCESS

2.1 INTRODUCTION

Software is developed through a set of activities which are essential for software development of software; a collection of procedure is adopted to achieve certain goals or standards. Therefore the process that deals with technical and management issues of software development in mind is known as software process.

Computer software is loaded into the computer memory and the computer becomes capable of operating the software. The instructions are passed from application software to the hardware through the systems software. The following points are important to be noted about software process:

1. Processes use resources which under terms and conditions product intermediate and final product.
2. Processes are made up of sub processes and each sub process has its own process model.
3. A process has entry as well as exit criteria which controls and monitors the beginning and completion of an activity.
4. A process includes guidelines and it forces uniformity in all activities.
5. Many technical and managerial issues are needed to develop a software.

2.2 CHARACTERISTICS OF SOFTWARE PROCESS

A software process has so many characteristics which have been listed below:

1. **Comprehensiveness**: The process should be easy to understand the user.
2. **Visibility**: The progress of the process should produce clear results.
3. **Robustness**: The process should continue even if there is a unexpected problem.
4. **Supportability**: The process should be such as to support the process activities. It should also make the testing easy and it should be able to produce the software easy to maintain.

5. **Acceptability**: The process should be acceptable to the engineers and it should be usable.

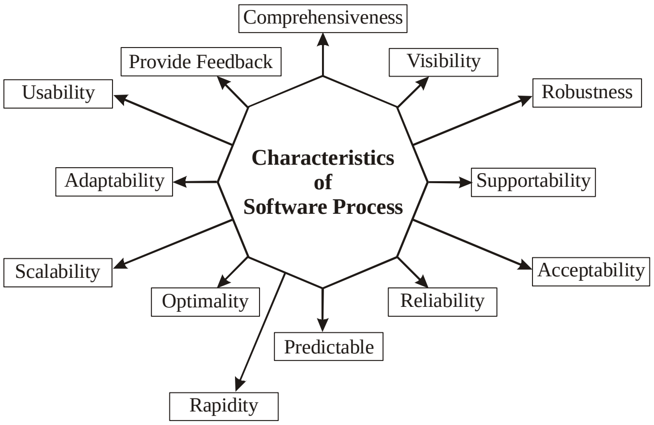


Fig. 2.1 Characteristics of Software Process

6. **Reliability**: The errors in the processes should be avoided or trapped before they result in errors to the product.
7. **Predictable**: The outcome of the project should be predicted before the project is completed.
8. **Rapidity**: The process of delivering the system with given specifications should be rapid.
9. **Optimality**: The process should be able to produce high quality software at a very cheap rate.
10. **Scalability**: It should be applicable for large scale software projects.
11. **Adaptability**: The process should adaptable to the changing requirements.
12. **Usability**: It should have proper user interface and adequate document.
13. **Provide feedback**: There must be feedback information for improvement.

2.3 **PROJECT MANAGEMENT PROCESS**

2.3.1 **Definition**

It can be defined as an activity to achieve an objective through the wisely use of knowledge, skills, tools and techniques by the Project Manager.

Project Management is an artistic use of various resources so that the project is completed on time and within budget. It should provide satisfaction to the End user. Project Management is combination of control, Leadership, Team work as a unit, resource management etc.

Process Groups

For the successful completion of a project; a proper system of different activities need to be followed. These processes can be placed into five inter-related process groups: 1. Planning, 2. Organisation, 3. Staffing, 4. Directing, 5. Controlling.

The main job of management is to enable a group of professional to work towards a common goal.

1. **Planning:** This is an important part of project management. The following points are kept in mind by the project management.

(a) What objectives are to be achieved?
(b) What resources are needed?
(c) How and when the resources are to be acquired?
(d) Ways and means to achieve the goal.
(e) To determine the flow of information, people and products within the system.

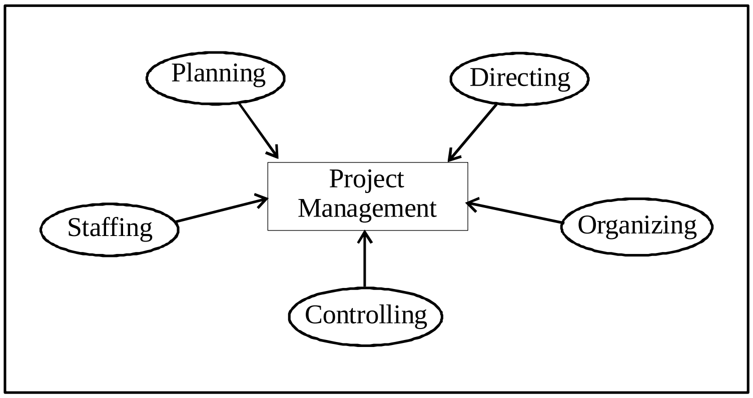
2. **Organizing:** It involves the division of duties and responsibilities among different groups in such a way that the goals of the enterprises are achieved. This is possible only when proper planning is done by the professional within the groups. Best organization means best results in the end.

3. **Staffing:** It deals with hiring of qualified and well trained personnel for the positions that are recognized by the organizational structure. It carries the process of recruiting, hiring, training, rewarding, retaining, compensating, developing and promoting employees.

4. **Directing:** It is the process guiding and leaving the subordinates so that they may understand and identify the organizational structure and the goals to be achieved by the enterprise.

5. **Controlling:** This step includes the process of measuring and correcting activities. The performance of each group is measured against plans and if there is any deviation, then corrective actions are taken to achieve the desired goal.

Software Project Management is based on sound technical skills and Management Skills. The Technical Process of software development depends on the application of sound management principles. It is not possible for one person to handle today's large and complex programs therefore team work is necessary under the leadership of the project manager.



Components of Project Management Process

Fig. 2.2 Components of Project Management Process

2.3.1.1 **Characteristics of Project**

A project is a temporary endeavor which is taken to provide a unique product or service. A project is different from production work in the sense that all projects have a beginning and an end. Projects are of different size. If can be quite small such as the repairing of a sofa set. Projects can be too big involving thousands of people and investing large sum of money. A project can take place at any level of organisation or even can include the whole organisation. A project can be of small duration (days or weeks) or long duration (to be completed in several month years).

Projects are of two types:

(i) **Temporary Projects:** Each and every project has a beginning and an end. The project can be closed due to the following reasons:

(a) Project's objective is achieved.
(b) When if become, clear that it is difficult to achieve the objectives in a practical way.
(c) The project is no longer required now.

(ii) **Unique Projects:** A project becomes unique when if tries to improve upon what was done in the past. Everything is planned and organized to bring a new product with new specifications.

A successful completion of a project demands better planning, well defined organization technical monitoring, project scheduling, cost, time and effort estimations. The proper measurements and metrics make a project worth its price and qualities. Composition of teams, assigning different tasks, providing resources and establishment of project standards are the major characteristics of a good project

2.3.1.2 **Project Manager Manages the Project**

Project Manager

The project manager is the most important individual who uses both the process and set of tools and techniques to define the aims and objectives of the project. The project manager plans the work to achieve the goals. He leads the project and support teams, monitors progress and so manages everything that the project is completed in a satisfactory way.

The software project manager keeps an eye on day by day operations. He can have the responsibility for multiple projects. The need for a good project manager has risen properly in the industry and commerce. Project manager is the central axis of the project-based organizations because it is establishing itself both as a professional career path and a way of controlling business.

The main responsibility of the project manager is to develop project plans including the work breakdown structure, time estimated, planning resources, training plan, risk management plans and identifies other plans. He uses PERT chart/Gantt chart etc. for scheduling all the tasks. **The main responsibilities of the project manager are:**

1. Define Goals: He defines goals, plans and monitor's tasks and resources. He recognizes and find solution to the issues and in addition to it controls costs and budgets.
2. Leads System Development: The project manager showed have a good understanding to the project management task and he leads system development on an expert system developer.
3. Manages and organizes the primary resources needed to complete the project. He interacts with people linked to the project and mobilizes resources and equipment very well.
4. Control and co-ordinates every aspect of the project and motivates the team members to achieve the project objectives.

To conclude we can say that the project manager, plays a vital role in selecting everything and organizes, plans and controls the development process. He communicated with different persons, participants in meetings and gives presentations.

2.3.2 Four P's in Project Management

A successful venture is the one where the project manager successfully builds the right product using the right process for the right project with the involvement of efficient people.

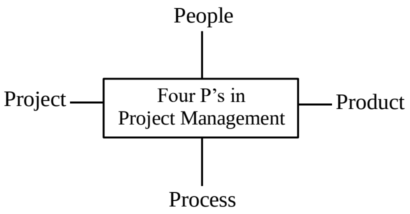


Fig. 2.3 Four P's in Project Management

1. **People:** The skillful and efficient people are the greatest asset of a software organization. The people represent the intellectual capital and efficient managers get best out of their investment in people. Success comes to those organizations which respect the people in the project team. The following people are involved in the software process:

(i) Senior Managers: Defines the business issues.
(ii) Project Managers: Who man, motivate, organize and counting the programmers?
(iii) Programmers: Who deliver the technical skills needed to develop a product?
(iv) Customers: Who specifies the requirements?
(v) End users: Who actually interact with the software once it is developed?

The people who work in a project have to work as a cohesive group and contribute their best to the project. They will have to be organized because the team structure has a direct influence on the product quality and project productivity. The project manager has to solve a lot of problems with the help of the people in his team in the most effective manner. He has to motivate plan and organize the work of the programmers for the proper functioning of the work. The managerial activities of the project manager and his leadership skills have a great influence in building a letter team. He must have the client management skills too. He should be able to communicate effectively with his clients.

2. **Product:** The software product is a tangible item that is produced by a project according to the required specifications. The project manager has to identify opportunities for the use of the product. He must explore the idea about the product in detail. For this scope of the project must be established.

Software Scope: The software product must be built in such a way to fit into a larger system. What constraints are needed must be known in advance. Software project scope must be unambiguous and understandable at the management and technical levels. Quantitative data are stated explicitly, Limitations are noted and mitigating factors are described.

Problem Decomposition: It is an activity that is important for software requirement analysis. Decomposition of the problem is applied in two major areas:

(i) The functionality which must be delivered

(ii) The process which will be used to deliver it. Software functions are evaluated and refined to furnish more detail prior to the beginning of estimation. As cost and schedule estimate are functionally oriented; some degree of decomposition is often useful.

3. **Process:** The project manager must decide to select the process model which is most appropriate to be produced by the project team the first requirement is to build the project environment in while the software team works. The attributes of the product must be kept in mind. After the selection of the process model, the preliminary plan is established and process decomposition begins the organization adopts a set of framework activities which include:

(i) Customer communication

(ii) Planning

(iii) Risk analysis

(iv) Engineering activities

(v) Documentation and Training

(vi) Customer evaluation etc.

A software team must have the degree of flexibility in selecting the software engineering paradigm which suits well for the project. A relatively small project similar to the post efforts is chosen. If the deadline is very tight then an incremental strategy is adopted. Similarly projects with other characteristics are selected. In the end a more complex project with a broader scope and significant business impact is adopted.

4. **Project:** The software project management activity is just like an umbrella activity within the software engineering process. It begins before the start of any technical activity and continues throughout the definition, development and support of computer software. The project management activities include:

(i) Measurement and Metrics

(ii) Management activities

(iii) Project planning

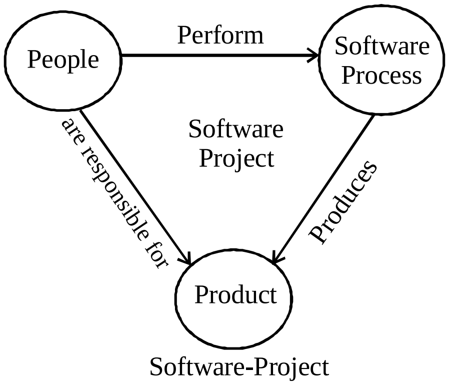
(iv) Scheduling

(v) Tracking

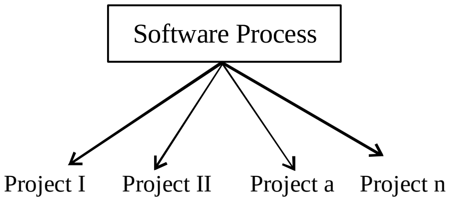
(vi) Risk management activity

To avoid problems in the project activities, the project team follows the five past common-sense approach i.e. (i) Start on the right tool, (ii) Maintain Momentum, (iii) Track process, (iv) Mask smart data.

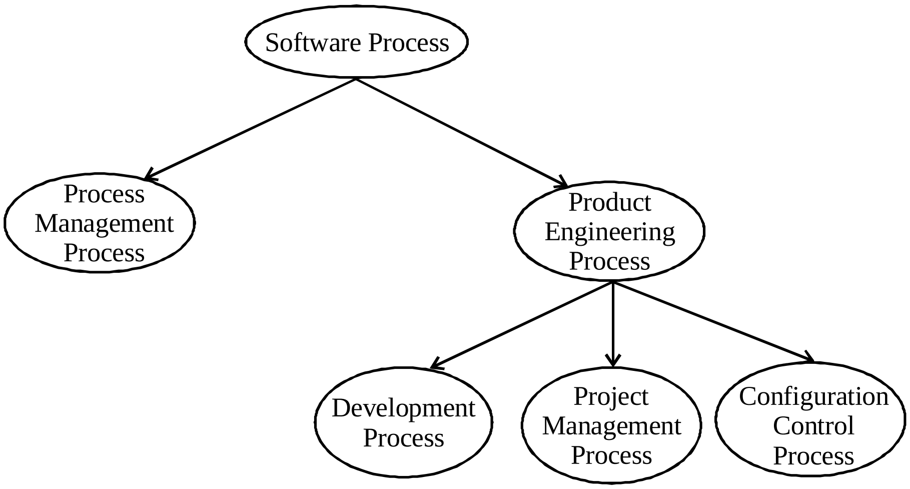
The inter relationship between 4 P’s is best illustrated with the diagrams as under:



(A) Inter-Relationship between 4 P's



(B) Inter-Relationship between Process & Project


Software Process
Process Management Process
Product Engineering Process
Development Process
Project Management Process
Configuration Control Process

(C) Process Categories

2.3.3 Activities of Project Management

The following are the major activities which are to be performed by the Project Manager for successful management of the software project:

1. **Proposal Writing:** These are no set rules or guidelines for proposal writing. It only depends on the professional skills and experience proposals describe the aims and objectives of the project and the manner in which it will be carried out. If the proposal writing is effective; only then it will be accepted and contract will be awarded.

2. **Project Planning:** It is just like a Guide Map which guides the development team towards the achievement of goals. Project plan keeps the following points in mind.

- Identifying different activities
- Setting of Goals
- Identifying the final result in the form of deliverables to be produced by the project.

3. **Project Scheduling:** It involves the following things

- Breaking the main project into smaller tasks and identifying activities
- Estimating resources for these activities
- Judging the time required to complete these activities
- Allocation of people to different activities and coordinating their tasks.
- Full utilization of human resources

4. **Project Cost Estimation:** This activity is concerned with guessing and calculating the resources required completing the project plan.

5. **Project Monitoring:** In this activity the project manager check the progress of the project from time to time and compares the actual and planned progress on costs. If there is any deviation then project manager takes corrective steps.

6. **Project Review:** This activity is concerned with observing the overall progress and technical development and to study whether the project's status is in accordance with the aims and objectives of the organization commissioning the software.

7. **Personnel Selection and Revaluation:** The project manager selects a balanced team of experienced employees and youthful fresher's. It is necessary that the services of the staff are utilized effectively.

For selecting personals, a project manager has to keep in mind the following steps:

- Project Budget
- Impossible to appoint new staff
- Providing Training to employees

8. **Project Report Writing:** A Project Manager prepares Project Report. He is responsible to submit this report to both the client and contractor organization. He must have the written as well as oral skills to present his report in a very effective way. The language of the project report should be very simple and clear to understand.

2.3.4 **Issues in Project Management**

There are so many issues which are to be kept in mind in the Software Project management. There are specific goals that are to be achieved. Certain Laws and Principles have to be followed. The difficulties and challenges which stand in the way of management have to be overcome.

Goals

1. To develop the software within the specified budget and within the planned resources. It should be economically justifiable.
2. Time schedule is of utmost importance. The software should be produced on time and is ensured to deliver on time.
3. To develop the software according to the parameters and requirements of the user.

Principles

The following are the principles that are to be followed:

1. Before starting the project, aims and objectives should be clearly understood.
2. The project manager should understand various constraints which may be of hardware, software, cost or any other type of constraints.
3. Making plans in advance to achieve the goals in a better way. Cost estimation, Need for human resources and other factors should be planned well.
4. The project should be monitored so that no problem goes unnoticed.
5. Adjustments, modifications and appropriate changes as and when necessary should be made for appropriate result.
6. A healthy and good work environment should be provided so that each member gives his/her best performance.

Difficulties and Key Challenges

Software project are unmatchable and developed.

1. **Redundancy Problem:** Software projects are unmatchable and developed only once. They face redundancy problem in some cases.
2. **Multiple Solutions:** It becomes very difficult to decide which way to go when there are multiple solutions for a given problem.
3. **Difficult Cost Prediction:** It is difficult to predict personnel cost because they have different capabilities.
4. Technological advancement in hardware and software technologies creates a big obstacle in the development of the software.
5. Software is very difficult to monitor due to its intangibility.
6. Changes in software specifications and design can take place during any phase of software development.
7. **Not Well Designed:** The responsibilities of different team members are not sometime well defined. The user requirements are not precisely defined.

All these difficulties need to be overcome so that the software is developed within the specified time and estimated costs and fulfills the user requirements.

2.4 **SOFTWARE CONFIGURATION MANAGEMENT PROCESS**

Introduction

Software configuration management or in short SCM is the discipline of identifying, defining and controlling changes throughout the life cycle. It includes revision control and the establishment of baselines. Sometimes it so happens that changes arise due to near business or market conditions and the atmosphere so created requires a change in the final product. New user requirements also need modifications of data or functions produced by the software. Other reasons may be business growth, downsizing, cost and time constraints. SCM manages changes in the intermediate products in a controlled manner so as to maintain the integrity of the product. The traceability is also maintained and the development of the product is made more manageable.

Software design may change at different points of time like after testing phase or coding phase. The main points or goals of SCM are as follows:

1. **Configuration Identification:** The structure of the product is identified and then its components and their types are also identified. CCM then makes these products unique and accessible in some form.
2. **Control:** Implementing a controlled change process. For this a change control board is set up which approves or rejects all change requests. It answers consistent software by the creation of a baseline product.
3. **Configuration Status Accounting**: Recording and reporting the status of the components and change requests. It collects all information and vital statistics about the status of the development process.
4. **Auditing and Reviewing:** Validating that the configurations contain all the intended parts are sound. It keeps consistency with respect to specifying documents and ensuring that the product is a well-defined collection of requirements, architectural specifications and user manuals.
5. Build management, process management and environment management all ensure adherence to the organization’s development process.
6. Facilitating team work and making sure that every defect has been tracked back to the source.

Software Configuration Items

It is one of the smallest unit of change. EEE defines it as “an aggregation of hardware, software or both that is designated for configuration management and treated as a single entity in the configuration management process”.

Base lines

Whenever there is a need for a change, a basis for the change must be clearly known to making changes. A baseline helps to control change. IEEE defines a baseline as a specification or product that has been formally renewed and agreed upon that thereafter serves as the basis for further development and that can be changed only through formal change control procedure."

Activities of SCM

The various activities involved in configuration management are:

- **Configuration identification**: The structure of the product is identified and then its components and their types are also identified. CCM then makes these products unique and accessible in some form.
- **Change control**: Implementing a controlled change process. For this a change control board is set up which approves or rejects all change requests. It answers consistent software by the creation of a baseline product.
- **Configuration audit:** Validating that the configurations contain all the intended parts are sound. It keeps consistency with respect to specifying documents and ensuring that the product is a well -defined collection of requirements, architectural specifications and user manuals.
- **Configuration status reporting:** All these activities ensure that all the changes are properly implemented.

Changes in Software Development

— Diagrammatic Illustration.

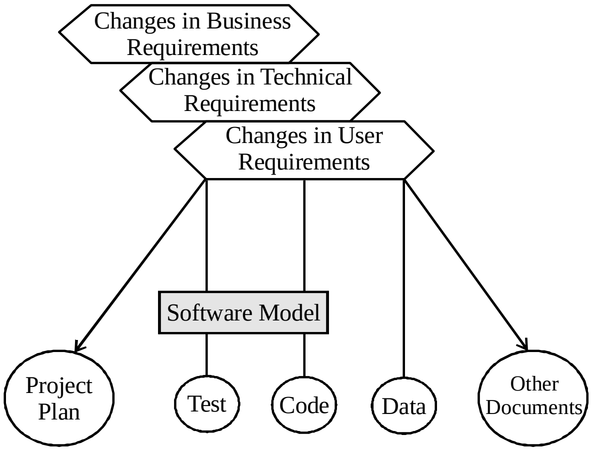

Changes in Business Requirements
Changes in Technical Requirements
Changes in User Requirements
Software Model
Project Plan
Test
Code
Data
Other Documents

Fig. 2.4: Diagrammatic Illustration of Changes in Software Development

M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 3: PROJECT PLANNING

3.1 **SOFTWARE PROJECT PLANNING**

Project Planning is the most important aspect of software project management. Project planning prepares a list of activities that need to be done to achieve the goal. The main purpose of project planning is to clarify the goals, needs and constraints. The success of a project depends upon its planning because poor planning is a sure ticket to failure and poor quality result with high maintenance cost. A good planning prepares a road map in advance what to do, how to do, when to do, who is to do and how much will it cost to complete the project.

3.1.1 **Main Objectives**

Project Planning is a well-organized and an integrated process which keeps in mind the ultimate goal to achieve:

- Project Planning aims at better utilization of resources.
- To finish the project in time.
- Defines roles of different team members.
- Assigns duties and responsibilities.
- It determines project constraints.

A large number of people are involved in the project planning. These include

(i) Senior Management which is responsible for employing personnel's and providing resources.

(ii) The project management team which includes project managers and developers. They are responsible for planning determining and tracking activities, selecting all ways and means to achieve the goal. The project manager is the key to successful project planning.

3.1.2 Principles of Project Planning

The following principles or fundamentals of software project planning are to be followed:

- **Planning is the basic need:** The software project management process begins with a group of activities called Project Planning. This is one of the most important management activity. It is necessary for modern business because helps in better co-ordination and reduces/removes uncertain. Planning is a never-ending process.
- **Risk Analysis:** It is the duty of the Senior Management and Project Management to study all the risks and hindrances that may team affect the final output.
- **Tracking of Project Plan:** The project plan should be continuously tracked and modified as per the requirements.
- **Quality Products:**Those process should be identified which can ensure quality in software.
- **Description of Flexibility:** Planning should be flexible so as to accommodate new changes the project is in progress.

3.1.3 Components of Project Planning

Different types of projects can have different components depending upon their working and implementation. Here are some common and powerful components of project plan.

1. **Project Scope:** Each project has its own wide project scope.
2. **Project Schedule:** Every project has its own time frame within which it must be completed.
3. **Project team organization:** Each project has a well-organized team under a project leader.
4. **Technical Description:** Each project has a well-organized team under a project leader.
5. **Project Standards, Procedures and proposed technical tools**: For success, completion of the project, it must have all the standards, process and technical tools.
6. **Quality Assurance Plan**: Each project has the assured plan for successful completion of the project.
7. Special Development Tools and Techniques are needed.
8. **Configuration Management Plan:** For the long range planned projects.
9. **Documentation Plan:** A project must have the documentation which is well-planned for the maintenance of the project at the later stage.
10. **Data Management Plan:** There should be Data Management Plan for every working project in the short range planning.
11. **Resource Management Plan:** There must be a Resource Management Plan to implement the project in a successful manners.

12. **Test Plan:** A detailed and ordered test plan is needed for the successful testing of two project.
13. **Training Plan:** It must be set for the users for the successful completion of the project and working of the project.
14. **Security Plan:** There should not be any un-authorized access. For this security plans like the secret purpose password should be established.
15. **Risk Management Plan:** A Project Manager must consider all the constraints and possibilities to avoid and type of risk management.
16. **Maintenance Plan:** Maintenance procedures and plans are needed to avoid any type of threat during working hours.

Planning is done on both formal and informal basis. The main aim is to provide the guidelines for decision making are based on future expectations.

3.1.4 **Activities in Project Planning**

Project Planning has various aspects like estimating, scheduling and assigning the project resources in such a way to deliver the find product of desired suitable quality. However activities have an important role which can determine the success of the project. The various planning activities team members will need to do are given as under:

- The recruitment of the team and building of the team as a unit.
- To organize the project
- To know and confirm the start and end dates by preparing a project schedule.
- To create the project budget.
- To know about the customer requirements for the final result.
- Limiting the project scope boundaries and defining them properly.
- To write a description of the find result.
- To assign duties
- To assign accountability.

A typical project plan can include the following activities which have been briefly discussed as below:

1. **Project Scope Definition and Scope Planning:** The assumptions, constraints and user expectations along with business needs, technical needs and project deliverables etc are documented. Project objectives and any other thing that defines the final product requirements is also documented.

2. **Quality Planning:** This is an important part of project planning. Quality of the final product can’t be over looked. For this various factors that influence the quality of the final product are taken its account.

3. **Project Activity:** Definition and activity sequencing: All activities are defined in advance which are required to deliver the product by producing the various product deliverables.

4. **Time, Effort and Estimation of Resources:** Effort, time and resources required are clearly estimated and documented in this step. Many techniques like function points, lives of code, complexity of code, Bench mark etc. can be used to calculate effort.

5. **Risk Factor Identification:** One should be ready to face the unexpected things in life. It is very important to pin point the risk factor based on the assumptions, constraints, user expectations and challenging conditions.

6. **Schedule Development:** Project Scheduling is one of the main task of project planning. Time frame work is to be prepared for each of the inter-dependent activities. It will have an influence on the cost estimates and the cost benefit analysis.

7. **Risk Management Planning:** In this process risk factors are identified and analyzed. A risk resolution plan is prepared to minimize the impact of the risk factor on the project.

8. **Cost Estimation and Budgeting:** The costs involved in the execution and implement of the project is estimated and only then budget allocation is done for different phases of the project.

9. **Organizational & Resource Planning:** The main objective of resource planning is the smooth and efficient running of the project. There are many types of resources like equipment, personnel, facilities, money etc. These resources will have to be fully utilized to achieve the goal.

10. **Project Plan Development & Execution:** The information gathered from all planning processes is used and each of the project tasks and activities are monitored and controlled from time to time delays are analyzed and the project plan is adjusted accordingly.

11. **Performance Reporting:** The progress of each of the tasks is compared with the schedule. Various Techniques such as EVM (Earned Value Management) can be used to measure the performance. Tools like PERT Charts, GANTT Charts, Logical bar charts, Histograms, Pie charts etc. can be used to report the performance.

12. **Planning Change Management:** If the need arises then certain aspects of the project can be changed, but the changes should not have any negative effect on the environmental or the performance. Changes should be studied properly and only then the project plan may be modified.

13. **Project Rollout Planning:** The success of the project depends on the project rollout and implementations. Where a project is rolled out; it has greater impact on the entire technical systems and the business systems. The users should be read to accept it and use it effectively. The users need to be trained to adapt to the new system.

3.2 SOFTWARE PROJECT MANAGEMENT PLANNING

This is an important part of project management. The following points are kept in mind by the project management.

(a) What objectives are to be achieved?

(b) What resources are needed?

(c) How and when the resources are to be acquired?

(d) Ways and means to achieve the goal.

(e) To determine the flow of information, people and products within the system.

Organizing

It involves the division of duties and responsibilities among different groups in such a way that the goals of the enterprises are achieved. This is possible only when proper planning is done by the professional within the groups. Best organization means best results in the end.

Staffing

It deals with hiring of qualified and well trained personnel for the positions that are recognized by the organizational structure. It carries the process of recruiting, hiring, training, rewarding, retaining, compensating, developing and promoting employees.

Directing

It is the process guiding and leaving the subordinates so that they may understand and identify the organizational structure and the goals to be achieved by the enterprise.

Controlling

This step includes the process of measuring and correcting activities. The performance of each group is measured against plans and if there is any deviation, then corrective actions are taken to achieve the desired goal.

Software Project Management is based on sound technical skills and Management Skills. The Technical Process of software development depends on the application of sound management principles. It is not possible for one person to handle todays large and complex programs therefore team work is necessary under the leadership of the project manager.

3.3 SOFTWARE COST ESTIMATION

Cost estimation is the most important task of a project. An accurate and scientific estimation of cost, effective usage of resources and time schedule is the basis for a successful completion of a project. Cost over-run may enrage. The customer and it can lead to the cancellation of the project. There are some factors such as complexity of the project, time availability, reliability and software size which mean a lot of cost estimation. It can be done during any phase of software development. It depends upon software information regarding user requirements, design and source code etc. It becomes easier to estimate cost in the later stages as more information becomes available. To avoid unforeseen delays and risks; cost estimation should be done throughout the life cycle of the project. The following diagram accurately depicts how the cost estimation becomes more and more accurate as additional software information is pouring in.

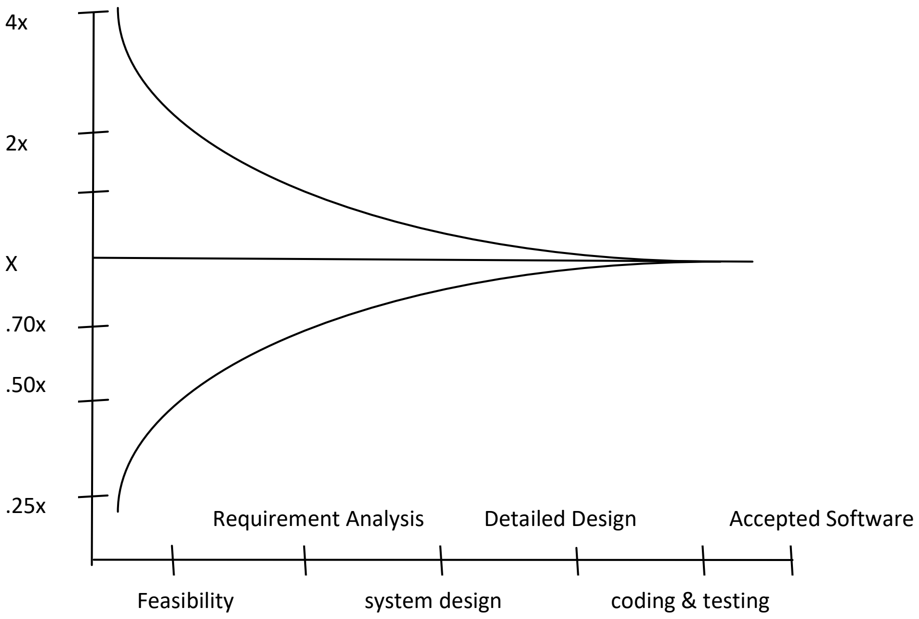


Fig. 3.1: Accuracy of Cost Estimation

The diagram clearly shows that as we proceed from requirement phase to coding and testing phase via system design phase; cost estimates becomes more accurate.

Cost Factors

The following factors influence the cost estimation:

1. **Experience:** If the software developer is well trained in programming language, operating system and hardware the cost will be less.
2. **Product Complexity:** The cost of software project rises up with the level of complexity. Three levels of product complexity are organic, semi-detached and embedded programs.
3. **Project Size:** A large sized project is more costly as compared to a small sized project. More effort is required for a large sized project.
4. **Available Time:** If development time is decreased, then the software project requires more resources and effort and thus the cost increases.
5. **Programmer Ability:** Efficient and capable software programmers bring down the expenditure whereas inefficient programmer increases the cost. Therefore, it is necessary that a programmer should be well-trained.

6. **Level of Technological Know:** How also has a great influence on cost estimation? Modern practices, System analysis, Design techniques etc. have a great effect on cost estimation.

7. **Reliability:** Reliability of the software project depends on the level of accuracy, robustness and consistency etc. cost estimates depend on the level of analysis, design, implementation and verification effort.

The cost estimation should be well-planned. It should be reviewed at regular intervals. The software process should be continually updated and improved upon.

3.4 SOFTWARE SIZING

Sometimes a problem is too big to the understood in a single attempt, similarly software cost estimation is not simple as it appears to be a complex problem like software cost estimation is broken up into fragments to achieve an accurate cost estimate.

Mainly two approaches are adopted:

(i) Problem based estimation
(ii) Process based estimation

A project planner must establish the estimate of software size which is considered to be the Quantitative result of the software project. The software is divided into smaller parts because it is convenient to calculate and know about the size of the smaller components. They are less complex. The addition of these smaller components gives an idea of the overall estimate of software size.

Sizing Approaches: There are two types of approaches for estimating size:

1. **The Direct Approach:** Measures sizes in terms of (LOC) lines of code.
2. **The Indirect Approach:** Measures in terms of functional point (FP).

Parameters: The following parameters influence the accuracy of size estimates:

1. The degree of properly estimating the size of the software.
2. The capability of converting Size Estimate to human effort, calendar time and money.
3. The ability of the software team.
4. The stability of product Requirements and the overall environment supporting the development process.

3.4.1 Approaches/Types of Sizing

The project planner finds that, the estimation of size is very important in determining the cost estimation. This problem is also known as Software Sizing. To solve this problem, the following methods can be used:

1. **Fuzzy Logic Sizing:** The planner identifies the application type and its magnitude on a quantitative scale. The magnitude is then refined within the original range.

2. **Function Point Sizing:** In this method the functionality produced by the software system is measured the function point estimates.

3. **Standard Component Sizing:** Standard components can be in the form of modules, screens, reports, lines of code and soon. The number of time a component occurs is estimated and the historical data helps to find out the delivered size of each standard component.

4. **Change Sizing:** Through this method, already existing project is modified and the number and type of the modifications are estimated.

3.4.1.1 Lines of Code

It can be defined as the number of delivered lines of code. Lines of code are a measure to calculate productivity metrics. LOC is used as follows:

(i) As an estimate variable to size each and every element of the software.
(ii) It is used as baseline metric obtained from the previous projects and used with estimation variables to develop projections for cost and effort.

Line of code is the most commonly used software size metrics. LOC depends on the programming language. In assembly language lines of code will be higher as compared to lines of code in high level language like C++/Java.

In the earlier stages of development, less information becomes available where as full information becomes available only at the end of the project. So exact number of lines of code can be determined after the completion of the project.

Guidelines

1. One line of code is for One Logical line of code.
2. Lines of code delivered as a part of software are included.
3. Lines of code for drivers, test stubs and other support software are excluded.
4. Software code by software developer is included whereas the code created by the application generators are excluded.
5. Declarations in the programme are counted as LOC whereas comments in the programme are not counted as LOC.

Keeping in view the historical Data, The project planner estimates. Three values optimistic S_{opt}, most likely (S_m) and pessimic (S_{pess}) for each size.

The equation ($S_{opt} + 4 \times S_m + S_{pess}$) is used to compute expected value. The following example of a software will clarify how the estimating size is determines. Here in the example software is a combination of six functions — (i) user face, (ii) word processing, (iii) file storage and retrievals, (iv) Data base management, (v) Word processor and (vi) Peripheral control.

Estimating Size

Function	$\mathbf{S_{pess}}$	$\mathbf{S_m}$	$\mathbf{S_{opt}}$	**Expected Size** $\mathbf{S = \frac{1}{6}}$ $(S_{opt} + 4 \times S_m + S_{pess})$
User interface	1500	1800	2100	1800
Word Processing	1800	2400	3000	2400
File Storage & Retrievals	1700	2100	2400	2083
Data base management	2500	3000	4000	3083
Word Processor	1200	1600	2000	1600
Peripheral Control	1300	1700	2000	1683
Total Expected Lines of Code =				**12649**

Table showing method of calculating Size Estimation

The developer estimates the size of each and every function:
S for user interface =
S for word processing = and soon
We see that size of the software in terms of
LOC is = 12649

3.4.1.2 **Function Point**

Function Point lines of code are the most important methods for size estimation. The function point metric was devised and suggested by A.J. Albrecht. This is used to measure functionality delivered by the system; Function point estimates in determining estimate effort required to design, code and test software, predict the number of errors and it so forecasts the number of components being used in the system.

The measures of software information domain value and software complexity are the two things needed to find out function point through an empirical relationship.

Software complexity is of three categories; of simple, average and complex level.

Information domain value is a combination of the following points.

1. **Number of external inputs or (EI):** Users and other applications are a source of providing application oriented data.

2. **Number of external outputs or (EO):** Theserefer to Reports, Screens, Error, messages etc. The user gets information from each external output. Individual data views are not counted separately.

3. **Number of external inquires (EQ):** Online inputs are referred to as external inquiries. These in turn. Generate responses in the form of online outputs. In this case each and every district inquiry matters and is counted separately.

4. **Number of internal logical files/(ILF):** Logical grouping of data present in the application boundary e.g. master file as part of database is known as internal logical file.

5. **Number of external interface files (EIF):** Logical grouping of data which is external to the application e.g. Data files on tape or disk is known as external interface file.

The complexity value for each count is determined after gathering information about information domain value function point method helps in deter mining about a particular entry whether if is simple, average or complex.

Estimating Functional Points: Functional points in software are estimated by the equation as given below.

F.P = Count Total (o.65 + 0.01 × Σ (f1)]

Count Total = Sum of all F.P. entries

fi = (i = 1 to 14) are value adjustment factors.

The value adjustment factors are based on the answers to 14 Questions. These questions are answered using a scale of 0 to 5 where

0- means – No influence
1- means – incidental
2- means – Moderate
3- means – Average
4- Significant
5- Essential

Weighting Factor Table 1

Measurement Parameter	**Count**		**Simple**	**Average**	**Complex**		**F.D. Count**
Number of User inputs	21	×	3	5	7	21×5=	105
Number of user output	15	×	2	4	6	15×4=	60
Number of user inquiries	25	×	4	5	6	25×5	125
Number of files	7	×	8	10	14	7×10	70
Number of external interfaces	3	×	5	7	10	3×7	21
Count Total							**381**

Value Adjustment Factors Table II

Sr. No.	**Factors**	**Disadvantages**
1.	Backup and Recovery	2
2.	Data Communications	2
3.	Distributes Processing	1
4.	Performance Critical	3
5.	Existing operating environment	2
6.	Online data entry	3
7.	Input transactions over multiple screens	4
8.	Master files updated online	3
9.	Information domain value complex	4
10.	Internal processing complex	2
11.	Code design for reuse	3
12.	Conversions/installation in design	3
13.	Multiple installations	4
14.	Application design for change	4
	Value adjustment factor	40

Computations for Function point: To estimate size in terms of function point table I and Table II are used as follows

FP count = Count X weighting factor (AV)

From Table I FP for number of used inputs = 22×4=88

using Table I and II F.P. count Total × [0.65+0.01×?fi]

= 381× [0.65+0.01×40]

= 381× [0.65+0.40]

= 381× 1.05

= 400.05

= 400

3.4.1.3 **"Fuzzy Logic"**

A project estimate is only as good as the estimate of the size of the work to be accomplished. Different approaches to software sizing are given as under:

1. Lines of codes
2. Functions point
3. Fuzzy logic
4. Change sizing

Fuzzy Logic sizing is a type of approach in which the project planner must identify the application type and its magnitude on a quantitative scale. After this the magnitude is refined which in the original range.

"Fuzzy Logic" approach applies the approximate reasoning techniques that are the corner stone of fuzzy logic. The personal experience can find a place but the planner should also keep on eye on the historical database of the projects so that estimates can be compared to original/actual experience.

3.4.1.4 **Change Sizing**

Sizing signifies the project planner's first major challenge. Sizing steps are essential to make valid comparison cross systems or within systems. Productivity can be computed with the help of a software sizing measure.

The change sizing approach is followed only when there arises a need for changes in the already existing project specifications. This modification is used in the new project to bring a required change in its accomplishment. The planned estimates, the number and type (e.g. reuse, adding code, changing code, deleting code) of modifications that must be finalised. By the application of an "Effort Ratio" for each type of change, the size of the change may be guessed and this new estimate will help to produce the software according to the user requirement.

3.5 **PROBLEM BASED ESTIMATION**

LOC and F.P. data are used in two ways:

- As an estimation variable to 'size' each element
- As baseline metrics gathered from past projects to develop cost and effort projections.

Although LOC and F.P are clearly different techniques but they have many common characteristics. After a bounded statement about software scope, the project planner decomposes the software into problem functions so that they may be estimated individually. LOC or FP is then estimated for each function.

Using baseline productivity metrics, proper estimation variable and cost or effort for the function is calculated.

LOC and FP estimation are different so far as their technique of decomposition and the target of the partitioning is concerned when LOC is used the decomposition approach supposes that all functions can be broken up into sub functions. LOC estimates depend on the degree of partitioning. For FP estimates decomposition works in a different manner. Instead of focusing on function, it estimates on the basis of information domain characteristics i.e. inputs, outputs, data files, inquiries and external inferences. In addition to 14 complexity adjustment values are estimated and a FP value is derived that can be linked to past data and this is used to generate an estimate.

3.6 **PROCESS BASED ESTIMATION**

In order to achieve objectives in software project developed a specified and certain process is followed. The estimation of effort in a software is to base the estimate on the process to be followed. The whole process is divided into smaller tasks and after the identification of these tasks; the effort for each task is estimated. These smaller tasks are such as analysis, design and coding etc. These tasks are ordered according to the specified sequence. The project managers take the product size as a major productivity indicator and they become capable of tracking a project. A number of activities are performed in process based estimation. These activities can be listed as follows:

1. Customer communication
2. Planning
3. Risk Analysis
4. Engineering (analysis/Design)
5. Construction release and so on

It is the duty of the project planner to combine the functions and activities to calculate the effort required for each function. The labour rates which vary from task to task are applied. Top level management activities are more expensive on compared to activities performed by the junior staff in the beginning of the framework activities. The accuracy in process based estimation depends on many parameters.

- The degree of sizing of the software and the proper estimation.
- Changing size estimate into human effort, calendar time and money.
- Ability of a software team
- The stability of product requirements
- Environment supporting the development process.

The software considered for process based estimation is divided into functions and for each function a set of tasks and activities are taken last of all, the effort for each function and activity is calculated.

Activity	Customer Communication	Planning	Risks Analysis	Engineering		Construction Release		CE	Total
Task				Analysis	Design	Code	Test		
Function 1				0.50	2.00	0.40	5.00	n/a	7.90
Function 2				0.70	4.00	0.60	2.50	n/a	7.80
Function 3				0.50	3.00	1.00	3.50	n/a	8.00
Function 4				0.50	3.00	0.75	2.50	n/a	6.75
Function 5				0.50	2.00	1.00	2.50	n/a	6.00
Function 6				0.25	1.00	0.50	2.50	n/a	4.25
Function 7				0.50	1.00	0.50	1.00	n/a	3.00
Total	0.25	0.25	0.25	3.45	16.00	4.75	19.50	-	43.70
%Effort	1%	1%	1%	7.66% = 8%	35%	11%	43%	-	

Table showing percentage effort for each function & Activity

3.7 **COCOMO-MODEL (CONSTRUCTIVE COST ESTIMATION MODEL)**

Barry Boehm gave birth to the idea of COCOMO Model in early 19805 to estimate total effort required to develop a software project. This Model is widely used and if is based on the observation of previously developed software projects. According to Boehm any software development project can be categorized into three type's organic, semidetached and embedded depending on the development complexity.

Organic projects: These projects are in an area in which the organization has enough experience. These project are small in size (not more, the 50 KOLOC). Such system has small teams with prior experience. The term members work together with through understanding and co-operate to accomplish user requirements which are less stringent examples: Simple business system, simple inventory management system etc.

Semi-detached: In such a project, the development team consists of a mixture of experienced and inexperienced staff. The size of such projects is not more than 300 KDLOC

Examples: Operating system, compiler design and data base design

Embedded Projects: These projects are ambitions and novel, they have size more than 300 KDLOC. The organization has little experience developers have to achieve stringent user requirements and these systems have tight constraints from the environment (hardware, software and people)

Example: Software systems used in avionics and military hardware, real-time command system.

COCOMO model estimates the total effort in terms of person-months of the technical project staff. the efforts estimate includes development, management and support tasks but does not include the cost of the secretarial and other staff. Our person month means the effort an individual can typically put in a month. The basic steps in this model are:

1. Obtain an initial estimate of the development effort from the estimate of thousands of delivered lines of source code. (KOLOC)
2. Determine a set of 15 multiplying factors from the different characteristics of the project.
3. Multiply the initial estimate (E_i) with all the multiplying factors.

The initial estimate E_i in person\-months is determined by the equation E_i = a* (KDLOC) b. The value of the constants a and b depend on the type of the project. KDLOC is used as the measure of the size. COCOMO is based on the hierarchy of three models: basic model, intermediate model and advance model.

Basis Model: In this model only the size of the project is taken in given an approximate of the project parameters.

The Expression for calculating effort is as follows

Effort = a * $(KLOC)^b$ PM

Basic Model			
Project Type	Constant (a)	Constant 'b'	E
Organic	2.4	1.05	2.4× (30) 1.05 for 30 KDLOC = 85 pm
Semi-detached	3.0	1.12	3.0× (30)1.12 for 30 KDLOC
Embedded	3.6	1.20	$3.6\times(30)^{1.20}$ for 30 KDLOC

Effort Calculation

By using logarithmic tables; effort can be found out. Estimation of development time (Tdev): The formulator estimating the development time based on the effort are as follows:

Organic : T_{dev} = 2.5 (effort) 0.38 months
Semidetached: $T_{dev} = 2.5\ (\text{effort})^{0.35}$ months
Embedded : $T_{dev} = 2.5\ (\text{effort})^{0.32}$ months

e.g. if an organic type software product is estimated to 32000 lines of source code and a software engineer gets Rs. 6000 as salary par month; then nominal effort, nominal development time and cost required to develop the project an be calculated as

Effort : $2.4\ (3.2)^{1.05} = 91$ PM

T_{dev} : $2.5\ (91)^{0.38} = 14$ months

Cost: 14× 16000 = 224000

Intermediate COCOMO Model

This model takes other relevant parameters into account besides development time and effort the intermediate COCOMO model refines the initial estimate (E_i) by using a set of 15 multipliers based on different characteristics of the software development to calculate total effort in this model, the procedure is as follows:

1. Calculate (E_i) considering the size in terms of KDLOC.
2. Identify 15 parameters and each parameter is given a numerical value called multiplying factor.
3. Effort adjustment factor (EAF) is derived by multiplying all multiplying factors with each other.
4. Multiply initial estimate in step 1 with EAF.

15 Different attributes called cost driver attributes determine the multiplying factors which depend on product, computer, personnel and project attributes are shown in the table below:

Cost Drivers		Rating				
		Very Low	Low	Normal	High	Very High
	Product Attributes:					
1	RELY = Required Reliability	0.75	0.88	1.00	1.11	1.40
2	DATA = Database size	0.94	1.00	1.08	1.16	-
3	CPLX = Product Complexity	0.70	0.85	1.00	1.15	1.30
	Computer Attributes:					
4	TIME = Execution Time Constraint	-	-	1.00	1.11	1.30
5	STOR = Main Storage Constraint	-	-	1.00	1.06	1.21
6	VIRT = Virtual Machine Volatility	-	0.87	1.00	1.15	1.30
7	TURN = Computer Turnaround Time	-	0.87	1.00	1.07	1.15
	Personnel Attributes:					
8	ACAP: Analyst Capability	1.46	1.19	1.00	0.86	0.71
9	AEXP: Applications Experience	1.29	1.13	1.00	0.91	0.82
10	PCAP: Programmer Capability	1.42	1.17	1.00	0.86	0.70
11	VEXP: Virtual Machine Experience	1.21	1.10	1.00	0.90	-
12	LEXP: Language Experience	1.14	1.07	1.00	0.95	-
	Project Attributes:					
13	MODP: Modern Programming Practices	1.24	1.10	1.00	0.91	0.82
14	TOOL: Software Tools	1.24	1.10	1.00	0.91	0.83
15	SCHED: Development Schedule	1.23	1.08	1.00	1.04	1.10

Determination of multiplying factor for different Attribute

Advance Model: In this model, effort is calculated as a function of programme size and a number of cost drivers for each phase. This model covers the limitations of both basic and the intermediate model. Both there models consider a software product as a single homogeneous entity. But most large systems are made up of several smaller sub systems. These sub systems have widely different attributes. Some may be organic type, some semidetached type of some embedded. Complexity of subsystems may be different.

There are four phases in the advance COCOMO model. They are (1) Requirements Planning and Product Design (RPD) (2) detailed design (DD), (3) Code and Units Test (CUT) (4) Integration and Test (IT)

Each cost driver is rated as very low, low, nominal, high and very high. Multiplying factors are assigned for Analyst capability cost driver (ACAP) for each phase of advanced model are tabulated below:

Rating	RPD	DD	CUT	IT
Very Low	1.80	1.35	1.35	1.50
Low	0.85	0.85	0.85	1.20
Nominal	1.00	1.00	1.00	1.00
High	0.75	0.90	0.90	0.85
Very High	0.55	0.75	0.75	0.70

Rating of Cost drivers

3.8 **THE SOFTWARE EQUATION**

The software equation is a dynamic multi variable model that assumes distribution of efforts over the useful life of the project. The software equation can be derived from data obtained by studying several existing projects and it been done by collecting over 400 contemporary software projects. For the calculation of effort, the following equation has been framed:

$$E = \left[\underset{\substack{\Downarrow \\ \text{LOC}}}{\text{Size}} \times B^{0.333} / P\right]^3 \times \left(1/t^4\right) \text{.......equation}$$

Where

P = Productivity parameter which indicates the maturing of overall process and management practices. This parameter also indicates:

(a) The level of programming languages used.
(b) The state of software environment.
(c) The skill and experience of the software team.
(d) Complexity of software application.

T = it indicates project duration in months or years

E = Efforts in person-months or person-years.

B = Special skills factor — the value of B increases over a period of time depending upon the importance and need for integration, testing, quality assurance, documentation and management.

Software equation has to important independent indicators they are:

I. An estimate of size (in LOC)
II. Time duration in months or years.

Potnam and myers suggested another set of equation derived from the software equation for simplification of estimation process.

t_{min} = 8.14 [LOC/P]$^{0.43}$ in months for t_{min}> 6 months equation-II (a)

E = 180 Bt^3 in person – months for E $\geq$ 20 person months eqn-II (b)

____For small programs with sizes between 5KD LOC and 15KD LOC, the value of B is 0.16 and for programme with sizes greater than 70KD LOC; the value of B is 0.39

3.9 PROJECT SCHEDULEING

Definition

Project Scheduling can be defined as the process of effectively managing the tasks of the project by identifying and organizing the tasks into a series of events. Project scheduling prepares details such as (i) start dated of the project, (ii) end date of the project, (iii) milestones, (iv) specifying the resources required and (v) tasks for the project. In ensures a harmonious completion of the project on time. It also helps to avoid the extra cost incurred when the project gets delayed. Project scheduling tries to predict the future. Although it is not possible to say with certainly about the length of time yet there are methods which can increase likelihood of being close.

Principles of Project Scheduling

The most commonly adopted principles for project scheduling are as follows:

(i) **Compartmentalization:** The project is divided into several tasks to manage the project effectively.

(ii) **Interdependency:** All the activities of the project are inter-related and not independent of each other. Various activities are performed in a sequential manner. Some activities depend so much on other activities that they cannot begin until the activity on which they depend is completed.

(iii) **Time allocation:** Another principle of scheduling is time allocation. It determines the time for performing specified activities. It is also very important to estimate the effort required. The project management should assign final time to each team member.

(iv) **Effort Validation:** The project manager assigns tasks to the team members according to their capabilities and efforts required for the task.

(v) **Defined Duties and responsibilities:** The roles and responsibilities of each project team member should be decided in advance depending on their skills and capabilities.

(vi) **Defined Outcomes:** According to this principle, the outcomes of every task should be specified. A product is outcome of a task and these products are well combined in deliverables.

(vii) **Defined Milestones:** Mile stones be specified when task products are complete and it should be renewed for quality.

Delaying Factors

There are various factors which are responsible for delay in the project schedule. The most common factors are:

- **Unrealistic Deadlines:** Sometimes the allocation of time for completing a project is not practical because inexperienced individuals establish a wrong deadline. The project gets delayed if deadline is not achieved.
- **Change in User Requirements:** There is delay if user requirements are changed after the project has already started. This disturbs the project schedule and thus more time is taken.
- **Under Estimation of Resources:** If there is under-estimation of resources, it leads to delay in the completion of tasks of the projects.
- **Lack of Analyzing Risk Factors:** If risk factors are ignored during project planning and scheduling; these risk factors put on a bad effect on the software development.
- **No Communication among Team Members:** Lack of communication among team members makes it difficult for the project management to complete the project on time. Personal problems of team members also stand in the way of the completion of the project.
- **Lack of Action on the part of:** If the management fails to take timely action and does not understand that the project takes more time to complete; then it also becomes the reason of delay.

Activities in Project Scheduling

Various activities are involved in project scheduling. There are to be observed sincerely. The experience in the project area and in the scheduling. Those persons are interviewed who had experience with similar projects. Transition between activities is taken into account uncertain resources of talent, equipment or data are also studied so involved in project scheduling is as follow:

1. Gant Technique
2. Mile stone Scheduling
3. Pert and CPM

The Gant technique is used where production operations are highly repetitive and work performance of different departments can be combined.

In the mile stone scheduling system the milestones are developed and established in the planning phase.

PERT/Networking and CPM method are designed to be used in the development phases for identifying the critical path, float and slack.

PERT Chart (Program Evaluation and Review Technique Chart)

A PERT chart is a graphical chart meant to determine the activities that from the “Critical Path” which if delayed will harm the overall project and the project will be delayed. The use of PERT chart is justifies in large projects in which activities can be divided into two categories (i) Critical activities, (ii) Non-critical activities. The critical activities should not be delayed.

A PERT chart is a method of project management which is used to schedule, organize, and coordinate different tasks within a project. If analyses the involved tasks and determines the time needed to complete each task and thereby identifies the minimum time needed to complete the total project in time.

The PERT chart methodology was first used by U.S. Navy in 1950s to manage the Polaris Missile Project.

Advantages of PERT Chart

1. It illustrates the project in a graphic form.
2. It guesses expected time required to complete a task.
3. It provides information regarding the time expected to complete the project.
4. It describes if there is the probability of completion of the project before the date specified for it.
5. It specifies the activities forming the critical path.
6. It provides information about start and end dates of different activities.
7. It describes the inter-dependence of one or more tasks on each other.

Steps for Creating a PERT Chart

1. Identification of different activities and milestones.
2. Recognizing sequence of Activities: Different activities are inter dependent and inter-related. It is decided whether the activities are serial wise or concurrent and then relationship among activities is depicts.
3. Preparation of PERT chart.
4. Estimation of Time: An activity can be shorter or longer depending upon the time required completing it. The time estimates can be months, weeks or days hence three categories arise.

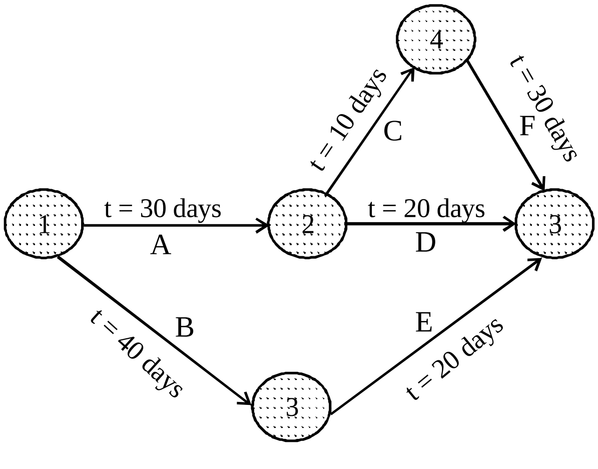
Optimistic Time: It is the shortest time needed by an activity to be completed.

Most Likely Time: It has the highest probability for the completion time required.

Pessimistic time: It is the longest time taken by an activity.

5. Critical Path Determination: Critical path for each activity is specified which determines the time required as per the project schedule.
6. Updating of PERT Chart: If any need arises; PERT chart is modified to avoid delay or when additional resources are to be provided.

Illusion of PERT Chart



PERT Chart

Fig. 3.2: PERT Chart

Milestones

In figure 1, 2, 3, 4, 5 represent the mile stones (represented by circles). The can also be represented by rectangles. When a mile stone is completed, It gets one number higher than the previous milestone. Each milestone is linked to other milestone by one or more arrows.

Activities

English letters A, B, C, D, E, F represent different activities. The direction of arrows shows the sequence of activities. There are two types of activities i.e. Serial activities and concurrent or parallel activities.

(a) (i) Activities A, C,F are serial activities taken in the same sequence. Similarly Activities B and E are also in the same sequence.

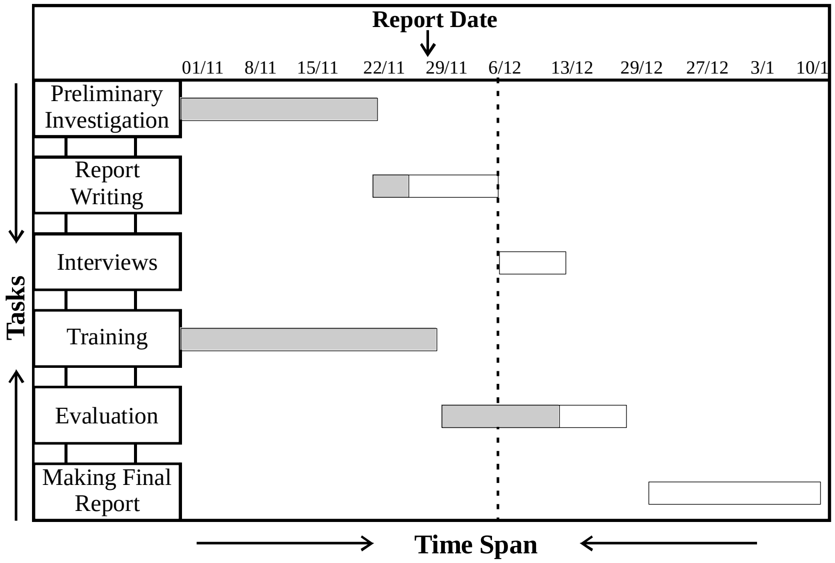
(ii) Activities A and B are concurrent activities. Similarly C and D activities are concurrent activities as these are performed simultaneously.

(b) Time for each activity is depicted by 'L' in days. It can be depicted in weeks or months also.

Gantt Chart

There are various techniques of project scheduling, task method and tracking the schedule which keep an eye on the activities that are completed as per the project schedule. These techniques provide information about activities in a graphical form – through bar charts or histograms etc. It becomes easy for the management activity. The commonly followed techniques include Gantt Chart, PERT Chart etc.

Gantt chart is in the form of a graphic diagram. This chart prepares a picture of the activities of the project. The horizontal bars in the diagram show the total time span required. This time span is divided into months, weeks and days. The time taken to complete an activity is shown in parts. The vertical column shows the activities involved. The graphical chart also shows the start and end dates of each activity. This is the reason that Gantt chart is also known as timeline chart.



Gantt Chart

Fig. 3.3: Gantt Chart

A Gantt chart is commonly prepared on a graph paper. If the Gantt chart is big and complex then other applications like Microsoft fixed etc. A project manager comes to know about the status of the activities. Gantt chart has many advantages as given below:

1. Representation of the project in a graphical form

2. It shows the progress of each activity.
3. It provides a record of the different tasks facilities.
4. It depicts milestones after each activity is completed.
5. It clarifies the tasks which are assigned to project management team members.

There can be a change in the schedule of two projects. Hence Gantt chart also varies accordingly. The diagrammatic representation given above shows the schedule if a project:

1. The horizontal bars show the total time span in the form of months, weeks and days.
2. The time for each activity is shown in increments.
3. The vertical axis shows different tasks like preliminary members, Training and so on.
4. Shaded parts show the part of the activities that has been completed.
5. Unshaded parts show uncompleted part of the activity.
6. Horizontal bars have different lengths showing the time required to complete a specific activity.
7. Time span of one or more activities can overlap each other.
8. With the progress in the project the unshaded bars are shaded showing the completion of activities.
9. The vertical line represents the report date on which an arrow is drawn.

3.10 **BASICS OF SOFTWARE MEASUREMENTS**

Measurements in everyday life has a significant role to play. Without accurate measurements; we shall not be able to take right and timely decisions in area of production and manufacturing. Software engineering uses the concept of measurement in a big way to estimate costs, to monitor inventories and asses quality. This is desirable to understand and improve the software process by predicting, planning and controlling the software projects measurements are in no way an conditional or non-value added task but it has evolved into a very significant discipline in software engineering. This is the reason that most of the organizations are using measurements to ensure better quality products. Software engineers use measurements to gain insight into the design and development of the work products.

Measure, Measurements, Metrics and Indicators are the terms which are used in measuring various aspects of software. Although they have resemblances but they are different.

Measure: It means to mention some aspect of the object in a quantitative way. It can be defined as the quantitative indication of amount, dimension, capacity or size of the products and process and their attributes. It is established. When a number of errors are detected in a single review.

Measurement: The method of determining a measure a known as measurement. It is the process of assigning number or symbols are assigned to different attributes. It is established when different components are reviewed and each unit is tested to collect the measure of a number of errors in all these components. Clearly defined rules are followed.

Metrics: A metric is used to compare two or more measures. It is a quantitative measure of the particular characteristic of a program’s efficiency. It also assesses the level of complexity, strength of the module and estimates time and cost of the project.

Indicator: This term is used to denote a representation of metric or a number of series of metrics that provides information about the software development project in process. The indicators enable the project manager to adjust the process, the project or the product for better results. It draws a person’s attention to a particular problem, tracks potential risks and evaluates the project team’s ability to control the quality of the products.

3.11 **TYPES OF SOFTWARE MEASUREMENT**

Some sort of measures are required to assess the quality of the final product in a software design and to better understand the models produced. The main aim is to improve the software process on a regular basis. This helps in estimation, quality control project control throughout the life cycle of a software

Project Decision making and further planning becomes easy and fruitful. There are two types of measurements:

1. Direct Measurement
2. Indirect Measurement

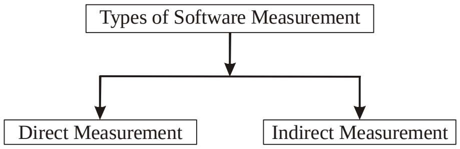


Fig. 3.4: Types of Software Measurement

3. **Direct Measurement:**When the attribute of an entity is measured directly without taking consideration of other attributes, we name it as direct measurement. Direct measures include cost and effort applied and products like lines of code, execution speed, memory size and other defects that have been found.
4. **Indirect Measurement:**There are some attributes like functionality, quality, complexity, reliability, maintainability efficiency can't be measured directly. They are calculated on the basis of the study of other attributes.

The measurements once taken help the project manager and the entire software team to take right decisions that will lead to successful completion of the project.

3.12 **CRITERIA OF MEASUREMENT**

The measurement program is considered successful if it follows the following key characteristics. diagram

1. **Objective and Repeatable:** The measurement program should be very easy to calculate and understand the same measurement tool should be possible to repeat again and again on similar set of data, without any problem.
2. **Timely:** The measurement program must be should be made available at the correct time to put on influence on the quality development and maintenance of the software project.
3. **Iterative:**The measurement process should be iterative which means that the project manager and team members should continually focus measurement efforts on the most critical areas. It should continually work unstopped during the project life cycle to provide necessary information and knowledge to the project managers.
4. **Related to information needs:** The whole process which includes collection, analysis and reporting of data must be related directly to the information needs of the decision taking persons. The information needs should be filled on a priority basis and addressed accordingly.
5. **Interpretation:** Measurement should enable the project manager and team members to gain insight into the design and development of the work products.
6. **FeedBack:** There must be some follow up and measurement communicator's recommendations derived from product metrics to the software team. This develops appropriate metrics for software under consideration.

3.13 **PROCESS OF MEASUREMENT**

A software measurement process is a management tool which if conducted effectively helps the project manager and the entire software team to take steps that lead to successful accomplishment of the project. This process is all about defining what information is needed by the decision makers. Five sets of activities are done:

1. **Formulation:** To measure and develop appropriate metrics.
2. **Collection:** To collect data to derive the formulated metrics.
3. **Analysis:** This calculates and analysis metrics and the mathematical tools are used.
4. **Interpretation:** To attain insight into the quality of the product.

5. **Feedback:** The suggestions / recommendations are forwarded to the software team.

The important stages in this process are:

(i) **Plan Measurement:** In this stage, information needs are identified and the most appropriate measure is chosen to address those needs. Only important measurements are collected which are relevant to the information needs; unnecessary ones are ignored. This stage also determines the resources and the technologies required to implement a software measurement program. An organization is interested in assigning defects, errors, cost and schedules linked to the development process.

(ii) **Perform Measurement:** In this step procedure for defined measure is documented to collect each measure. in other words we can say that this stage revolves around the collection and processing of measurement data. The procedures should be clear and repeatable. A mechanism is needed to verify the correctness and effectiveness of measurement.

(iii) **Evaluate Measurement:** This stage is concerned with the study analysis and evaluation of the main events of the process to ensure that the measurement approach and alerts about extra cost and warns about schedule over runs.

(iv) **Provide feed Back:** Preparation of Reports, Review exercises and project meetings etc. Provide valuable feedback to implement efficient measurement program. This activity ensures that measurement is amply supported at the project level and organization level.

The decision makes and concerned team members operate within these stages and in the end use the measurement information to develop the software as per requirement specifications.

3.14 **BASIC CONCEPTS OF SOFTWARE METRICS**

Metrics is defined as the quantitative measures that allow software engineers to identify the effectiveness and improve the quality of software process, project and product IEEE defines as metric as "a quantitative measure of the degree to which a system, component or process possesses a given attribute" soft metrics serves the main purpose of identifying and controlling essential parameters that has an effect on the software development. It is also helpful in fulfilling the following objectives:

(i) Quantitative measurement of the size of the software.
(ii) Assessment of the level of complexity.
(iii) Assessment of the strength of the module.
(iv) Assessment of the testing techniques.
(v) It specifies when to stop testing and determines the date of release of the software.
(vi) It estimates cost of resources and project schedule.

Software metrics are needed to quantity the development process and maintenance of software. We get information about the status of an attribute and helps evaluate if an objective way. It becomes easy to make modification plans. To achieve the desired goal, software metrics are used in different projects for a long period of time to obtain indicators software metrics help in project planning and project. Management activity and thus project younger and project team members become capable of taking right path to complete the project.

3.15 **IMPORTANCE OF SOFTWARE METRICS**

Software Metrics has achieved a very crucial role in finding answers to questions within the discipline of software engineering software metrics is necessary to decide the following things :

(i) Total time required to complete the project.
(ii) Total cost required to complete the project.
(iii) Number of persons and other resources required.
(iv) Approximate maintenance cost required.
(v) What is needed to test better quality and find the ways for better quality.
(vi) Number of errors to be discovered before delivering the product.
(vii) Estimation of effort to make modifications.

Software metrics help in minimising the software problems and thereby making if possible to efficiently measure and evaluate the attributes of the software. Thus we see that software metrics are helpful in giving the right information about the software development process and in this way help in determining the areas of improvement and build a successful system.

3.16 **CATEGORIES/TYPES OF SOFTWARE METRICS**

An effective software metrics helps a software engineer to develop software as per user requirements within estimated schedule and cost estimate etc. For this different type of metrics for measurement are there. They have been categorized on follows:

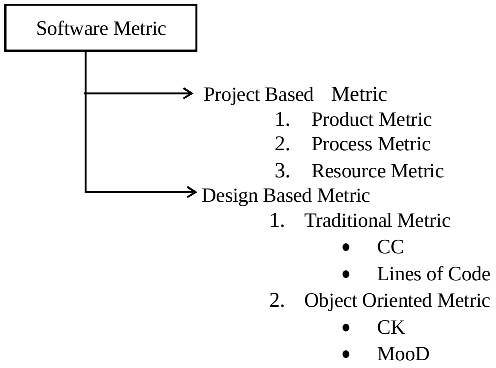


Fig. 3.5: Software Metric

3.16.1 **Projects Based Metrics**

Project metrics illustrate the project characteristics and their execution. Project metrics help the project manager in assessing the current projects. He can identify potential risks and problem areas. After that he adjusts work flow and evaluates the project team’s ability to control the quality of work products Projects metrics are used for tactical purposes and not for strategic purposes.

Project metrics help to decrease the potential risks and problems. By making necessary adjustments; project metrics help to minimize the development schedule. This avoids delay, the other purpose of project metrics is to assess the product quality and to modify the technical issues as and when required. By reducing the number of errors and defects, the quality of the project improves and this leads to decrease in the overall cost of a software project.

In project metrics; often the first application takes place in estimation process. The study and the metrics of the previous projects help in estimating time and effort for the current project. As the project moves ahead, original estimates of time and effort are compared with the new measures of effort and time. This helps the project manager to supervise and control the project.

Project metrics are used to track the errors detected during each development phase. For example of software progresses from design to coding, project metrics are used to assess quality of the design and obtain indicators which will affect the approach chosen for coding and testing. Project metrics are used to measure rate of production in terms of models developed, function points and delivered source lines of code.

3.16.1.1 **Product Metrics**

At the end of each phase, a working product is developed in the software development process. Each product can be measured at any stage to check whether it is according to the required specifications or not. Metrics are developed for these products. If a product does not meet user requirements; then necessary steps are taken. Product metrics help the software engineer to detect and correct potential errors before they may result into bigger problems. Product metrics assess the internal attributes of the products so that the efficacy of the following may be judged:

(i) Analysis, design and code model.

(ii) Potency of test cases.

(iii) Overall quality of the software under development.

The following different types of metrics have been devised for products in the development process:

(i) **Metrics for Analysis Model:** These metrics study and find answers to various aspects of the analysis model like system functionality, system size etc.

(ii) **Metrics for Design Model:** Software engineers take the help of the metrics to assess quality of design that includes architectural metrics, component level design metrics and so on.

(iii) **Metrics for source code:** These metrics are used to assess source code complexity, maintainability and other characteristics.

(iv) **Metrics for testing:** These help in designing efficient and effective test cases and if also evaluates the effectiveness of testing.

(v) **Metrics for maintenance:** To assess the stability of the software product.

3.16.1.2 **Process Metrics**

Process metrics assess the effectiveness and quality of the software process. It also determines the maturity of the process and the amount of effort required to develop the project. Software engineers measure specified attributes to improve the quality of the project. In order to improve any process, a set of meaningful metrics are developed to measure its specified characteristics. These metrics are used to obtain indica for to devise a plan for process improvement. With the help of process metrics if becomes easier for a software engineer to assess the efficacy of the software process. Three face namely product, people and technology have a great influence on software quality and organization performance as is from the given diagram:

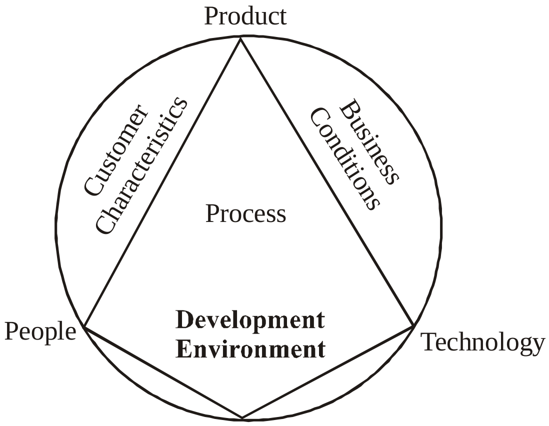


Fig. 3.6: Effect of Factors On Software quality & Organization

The skill and motivation of people, the complexity of the product and the technology level have a great influence on the quality and team performance. For measuring the desired qualities in a software process, a set of metrics is formulated based on different outcomes derived from the process. These outcomes include the following:

(i) Number of errors detected before the release of the software.

(ii) Defects found by the user and reported after the delivery.

(iii) Tune spent in the fixation of errors.

(iv) Work products delivered.

(v) Human efforts used

(vi) Time expended

(vii) Conformity to schedule

(viii) Wait time

(ix) Number of modifications

(x) Estimated cost compared

Types of process metrics: The process metrics are of two types —

(i) Private Metrics,

(ii) Public Metrics.

Private Metrics : This type of metrics are private to the individual and serve as an indicator only for the specified individual(s) e.g. defect rates by a software module and defect errors by an private individual.

Public Metrics: If includes that information which is private to both individuals and teams. E.g. project level defect rates, effort and related data collected analyzed and assessed to get indicators that help in organizational process performance.

Process Metrics Etiquette: Process metrics can be beneficial because the organization works to improve its process maturity. These metrics if mis-used can create problems for the organization. The following are the guidelines to be used by managers and software engineers to avoid misuse of the metrics.

Guidelines

(i) Rational thinking and sensitivity toward organization should be the first priority.
(ii) Feed back to the individuals or teams is very necessary to be provided.
(iii) Metrics should neither appraise nor threaten individuals.
(iv) Use of single metrics should be avoided.

As the process goes on, the derivation of simple indicators lead to a stringent approach called statistical software process improvement (SSPI). SSPI uses software failure analysis to know about all errors and defects faced during the development.

3.16.1.3 **Resource Metric**

Resource metrics is a source code metric and quality analysis tool unlike any other on the market. It provides a standard method for analyzing C, ANSI C++, C# and Java source code across operating systems.

The unique ability of Resource Metric to support virtually any operating system provides enterprise with the ability to standardize. The measurement of source code quality and metrics throughout organization. Resource standard metrics provides the fastest, most flexible and easy to use tool to exist in the measurement of code quality and metrics.

Example Resource Metrics

* Effort expended
— On tasks with a project, classified by life cycle phase software function.
— On extra project activities.
— Training
* Elapsed Time
* Computer Resources

3.16.2 **Design Based Metrics**

The goal of having metrics for the software design is to assess the quality of the design which finally assesses the quality of the product. The quality of the intermediate product is of interest in the development process because it is believed that this leads to a high quality product. The software design complexities are too many and knowledge collected from various fields can be useful in designing systems. It is important to develop software metrics from which meaningful indicators can be derived. With the help of these indicators necessary steps are taken to design the software keeping in view user requirements. Various design metrics such as architectural design metrics, metrics for object oriented design (MOOD), Component Level Design Metrics and user interface design metrics are used to indicate the complexity, quality and so on.

Design metrics can be extracted much earlier in a project and if helps to predict greater number of activities. Large number of Recent Software Design Metrics can be divided into three categories—(i) Network Metrics, (ii) Stability Metrics, (iii) Information flow Metrics. Software metrics measures the product under development and after a system are fielded. Design Metrics includes three activities:

(i) **Total Number of Modules:** This metric specifies the total number of modules. The fundamental benefit of this metric is that by using an average size of a module, a size estimate of the final product can be ascertained. If there is difference in the initial size estimate, it will require a change in plan especially allocation of personal, the schedule and the cost of the product etc. and it will have to be negotiated with the client.

(ii) **Number of modules called:** This metric is computed easily after the completion of the software design. For each module, we can calculate how many other modules if calls and this can also be used to determine how many modules call a particular module. A module with high fan in may point that module has functional cohesion of module with high fan out means that the module depends on to many modules where as a high fan in means that a number of other modules depend on this. During software design, the designer must evaluate those modules that are called by a large number of module, or that call a large number of modules.

(iii) **Number of Parameters:** The software a design specifies all those modules that will be present in the system. It also specifies then inter-connections with other modules. When the software design reaches its completion; a number of parameters of a module can be obtained. Some questions arise about the constitution of a parameter. Whether a complex object in a parameter list be counted as a single parameter? or it should be considered to be made up of many parameters ?

The simplest way is that to consider each logical entity in the parameter list as a parameter. The second way is to think about each basic data which is used in a module as a parameter.

This metric in a way tries to find “Coupling between Modules”. When the module performs so many functions and the parameter list is very large; then a high number of parameters can have low cohesion. Modules with higher number of parameters will be more tightly coupled with their caller modules than those with a small number of parameters.

This means that more effort and time will be needed to understand such modules. Any modification in them will affect other modules as well so the job of the maintainer becomes a bit difficult. It is clear that module with large number of parameters needs careful scrutiny.

3.16.2.1 **Traditional Metrics**

Traditional metrics are applied to the methods that carry out the operations of a class in an object oriented system. The method is a component of an object that operates on data after receiving the message and is defined as part of the declaration of a class. This method demonstrates how a problem is broken into segments and the capabilities other classes expect of a given class. The following three metrics given below are traditional metrics — (i) Cyclomatic Complexity, (ii) Size Metric, (iii) Comment Percentage

The traditional metrics have been widely used. Researchers and practitioners well understand them. The traditional metrics are related to software quality attributes which have been validated.

Metric–1: Cyclomatic Complexity: It is used to evaluate the complexity of an algorithm in a method. A method will low Cyclomatic complexity is considered better. Due to in heritance, cc cannot be used to measure the complexity of a class. Cyclomatic complexity of individual methods can be combined with other measures for the evaluation of complexity of the class. Although this metric is used to evaluate quality attribute complexity; it is also related to all of the other attributes. The formula for testing is equal to the number of edges minus the number of nodes plus 2. Number of independent test paths = Edges – nodes + 2. Number of independent test paths = Edges – nodes + 2

Metric–2: Size Metric: Size of a class is used to evaluate the case of understanding of code by developers and maintainers. Methods to measure size can be counting all physical lines of code, the number of statements and the number of blank lines. Coding language and the complexity of the method form the basis for evaluating the size measures. Size affects the ease of understanding, routines of large size will always pose a higher risk in the attributes of understanding, Reusability and Maintainability. The size metric include — Total lines of code, Total function calls, Number of windows.

Metric—3: Comment Percentage: The line counts which compute the size metric can be expanded further to take into account a count of the number of comments both on line i.e. with code and stand alone. The comment percentage is equal to the total number of comments divided by total lines of code less the number of blank lines. This metrics is used by Developers and Maintainers to evaluate the attributes of Understandability, Reusability and Maintainability.

3.16.2.1.1 **McCabeCyclomatic Complexity (CC)**

- It is a software metric developed by Thomas J. McCabe in 1976. It is a very useful, logical metric. It tests the complexity of the program. It also estimates the amount of effort required to understand the code. It is created by calculating the number of different code paths in the flow of the program. A program that has complex control flow will require more tests to achieve good code coverage and will be less maintainable.
- It measures the number of linearly independent paths through a source code. Cyclomatic complexity is calculated using the control flow graph of the program. The nodes of the graph correspond to indivisible groups of commands where as a directed edge links two nodes, if the second command immediately follows the first command.
- Cyclomatic complexity may also be applicable to individual functions. CC is an indication of the number of 'Linear' segments in a method. A method with no branches has a cyclomatic complexity of since there is 1 arc. This number is incremented whenever a branch is encountered. CC is procedural rather than an OO metric.
- Cyclomatic complexity helps in estimation of complexity of code, testing effort and program reliability.
- **The statements which represent branching as:** 'For', 'while', 'do', 'if', 'case', 'catch' optional and ternary operator (optional). If the source code contained no decision points like 'IF' statements or 'FOR' loops, the complexity will come out to 1 because there is only one path. But if the code had a single IF statement, there would be two paths through the code:

  (i) First path where the IF statement is evaluated as TRUE.
  (ii) Second path where the IF statement is evaluated as FALSE.

- **The formula of the cyclomatic complexity is defines as**:

  CC=E-N+P
  CC= Cyclomatic Complexity
  E= Number of Edges
  N= Number of Nodes
  P= Number of Connected Components

Example 1:

If number of edges= 7, number of nodes= 6 and number of connected components= 2.
Then cyclomatic complexity of ***figure*** *5.4 is:*
CC= 7-6+2 => CC=3

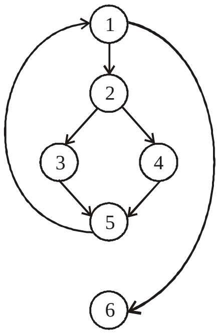


Fig. 3.7: Example of Cyclomatic Complexity

***Example 2**:*

If number of edges= 10, number of nodes= 8 and number of connected components= 1
Then cyclomatic complexity of ***figure*** *5.5 is:*
CC= 10-8+1 => CC=3

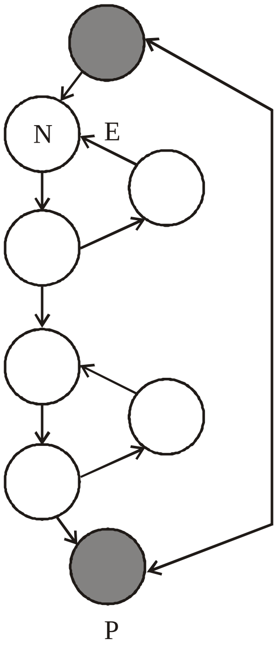


Fig. 3.8: Diagrammatic representation Structured Program

- It is clear CC is calculated by creating a connected graph of the source code. Each line of code is considered as a Node and the arrows between the nodes show the Execution Path. After cyclomatic complexity of a code has been calculated, then we can compare its complexity of other programs as per the standard range tabulated as under:

Cyclomatic Complexity	**Code Complexity**
(a) 1-10	It is a simple program with less risk
(b) 11-20	More complex program with moderate risk
(c) 21-50	Complex with high risk
(d) CC > 50	Untestable having very high risk

- Second method to determine the metric is to count the number of decision points (conditionals) and adding

CC = D + 1 Where D is the number of decision points in the source code.

Advantage of McCabe Cyclomatic complexity

(i) It can be used as a case of maintenance metric.

(ii) When it is used as a Quality Metric, gives relative complexity of various designs.

(iii) It can be computed early in life cycle of halstead's metrics.

(iv) If measures he minimum effort and best areas of concentration for testing.

(v) If guides the testing process by limiting the program logic during development.

(vi) It is easy to apply.

Disadvantages

(i) It does not measure the Data Complexity. It measures only the program's control complexity.

(ii) The same weight is placed on nested and non-nested loops. However, deeply nested conditional structures are harder to understand than non-nested structures.

(iii) CC may give a misleading figure in the case of simple comparisons and decision structures.

3.16.2.1.2 **Source Lines of Code**

SLOC is the most common metric or size today and is most widely used for size estimation. Line of code is easy to compute and enjoys some reputation. If highly depends on programming language because code writing varies from one programming language to another. Simple line metrics like errors per LOC, defects per LOC, cost per LOC etc. can be derived from LOC. Lines of code has also been used to pre-guess program complexity, development effort, and programmer performance.

LOC is the earliest and simplest metric for estimating the effort and size of a computer program even then there is no standard definition of LOC. It is due to this reason, different workers may obtain different counts rural emphasis is given to each line of code. LOC is often used during the testing as well as maintenance phases.

Advantages

It is very simple to measure.

Disadvantages

(i) It is dependent on programming language.

(ii) It does not accommodate non-procedural language.

(iii) Poor software design may lead to excessive and unnecessary line of code.

Guidelines for determining LOC

(i) One LOC is for one logical line of code.

(ii) Only those line of code are included which are delivered as part of software.

(iii) Test drivers, test stubs and other support software are excluded.

(iv) The software code written by the software developer is included whereas code created by the application generator is excluded.

(v) All declarations in the programs are counted as lines of code.

(vi) No comment is counted as lines of code.

3.16.2.2 **Object Oriented Metrics**

Source line of code and Functional Point Metrics can be helpful in estimating object oriented software projects. But these metrics do not succeed in providing details for effort and schedule estimation in incremental software development. Different types of metrics have been suggested for object oriented projects which have been given below:

(i) **Number of Scenario Scripts:** These are a series of steps which depict the interaction between user and application. There is direct relationship between number of scenarios and the application size and number of test cases. Which are developed to test the software. The scenario scripts are analogous to use cases.

(ii) **Number of key classes:** Key classes are independent components. The key classes give indications about effort required to develop software and the amount of 'reuse' feature to be used during the development process.

(iii) **Number of support classes:** Support classes can be defined as those classes which are needed to implement the system but are related indirectly to the problem domain. E.g. user interface classes, computation class. For each key class, a support class can be developed. Support classes and key classes function in a similar way.

(iv) **Average number of support classes per key class:** Key classes are defined very early in the software project whereas support classes are defined throughout the project. If average number of support classes per key class is pre-determined; then estimation process gets simplified.

(v) **Number of sub systems:** Sub system can be defined as a collection of classes that is supporting a function visible to the user. After the sub systems are identified if becomes easy to prepare on appropriate schedule, according to which work on sub systems is divided among project members.

Above mentioned projects along with other metrics like effort used, errors and defects detected are collectively used.

3.16.2.2.1 **CK (Chidamber&Kemerer's Metrics Suite)**

CK metrics for 00 Design is the deepest research / study in object oriented investigation. The following six metrics have been defined for it.

Metric 1 : Weighted Methods per class (WMC) : Let us suppose a class C1
Where M1...............Mn are the methods.
and C1............. Cn be the complexity of the methods
Then the formula for WMC = $\sum_{i=1}^{n} \text{Ci}$
The following points are to be noted:

- * The number and complexity of the method, give an indication of Time and Effort required developing and maintaining the class.
- * The number of methods in a class is proportional to the potential impact on children. Larger the number of methods, greater is the potential impact.
- * Classes having larger number of methods are likely to be more application specific thereby limiting the possibility of reuse.

Metric 2 : Depth of Inheritance (DIT) : In those cases where multiple inheritance is involved; DIT is the maximum length from the node to the root of the tree :

- * If a class is deeper in the hierarchy, if inherits greater number of methods and makes it more complex to predict its behavior.
- * Deeper trees constitute greater design complexity.
- * The deeper a particular class in its hierarchy; the more is potential reuse of inherited methods.

Metric 3 : Number of Children (NOC) : It is defined as the number of immediate sub classes which are under a class in the class hierarchy and are going to inherit the methods of the parent class :

- * If NOC is greater; the Reuse will also be greater because inheritance is a form of Reuse.
- * If NOC is greater; there are greater chances of improper abstraction of the parent class.
- * NOC is linked to potential influence of a class on the design. Greater NOC in a class may require.

Metric 4 : Coupling between object classes (CBO) : Classes having same properties are said to be :

- * Excessive coupling between object classes has a bad effect on the Modular Design and prevents Reuse.

- * Inter object class couples should be minimum to improve Modularity and Promote encapsulation. As the number of couples becomes large; the sensitivity to any change in other parts of the design also becomes large so the maintenance becomes more difficult.
- * The higher the inter object class coupling, the more rigorous the testing needs to be.

Metric 5 : Response for a class (RFC) :

- * It indicates number of methods that can be invoked in response to a message.
- * RFC measures potential communication between classes.
- * When RFC Grows; the complexity of a class becomes more and more whereas understandability decreases.
- * In case of a large number of methods invoked in response to a message; the testing and debugging of the class becomes more complicated and if becomes more difficult for a tester to understand.

Metric 6 : Lack of cohesion in methods (LCOM) : LCOM can be defined as the count of the number of methods pains whose similarity is 0 minus the count of methods pairs whose similarity is not zero. As the number of similar methods in a class becomes more and more; the cohesiveness increases:

- * Cohesiveness promotes encapsulation.
- * Lack of cohesion years that classes should be split into two or more than two sub classes.
- * Low cohesion increases complexity and it increases the chances of errors during the development process.
- * If there is any measure of disparateness of methods; it helps to identify flaws in the design of classes.

3.16.2.2.2 **Mood (Metrics for Object Oriented Design)**

The metrics for Object Orient Design refers to the basic structural mechanism of the 00 paradigm as encapsulation (MHF, DHF) inheritance (MIF and AIF), poly morphism (PF), message passing (CF) and are expressed as Quotients. It include the following metrics:

1. **Method Hiding Factor (MHF) :**It is defined as the ratio of sum of the invisibilities of all methods defined in all classes to the total number of methods defined in the system under consideration. The invisibility of a method is the %age of the total classes from which this method is not visible.
2. **Attribute Hiding Factor (AHF) :**It is the ratio of sum of the invisibilities of all attributes defined in all classes to the total number of attributes defined in the system under consideration.
3. **Method inheritance factor (MIF) :**MIF is defined as the ratio of the sum of the inherited methods in all classes to the total number of available methods (locally defined plus inherited) for all classes of the system under consideration.
4. **Attribute inheritance factor (AIF) :**It is defined as the Ratio of the sum of inherited attributes to the total number of available attributes (locally defined + inherited) for all classes of the system under consideration.
5. **Polymorphism Factor (PF) :** It is the ratio of the actual number of possible different polymorphic situation for all classes ci & the maximum number of possible distinct polymorphic situations for class ci.
6. **Coupling Factor (CF) :**It is defined as the ratio of the maximum possible number of couplings in the system to the actual number of couplings not computable to inheritance.

3.17 **SOFTWARE METRIC APPROACH**

Software Metrics are numerical data linked to the development of the software. They have the following four function to play:

(i) **Planning:** Software metrics lays the foundation for cost estimation, resource planning, scheduling, and budgeting.

(ii) **Organizing:** A projects organization is influenced by size and schedule metrics.

(iii) **Controlling:** They provide an important tool to supervise, monitor and control the progress of various development activities.

(iv) **Improving:** Software metrics are used to ensure a quality product in the end. This is achieved by continuous process improvement.

The following steps are required to select, design and implement software metrics so that a light software metric program is developed:

(i) **Identify Metrics Customer:** This is the first step. Those people are identified who will be using metrics for decision making and implement actions. These customers can be programmers, Testers, Experts and users.

(ii) **Select Measurable Goals:** The second step is to select goals at the organization level and the project level. At the organization level these may below cost provider, higher level of customer satisfaction and profit margins etc. At the project level the goals can be to deliver the software product in time, completion of project within budget and to achieve the desired level of quality.

(iii) **Ask Questions:** Some questions are raised and answers to satisfy these questions are found which will ensure the achievement of the goals. The questions can be:

(a) Whether the software product adequately tested or not?

(b) He all the errors properly corrected?

(iv) **Select Metrics:** The further step is the selection of the metrics suitable to provide information to the people that helps them to take right decisions to ensure the achievement of goals after solving problems.

(v) **Select a Measurement Function:** A measurement function is very necessary. This defines the way how to calculate the metric. Some metrics are measured directly by variables.

(vi) **Collect Data :**After taking the decisions about what to measure and how to measure, the data is collected accordingly to compute the metrics.

(vii) **Provide feedBack :**After the computation of the metrics, the need arises to provide feedback by evaluating the metrics. The patterns indicators and goals identified by the metrics are used to determine the need for action a further investigation or to describe the level of confidence in a given result. Software metric approach is better depicted in the following diagram:

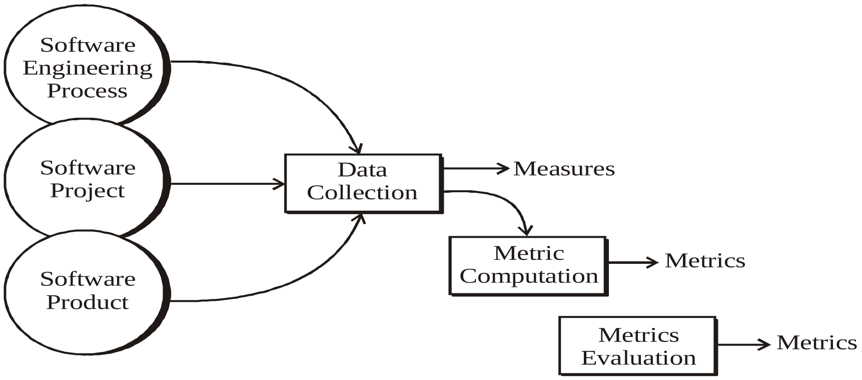


Fig. 3.9: Steps for developing Software Metric program

3.18 **GUIDELINES FOR METRICS**

An ideal software metrics is easy to understand, effective and efficient. An ideal software metrics is always validated and characterized effectively. The following guidelines should be followed to develop metrics:

(i) **Simple and computable:**The derivation of software metrics should be very easy to understand and if should take average among of time and effort.

(ii) **Consistent and objective:** Very clean results should be produced by software metric.

(iii) **Consistency of use of units and dimensions:** Mathematical calculations in metrics should carry the use of dimensions and units in a consistent manner.

(iv) Programming language should be independent and the metrics should be developed on the basis of Analysis Model, design model or program's structure.

(v) **High Quality Product :** If should lead to the production of high quality software product.

(vi) **Easy to Calibrate :** Metrics should be easy to adjust according to the requirements of the product.

(vii) **Easy to obtain:** Metric should be easy to be developed at a reasonable cost.

(viii)**Robust :** Metrics should be relatively insensitive of small change in process project or product.

(ix) **Value :** The value of metrics should be proportional to the value of software characteristic they represent. If one increases or decreases, the other should also increase or decrease relatively.

(x) **Validation :** Metrics should be validated before it is being used for making decisions.

(xi) **Focus :** The main focus of the metrics should be on the product and process.

(xii) **Never to use a single metric:** Software is complex and multifaceted. One should not obsess to much on a single metric. Other metrics should not be avoided.

(xiii)Metrics data should not be ignored or considered negative.

(xiv)Metrics should not be used to threaten the individual or teams.

These people might manipulate the data, the next time. Metrics should never be used as a 'stick'. These guidelines when followed in true sense result into a very high quality metrics.

3.19 QUALITIES OF GOOD SOFTWARE METRICS

A Good software metrics should possess the following characteristics:

(i) **Simplicity:** A good software metrics is very to understand. It should not be complicated.

(ii) **Computability:** It should be easy to compute and derive the metrics. It should not require a lot of time and effort for computation.

(iii) **Objectivity:** It should be of objective and should not be affected by subjective judgments. If should provide clear cut results.

(iv) **Consistency:** The software metrics should show consistency so that even an independent third party should be able to derive the same metric value and using the same type of information.

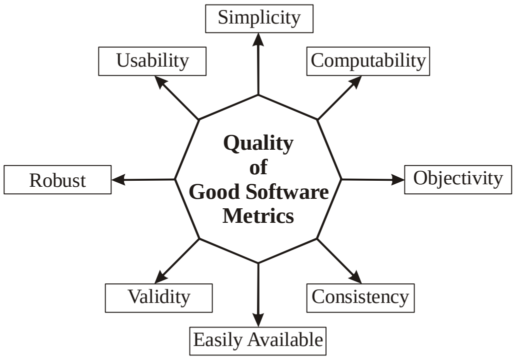


Fig. 3.10 Quality of Good Software Metrics

(v) **Easily Available:** It should not be too much expensive. It should be available at a reasonable cost.

(vi) **Validity :** The metric should measure what is required if the metric should provide valid result.

(vii) **Robust :**If there are minor changes in the product; the metrics should be able to withstand. The changes which can be accidental or intentional security breaches.

(viii)**Usability :** Software metric tries to measure or predict some attribute of some product or process. The software should be user friendly in order to avoid user's reluctance and failures.

3.20 **SOFTWARE QUALITY**

1. Each type of customer will have their own definition of quality. It means quality definition may differ from person to person. For example: an organization defines quality in terms of profits, whereas user defines quality as bug free, within budget and user friendly.

2. Quality can be defined as:

- ***According to Oxford dictionary: Degree of excellence***
- ***According to Edward Deming: Fitness for purpose***
- ***According to ISO: The totality of characteristics of an entity that bear on its ability to satisfy stated or implied needs.***

3. Quality of software depends upon many factors like bug-free, portability, reusability, delivered on time, within budget, user friendly, meet user's requirements and maintainability.

4. It is a relentless process means ongoing process.

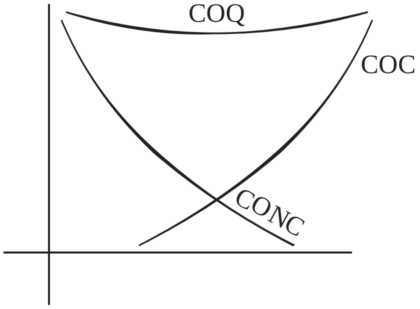
5. The quality system activities encompass the following:

- Audits of projects
- Review of System
- Development of standards and guideline.

6. Cost of quality depends upon the level of testing and maintenance.

Cost of quality = cost of conformance + cost of non-conformance

COQ = COC + CONC



7. Need of Software Quality

(a) If a software has a high quality, it means software has fewer defects. This saves time during testing and maintenance phases.

(b) Higher the quality of software, lower the maintenance cost. It increases the reliability and customer satisfaction.

3.21 **ATTRIBUTES OF QUALITY SOFTWARE**

The attributes are those factors which increases the quality of a software which are as follows:

1. Bug free
2. Portability
3. Reusability
4. Delivered on time
5. Within budget
6. Customer Satisfaction
7. Maintainability
8. User friendly

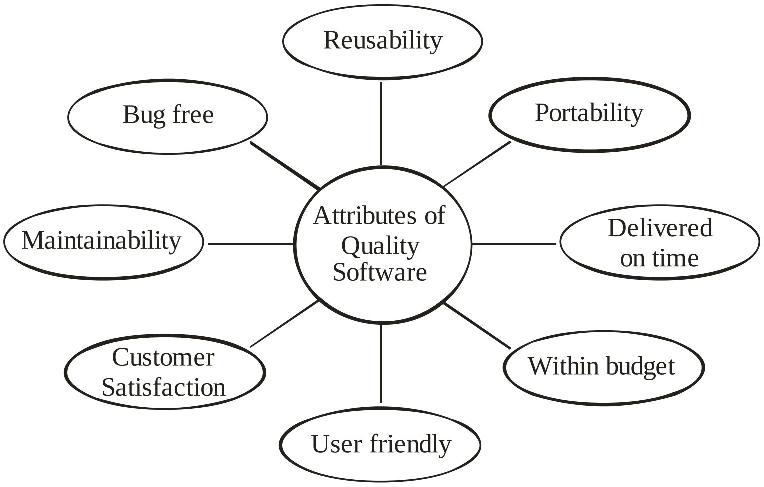


Fig. 3.11 Attributes of Quality Software

1. **Bug-free:**A quality software is always bug free. A software is a quality software if there are lesser number of bugs in the software product.

2. **Portability:** Portability is an attribute in which a software product can be adopted to run on computers (machines) other than the one for which it was designed. According to this feature, a quality software can run on different machines, different operating system environments and with other software products.

3. **Reusability:** According to this feature, we can make some changes in the software product to build a new version of the same product. In this process, minor changes are made in the old software product to make a new software product. In this process, different modules of the product are, reused to develop new product.

4. **Delivered on Time:**A quality software is always delivered on time. It means software is delivered in a fixed time period and within budget.

5. **Within Budget:**A software is called a quality software, if it is under budget of a client. If a client/customer cannot afford a software because of its high cost, then that software is not a quality software for that particular client/customer. A quality software should be within budget of a customer.

6. **User Friendly:** A software product is user friendly if its users find it easy to use. A quality software is always user friendly. A software with many features is never called a quality software if a user faces problems to use it. In a quality software product, users can easily invoke the functions and features of software product.

7. **Customer Satisfaction:** It is another important attribute of software quality product. A software is known as quality software for a particular user if it meets the user's requirements. Software should satisfy the needs/requirements of the customer/user.

8. **Maintainability:**A quality software is maintainable if errors can be easily corrected when they show up, functions of the software product can be easily added to the same software product. This attribute provides the suitability for debugging, extension of functionality and ease of modification.

3.22 **SOFTWARE QUALITY MANAGEMENT SYSTEM**

1. Software quality management system is also known as quality system. Software quality management system ensures that organizations use a adequate methodology to develop a desired quality software.

2. Software quality management system is the responsibility of whole organization. Software quality is not a responsibility of individual person. The top management of the organization should activity participate in this system. If top management of the organization will actively participate in the system, then every member of the organization will take the quality task seriously.

3. Software quality management system involves many activities like auditing and reviewing of the quality system, development of standards, procedures and guidelines etc to enhance the quality of software product.

4. In the software quailty management system, finished products are inspected to eliminate the defective products.

5. This process has mainly four stages which is known as quality assurance method.

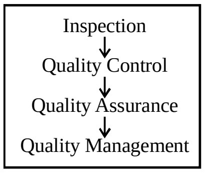


Fig. 3.12 Quality Assurance Method

In the first stage, the product is examined from each angle. In the second stage, quality control finds the defects and eliminate the defects. Quality control also determine the reasons behinds the defects and corrects the reasons of defects. In the third stage, quality assurance takes place. The planned systematic activities necessary to ensure that a component, module or system confirms to established technical requirements. The modern quality paradigm includes certain guidance for recognizing, defining, analyzing and improving the product process. In the last fourth stage, quality management aims at continuous process improvement.

3.22. 1 **Software Quality Management Principles**

The principles of software quality management system are as follows:

Principle 1:Organizations should clearly understand the needs of current and future customers. The aims of the organizations should meet the customer requirements.

Principle 2:Top management of organization should create such an environment in which all people can become fully involved in achieving the organizations common goal.

Principle 3:All people of organization should activily involve in achieving the organization's goals and benefits.

Principle 4:All resources and activities of an organization are managed efficiently to achieve common goal.

Principle 5:Organizations should take effective decisions and actions to achieve desired goals.

Principle 6:Achieving quality is a relentless process means ongoing process.

3.23 **SOFTWARE QUALITY ASSURANCE (SQA)**

1. Software quality assurance (SQA) is a planned and systematic approach to the evaluation of the quality.
2. SQA ensures that the standards, processes and procedures are established and followed throughout the software development life cycle.
3. In the software quality assurance, processes and methods are monitored throughout the software development life cycle.
4. The goal of quality assurance is to prevent introducing defects in the software application which helps to improve the development process. Its aim is prevention of defects to improve the quality.
5. Software quality assurance is pro-active process. It identifies the weaknesses in the process but it does not involve in the execution of program.
6. All people who are involved in the software development process are responsible for quality assurance.
7. Verification is an example of quality assurance.

3.23.1 **Importance of Software Quality Assurance (SQA)**

Software quality assurance (SQA) is an important process in the software development approach to enhance the quality of software product. It is a proactive process used to prevent the defects and to improve the quality of the software.

The following are some points which explains the importance of SQA:

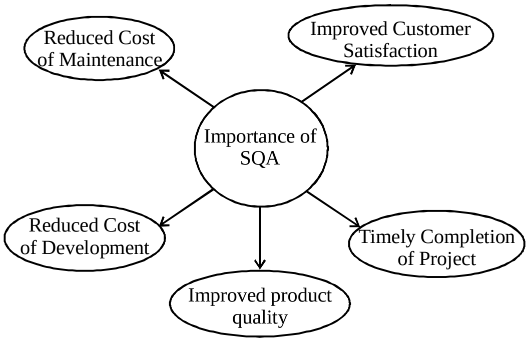

Reduced Cost of Maintenance
Improved Customer Satisfaction
Importance of SQA
Reduced Cost of Development
Timely Completion of Project
Improved product quality

Fig. 3.13 Importance of SQA

1. **Improved Customer Satisfaction:** SQA ensures that the developed software meets the customer's requirements and fulfill their desired goals. SQA ensures that software product does what it is supposed to do. Customer satisfaction is the most important factor, for which quality assurance is done.
2. **Timely Completion of Project:** In SQA process, all testing and development processes go smoothly and quickly. SQA assures that the software product is free from errors/bugs/defects. It means that software development project consistently reach towards completion within budget and time.
3. **Improved Product Quality:** The quality of software product gets improved by following standards, procedures and actions. The aim of quality assurance is to prevent introducing defects in the software product which improve the product quality.
4. **Reduced Cost of Development:** In this process, developers are working with SQA team. Both developers and SQA team monitors the software development process from planning to implementation stage to maintain quality. In this way, there are lesser chance of an error to come up in the software development process. The developers don't need to go back to the application and fix something all the time and cost of development is automatically reduced.
5. **Reduced Cost of Maintenance:** If a software product is properly quality assured it means software is bug free. It will give less chances to users to complain and time spent on maintenance will be much less.

3.23.2 **Activities of Software Quality Assurance(SQA)**

The Software Quality Assurance is the process of evaluating the quality of a product. It is an umbrella activity that ensures conformance to standards and procedures throughout the SDLC of a software product. For the enhancement of quality in the software product, the following activities are performed:

1. Formulating a quality management plan
2. Applying software engineering techniques
3. Conducting formal technical reviews
4. Applying a multi-tiered testing strategy
5. Enforcing process adherence
6. Controlling change
7. Performing SQA audits
8. Keeping records and reporting

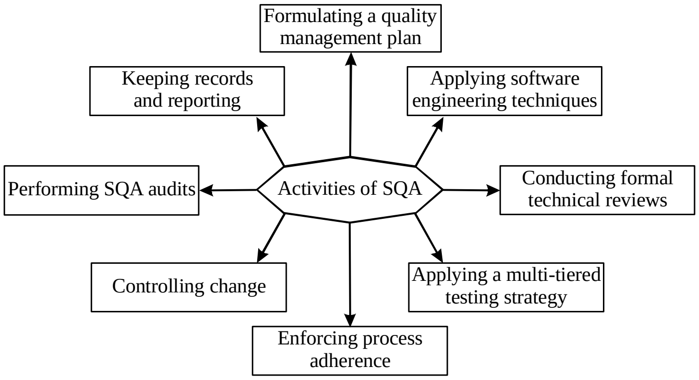

Formulating a quality management plan
Keeping records and reporting
Applying software engineering techniques
Performing SQA audits
Activities of SQA
Conducting formal technical reviews
Controlling change
Applying a multi-tiered testing strategy
Enforcing process adherence

Fig. 3.14 Activities of SQA

1. **Formulating a Quality Management Plan:** The first task of SQA is the formulation of a quality management plan.

- Thequality management plan identifies the quality aspects of the software product tobe developed.
- Plan check points for work products and thedevelopment process.
- Tracks changes made to the development processbased on the results of the checks.

2. **Applying Software Engineering:** The software engineering theories and practical techniques help the software designer to achieve high quality specification.

- Information gathering techniques help designer to collected information.
- The project estimation techniques calculate the estimation of the project.

3. **Conducting Formal Technical Reviews (FTR):** The formal technical review (FTR) in conducted to perform following activities:

- To assess the quality and design of the prototype.
- It is a meeting with the technical staff to discuss the quality requirements of a software product and its design quality.
- It helps in detecting errors at an early phase of development.
- It prevents errors for moving down to the latter phases and resulting in rework.

4. **Applying a Multi-tiered Testing Strategy:** The software testing is a primary task of SQA activity which is described as

- Error detection and correction in project.
- Different types of testing is performed in sequence, fest of all unit testing is performed and in the subsequence levels, integration testing and system level testing are applied.

5. **Enforcing Process Adherence:** It emphasizes the need for process adherence during product development. It is a combination of two tasks, product evaluation and process monitoring.

6. **Product Evaluation:** The product evaluation is conducted to ensure the following:

- To ensure that the standards lay down for a project are followed or not.
- To ensure that the compliance of the software product to the existing standards is verified.
- To ensure that the software product reflects the requirements identified in the project management plan

7. **Process Monitoring:** The process monitoring ensures and monitors the following activities:

- Does the appropriate steps to follow the product development procedures are carried out.
- Monitors processes by comparing the actual process with the written documented procedures.
- It ensures that the development and control processes described in the project management plan are correctly carried out.
- It ensure that products and processes confirm to standards and procedures.
- Audits ensure that product evaluation and process monitoring are performed.

8. **Controlling Change:** It evolves both human and automated procedures to provide a mechanism for change control. The change control mechanism is implemented during the development and maintenance stages.

- It ensures software quality by formalizing requests for change.
- It evaluates the nature of change.
- It controls the impact of change.

9. **Performing SQA Audits:** TheSQA audits scrutinize the software development process by comparing it to established processes.

- It ensures that proper control is maintained over the documents required during SDLC.
- It ensure that the status of an activity performed by the developer is reflected in the status report of the developer.

10. **Keeping Records and Reporting:** The keeping records and reporting involves the process of collection and circulation of information relevant to SQA. The results of reviews, audits, change control, testing, and other SQA activities are reported and preserved for future reference.

3.23.3 Software Quality Metrics

Software quality metrics are used to express the quality of software product. No single metric is there which appropriately explains the quality of the software product. Software quality metrics are some measurements which are related to system, process and documentation of software. Software quality metrics are divided into three types which are as follows:

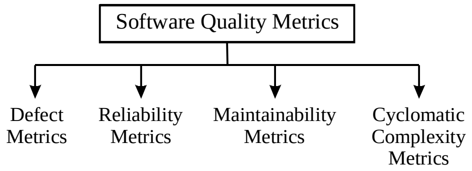


Fig. 3.15 Metrics of Software

1. **Defect Metrics:** Defect metrics provide the number of defects observed in a particular software product. It is used to reflect the man-hours spend during testing to resolve the defects. Defect metrics also explain the types of defects like computation defects, usability defects, requirement objects, design defects etc. to prevent the future problems.

2. **Reliability Metrics:** Reliability metrics are used to know the probability of software failure or the rate at which software will occur. Reliability metrics are discussed in chapter 11. 3.

Maintainability Metrics: Maintainability metrics are used to measure the maintainability of the software product. The maintainability of a program is related to its complexity. The complexity is measured by assuming no. & frequency of operators, no. & frequency of operands.

4. **Cyclomatic Complexity Metric :** It is a very useful and logical metric. It tests the complexity of the program. It also estimates the amount of effort required to understand the code. It is created by calculating the number of different code paths in the flow of the program. A program that has complex control flow will require more tests to achieve good code coverage and will be less maintainable. It measures the number of linearly independent paths through a source code. Cyclomatic complexity helps in estimation of complexity of code, testing effort and program reliability.

3.23.4 Principles of SQA

The SQA team has to follow some principles which ensure that the application will live up to the expectations of the users. Some commonly known important principles which are used for proper execution of software quality assurance (SQA) are as follows:

1. **Feedback:** The time is always a best friend and biggest enemy of any developer. If SQA team give feedback early to the developer, then developer can easily take action. In short, faster the feedback, faster the development will move forward.

2. **Motivation:** Quality assurance is a very tedious task. SQA team members should be motivated, passionate and have right mindset so that they can work efficiently, effectively and with creativity.

3. **Evolution:** This principle is for future use. SQA team should be able to mark every time something new is done. Every time something new happens, it should be always noted.

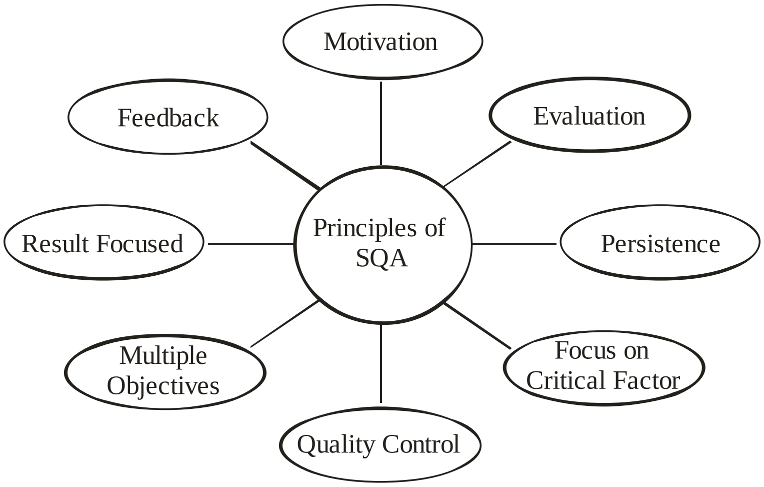


Fig. 3.16 Principles of SQA

4. **Persistence:** The SQA team members should be very patient in every aspect of the software development process. There is no perfect application and every port should be scrutinized without hesitation.

5. **Focus on Critical Factors:** There are many factors developed in the software are not as critical compared to others. SQA team members should be focus on more critical areas first.

6. **Quality Control:** Quality control is the pillar of quality assurance. Everything needs to have quality control from start to the finish. The biggest and tightest quality control should be executed as early as possible which emphasis on where to start.

7. **Multiple Objectives:** This principle is a challenge and risk for SQA team members. At the start of SQA planning, the SQA team members should have more than one objective and each objective should be focused on.

8. **Result Focused:** The SQA process should always look for results whenever a phase is set. SQA should be result focused because it effects to the clients and users.

3.23.5 Software Quality Assurance Standards

The following are some standards which assure the quality in a software product:

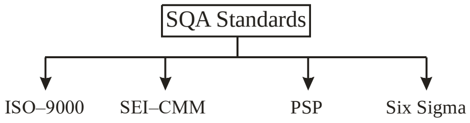


Fig. 3.17 Standards of SQA

I. ISO-9000

1. ISO is also known as International Standard Organization. It is a non-governmental organization. It is a network of 63 countries which is established to formulate standardization. In 1987, ISO published its 9000 series of standards. ISO 9000 interpret the ISO standards for software industry in 1991 because manufacturing industry and software industry have different standards. In India, ISO certification is offered by BIS (Bureau of Indian Standards) and IRQS (Indian Register Quality System).

2. **ISO 9000 Certification:** ISO 9000 certification is a contract between independent parties. It is basically a set of guidelines. It is a series of three standards which are as follows:

- ISO 9001: It is used in software development organizations. It is applicable to organizations which are engaged in design, development and servicing of goods.
- ISO 9002: It do not design products. It is involved in production only.
- ISO 9003: This standard is applicable to those organizations which are involved in installation and testing only.

3. **Reasons to get ISO 9000 Certifications:**

- If an organization have ISO 9000 certification, confidence of customers in that organization increases.
- This certification provides basic framework for the development of the software product.
- ISO 9000 certification points out the weak points of an organization and recommends some actions to the organization so that they can improve those weak points.
- This certification helps the organization to make development process more focused, effective, efficient and cost-effective. It increases the quality of a particular software product.

4. **Procedure to get ISO 9000 Certification**

- An organization who wants to get ISO 9000 certification, has to register itself to a registrar. It is known as application stage.

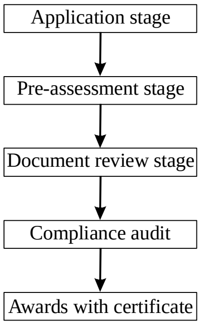


Fig. 3.18 Procedure to get ISO 9000 Certification

- After the registration, the registrar makes rough assessment of an organization. It is known as pre-assessment stage.
- In the third stage, the registrar reviews all the documents which are submitted by an organization during application stage. It is known as document review stage. The registrar suggests some possible improvements to the organization.
- During the fourth stage, which is known as compliance audit, registrar checks whether the organizations made those improvement or not which were suggested by him during document review stage.
- At the last stage, the registrar awards an organization with ISO 9000 certification after the successful implementation of all improvements suggested by the registrar.

After providing the ISO 9000 certification to the organization, the registrar monitors the organizations periodically to check whether the organizations maintain their quality or not. This process is known as continued surveillance.

5. **Issues related to ISO 9000 Certification**

- ISO 9000 certification does not provide any guidelines for an appropriate process and does not guarantees the process to be of high quality.
- There is no international accreditation agency exists which takes guarantee of ISO 9000 certification. Certification is awarded by non-governmental organization.

- After getting the certification, many organizations fail to maintain the quality.

II. **SEI-CMM**

1. It is also known as software engineering institute capability maturity model. It was proposed by SEI, USA in 1986.
2. It is a referenced model and used in two ways: capability evaluation and software process assessment.
3. It is intended to help software organizations to improve the maturity of their software processes. The focus is on identifying the key process areas.
4. SEI-CMM five maturity levels which are as follows:

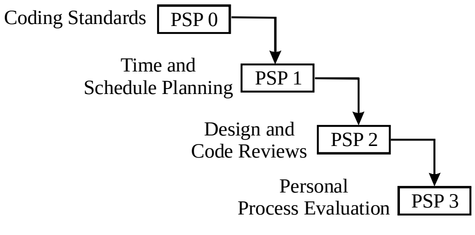
- **Levels 1—Initial Level:** It is characterized by adhoc activities. It is also called chaotic level because different engineer follow their own process. No standard process is defined and followed during the software development. When the engineers leave the project, then it becomes very difficult for new engineer to understand the process.
- **Level 2—Repeatable:** At this level, size and cost estimation techniques like COCOMO, function point etc. are used to establish the tracking cost and schedule.
- **Level 3—Defined:** ISO 9000 aims to achieve this level. In this level, review techniques are emphasized to achieve phase containment of errors. Development and management activities are defined and documented.
- **Level 4—Managed:** At this level, process metrics and product metrics are collected and measured. Basically focus is on software metrics. Pareto charts, diagrams etc. are used to measure process and metric product metric.
- **Level 5—Optimizing:** For continuous process improvement, process and product measurement data are analyzed in detail. In this manner, organization finds out the best software engineering practices and innovations.

III. **PSP—Personal Software Process**

1. PSP is suitable for individual use. It is a scaled down version of industrial software process.
2. It is a framework that helps engineers to measure and improve the way they work.
3. It does not tell engineers how to analyze, design, code and test the software product. It just helps them to improve their work.
4. The methods of PSP are as follows:

(a) **Time measurement:** Use stopwatch to get an objective picture of the time spent.

(b) **PSP planning:** PSP planning helps individuals to plan their project.



Levels of PSP

Fig. 3.19 Levels of PSP

IV. **Six Sigma**

1. The purpose of six sigma is to improve process to do things better, faster and at lower cost. It was pioneered by Motorola in 1995.
2. Six sigma is a rigorous and disciplined methodology that uses data and statistical analysis.

3. It improves the company's operational performance by identifying and eliminating defects from products.

3. Six sigma can be defined and understand at three distinct levels—metric, methodology and philosophy. It is applicable to every industry.

3.24 SOFTWARE QUALITY CONTROL (SQC)

1. Quality control is the pillar of quality assurance. Everything needs to have quality control from start to the finish.
2. Quality control is used to verify the quality of the output.
3. The aim of the quality control is to detect the defects and corrects the defects.

4. Quality control is a Re-active technique. It is product oriented. The testing team members are responsible for quality control.
5. Activities of quality control are : Reviews, Testing etc.
6. In short, quality control is a subset of quality assurance.

3.25 DIFFERENCE BETWEEN QUALITY ASSURANCE AND QUALITY CONTROL

Sr. No.	Quality Assurance	Quality Control
1	Quality assurance is the process of managing quality.	Quality control is used to verify the quality.
2	Quality assurance is a pro-active technique. It identifies weakness in the process	Quality control is a Re-active technique. It identifies the defects and corrects the defects.
3	The aims of the quality assurance is to prevent introducing defects in the software application.	The aim of the quality control is to detect the defects in the software application.
4	It is process oriented.	It is product oriented.
5	It does not involve executing the program.	It involves executing the program
6	Example—verification	Example—validation or software testing.
7	It is done before quality control.	It is done after quality assurance.
8	It is a staff function.	It is a line function
9	All people who are involved in the software development process are responsible for quality assurance.	Testing team members are responsible for quality control.
10	It is a set of activities for ensuring quality in software engineering processes.	It is a set of activities for ensuring quality in software products.
11	Activities are—Audits, Training, Process definition and Implementation.	Activities are—Reviews, Testing
12	It is a subset of software test life cycle (STLC).	It is a subset of quality assurance (QA).

M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 4: SOFTWARE REQUIREMENT ANALYSIS

4.1 **INTRODUCTION**

The software development must be based on the requirements raised by the customer. The requirements must reflect the user’s view in both for designing and implementation. The documentation of user requirements is done in a systematic in the form of a document SRS. It is written in a natural language, and contains a description of what the system will do without describing how it will do it. It is designed to describe user requirements, system requirements, limitation, functions, economic and social issues. The following section describes requirement analysis, need of SRS, format of SRS and detail about different tools.

4.2 **SYSTEM ANALYSIS**

1. System analysis is a planned and systematic investigation to determine the functions of a particular system.
2. Its aim is to study the application area and its problems. For this purpose, study involves many data gathering methods like interviews, consultations and observations etc.
3. A person called system analyst conducts this study and identifies various activities, objectives and procedures to achieve his goal.

4.2.1 **Role of System Analyst in Software Development Process**

System analyst is an engineer who gathers and analyzes the customer requirements. System analyst gathers data and analyzes the gathered data to understand the customer's requirements. The role of system analyst is as follows:

1. System analyst defines problem in a precise and cleared manner.
2. System analyst uses many data gathering methods like interviews, consultations and observations.
3. After the data collection, system analyst analyzes the collected data to get the correct solution of a problem.
4. System analyst also studies the pros and cons of various proposed solutions and choose the one best solution for a problem.
5. System analyst develops plans, design those plans with standards to achieve goals.
6. System analyst visit to customer and make SRS.

4.3 **REQUIREMENT GATHERING AND ANALYSIS**

1. Many projects are failed to satisfy customers. This is because developers started developing the software without clear understanding of what the customers exactly wanted.

2. Whether a developer uses his/her complete knowledge to develop software, but it is impossible to develop a satisfactory solution without a clear understanding of a problem. If a developer fails to understand the customer's requirement, he/she is unable to develop satisfactory software.

3. Today, in many organizations, developers start spending considerable time to understand the exact requirements of the customer. This is known as requirement analysis and specification phase.

4. The aim of the requirement analysis and specification phase is to fully understand the customer requirements and to systematically document these specifications. SRS (software requirement specification) document is the final outcome of the requirement analysis and specification phase.

5. The main activities performed during the requirement analysis and specification phase are:

(I) Requirement Gathering
(II) Requirement Analysis
(III) Requirement Specification
(IV) Requirement Validation

(I) **Requirement Gathering:** It is also known as requirement elicitation.

1. In the requirement gathering activity, system analyst collects the data. It sounds very simple but it is very difficult process. It is very difficult to collect the relevant and useful information from large number of people.
2. For this purpose, working model is used. This working model helps the system analyst to collect the customer's requirements. But in the absence of working model, system analyst uses his/her experience, imagination and creativity for collecting customer's requirements.
3. System analyst uses various methods/ways to collect the customer's requirements which are as follows:

(i) ***Interview:*** Interview is an effective data collection method. It is basically consist of asking questions, listening to individuals and recording their responses. In this method, one is interviewer and one is interviewee. Interviewer is one who takes interview and interviewee is one who gives interview. It can be conducted individually or as a group.

The following are the types of interview:

— ***Informational Interview:*** The objective of this interview is to ask for advice and gain more knowledge about a particular field. The knowledge that we gain here will be sharper and more informed.

— ***Telephonic Interview:*** A phone interview is a very cost effective way to collection information. It can last anywhere from 10 to 30 minutes. It is very challenging because interviewer can't see body language of interviewee.

— ***Face to Face Interview:*** It is sometimes known as individual interview. It is based on evaluation method, individual youth, volunteers or parents. The number of interviews and selection of interviewees will depend upon the purpose, time and resources.

Advantages of Interview:

- It provides deep and free response.
- It is flexible and adaptable method.
- It glimpse into respondent's tone, gestures.
- It has ability to probe, follow-up, clarify misunderstanding about questions

Disadvantages of Interview:

- It is costly in time and personnel.
- Sometimes, it become impractical with large numbers of respondents.
- It requires skill among interviewer and interviewee.
- It may be difficult to summarize responses.
- There may be chances of biasness among interviewer, respondent, situation

(ii) ***Questionnaire:*** In this type of data collection method, questionnaire is made which consists of multiple questions. These questionnaires are sent to people via mail, courier or by hand. This is a useful technique because no one knows your intention and body language. But it is the time consuming process.

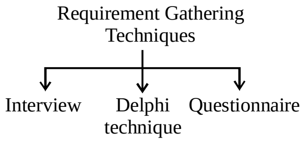


Fig. 4.1 Requirement Gathering Techniques

(iii) ***Delphi technique:*** The Delphi technique is a reliable and creative method. It begins with the development of a set of open-ended questions on a specific issue. These questions are then distributed to various experts. The responses to these questions are summarized and a second set of questions that seek to clarify areas of agreement and disagreement is formulated and distributed to the same group of experts.

Advantages of Delphi Technique:

1. It is conducted in writing and does not require face-to-face meetings.
2. It helps to keep attention directly on the issue.
3. It is inexpensive.

Disadvantages of Delphi Technique:

1. Its information comes from a selected group of people and may not be representative.
2. Its tendency is to eliminate extreme positions.
3. It is a time-consuming technique.
4. It requires skill in written communication.
5. It requires adequate time and participant commitment.

(II) **Requirement Analysis:**

1. The aim of requirement analysis activity is to analyze the gathered data and to obtain the clear understanding of the software product which is going to be developed.
2. The purpose of this activity is to fully understand the exact requirements of the customer and remove all ambiguities, incompleteness and inconsistencies in the requirements.
3. With the help of requirement analysis activity, the system analyst is able to clearly understand the following basic questions:

- What is the problem?
- What are the exactly data input to the system?
- What are the exactly data output to the system?
- What are the possible procedures?
- What are the possible solutions?
- Why is it important to solve the problem?

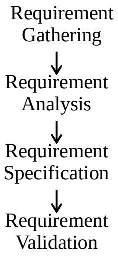


Fig. 4.2 Steps of Requirement Gathering and Analysis Process

(III) **Requirement Specification**

1. During the software requirement specification activity, analyzed information is translated into document form which defines the set of requirements. This document is known as software requirement specification.
2. The document is basically an agreement or contract between the customer and suppliers on what the software product is going to do.
3. The document is prepared by the senior analyst (SA).
4. The document specifies two types of requirements i.e. user requirements and system requirements.

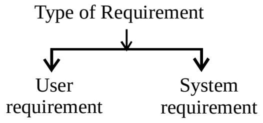


Fig. 4.3 Type of Requirement

User requirements are detailed description of customer requirements. Whereas system requirements are detailed description of the functionality to be provided by the system.

(IV) **Requirement Validation:**

1. In the requirement validation activity, it is checked that whether the requirements mentioned in the requirement specification are complete and consistent or not.
2. In this activity, errors are discovered in the requirement documentation. Requirement documentation is modified to remove the discovered errors.

4.4 SOFTWARE REQUIREMENT SPECIFICATION (SRS)

1. SRS is prepared by the senior analyst (SA).
2. SRS is basically an aggrement between the customer and suppliers on what the software product is going to do.
3. It is the final outcome of requirement gathering and analysis phase.
4. It is a complete description of behavior. It describes the scope of the product.
5. SRS document specifies the certain functions of a particular software product in a specific environment.
6. SRS document is very difficult to write. It covers all the requirements of the customer, functional and non-functional capabilities of the software product.
7. All functions/tasks of software development process are based on the software requirement specification (SRS).
8. Users of SRS document: Many people use SRS document according to their needs. The users of SRS document are: customers, software developers, test engineers, maintenance engineers etc. Software developers use SRS document to ensure that they are developing the software product according to the customer's requirements. Maintenance engineers use SRS document to understand the functionality of the software product.

4.4.1 Need of SRS document

1. SRS removes the communication gap between customer and supplier.
2. SRS document helps the client's/customer's to fully understand his/her own needs along with the developer's capabilities. It forces client to think and visualize his/her needs and discuss with developer in detail.
3. SRS document almost guarantees that after the development of software product, the client will be happy with this project.
4. With the help of SRS document, the client can determine whether the software product meets his/her requirements or not.
5. If a good quality SRS document is used in the software development process, then the developed software product is of high quality and low cost. The software can be developed and delivered in a fixed duration.

4.4.2 Attributes/characteristics of SRS document

The following are the attributes or characteristics of SRS document which enhances the quality of a software product.

1. SRS should be complete. The complete SRS specifies everything about the software product. The complete SRS includes a table of contents, page number, figure numbers etc.
2. SRS should be modifiable in easy manner when requirements are changed.
3. SRS should be unambiguous. Single requirement should have only one interpretation. Single requirement with many interpretation leads to confusion state and developed unsatisfactory software product.
4. SRS should be precise. It is used as a blue print for the coding in a program. It should also follow writing standards. SRS should precisely defines the system's capabilities along with interfaces.
5. SRS should be accurate. It should specifies exact requirements of the customer and system. The correct SRS clearly explains the functional and non-functional capabilities of the system.
6. SRS should be testable. The good SRS document easily identifies and removes its errors.
7. SRS should be traceable. Each requirement in SRS document should be traceable from source to implementation. It should also be able to work in backward direction means from implementation to source in necessary conditions.
8. SRS document should be consistent. Its capabilities, functions and performance levels should compatible and consistent. An SRS should not suffer with conflicting terms, conflicting characteristics and contradictory specifications.
9. SRS should be verifiable. Every single requirement in the document should be verifiable against the developed software product. It removes the communication gap between client and supplier.
10. SRS should be valid. It should be written in natural language so that all participants can easily understand, analyze and accept it.
11. SRS should be well structured. A well-structured SRS document is easy to understand and modify.

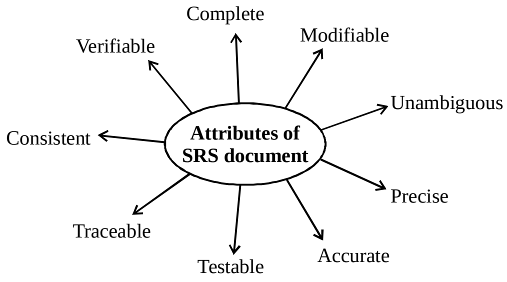


Fig. 4.4 Attributes of SRS Document

4.4.3 Structure/Organization of SRS

The SRS document should be organized in proper sections. The following is the structure of SRS document.

1. Introduction
    a. Purpose
    b. Overview
    c. Environmental Characteristics
        - Hardware
        - Peripheral
        - People
    d. Definitions, Acronyms and Abbreviations
2. Goal of Implementation
3. Functional Requirements
    a. User Class I
        - Functional Requirement
        - Functional Requirements
    b. User Class II
        - Functional Requirements
        - Functional Requirements and so on
4. Non-Functional Requirements
    a. External Interface
        (i) User Interface
        (ii) Software Interface
        (iii) Communication Interface
    b. Constraints
    c. Performance Requirements.
5. Behavioural Description
    a. System States
    b. Events and Actions

Fig. 4.5 Structure of SRS

The following are the explanation of structure of SRS document. These sections are also known as components of SRS document.

1. ***Introduction:*** It explains the context in which the system in being developed. It provides the overall description and environment characteristics of the system. The environmental characteristics specifies the hardware (central processing unit usage, memory usage, network communication etc.) on which the system will run. It also explains the people who are involved in the development and usage of the system. This section also explains the definitions, acronyms and abbreviations such as QoS: Quality of Service. It meets what users actually want and maximize their utility, GUI: Graphical user interface, CPU: Central processing unit of a computer etc.

2. ***Goal of Implementation***: It offers some general suggestions regarding software development. It provides guidelines for the successful development of software product. It is not tested by the user at the time of acceptance testing.

3. ***Functional Requirements:*** It is the critical part of the SRS. It explains all the functionalities required by the user. It specifies which output should be produced from the given input.All the operations to be performed on input data is specified in this document. It describes the relationship between input and output of the system. For each functional requirement, a detailed description of all the inputs, their resources, unit of measure and the range of valid inputs must be specified.

4. ***Non-Functional Requirements:*** This section explains the characteristics of the system such as maintainability, portability, throughput, usability etc. It explains the external interface

(user interface, software interface and communication interface) of the system.It also specify the constraints which restrict the choice of design like security, reliability and fault tolerance. All the requirements related to performance are clearly specifies. The performance requirements are of two types:

- ***Static requirements:*** These requirements do not impose constraints on the execution behaviour of the system.
- ***Dynamic requirements:*** These requirements specify constraints on the execution behaviour of the system.

5. ***Behavioural Description***: It is used to specify the possible states of the system and the transitions among these states due to occurrence of the system. It is not necessary for all the system.

4.4.4 **Reasons of Bad SRS Document**

The following are the reasons of bad SRS document:

1. ***Ambiguous***:The SRS document specify an ambiguity in the requirement. It arises many interpretations of that requirement.
2. ***Incomplete:*** Some requirements have been overlooked and the document left many important aspects of the requirement.
3. ***Inconsistent:*** The requirements in SRS document contradicts each other.
4. ***Unstructured:*** The document is not in structured manner and makes confusion among developers and users.
5. ***Over specification:*** It restricts the freedom of the designer in arriving at good design.
6. ***Wishful thinking:*** When the document specify those requirements which are very difficult to achieve.

4.5 **STRUCTURED ANALYSIS**

1. Structured analysis is the activity of deriving a structured model of the system requirements.
2. Structured analysis techniques help an analyst to decide what type of information is obtained at different points in analysis.
3. It helps to organize information in such a manner that analyst is not over-burdened with complex problems.
4. It is basically a technique/methodology which is used to build a system model which explains the flow and content of information.
5. Special notations and symbols are used to describe the system functionality.

4.5.1 **Objectives of Structured Analysis**

1. The aim of the structured analysis is to define a set of requirements that can be validated once the software is built.
2. It describes the requirements of the customer.
3. It establishes a base for the software design creation.
4. It is basically a documentation for the system which describes the clear and complete specification.

4.5.2 **Principles of Structured Analysis**

The structured analysis approach follows the following principles:

1. The behavior of the system must be represented.
2. The functions that the software is to perform must be defined.
3. The analysis process should move from essential information towards implementation detail.
4. The model should be partitioned in such a way that details should show in a hierarchical manner.
5. The problem of the software must be understood and represented.

4.6 **TOOLS FOR STRUCTURED ANALYSIS**

The following are the tools used for structured analysis:

(I) Data Flow Diagram (DFD)

(II) Data Dictionary (DD)

(III) Process Specification (P-Spec)

(IV) State Transition Diagram (STD)

(V) Entity Relationship Diagram (ERD)

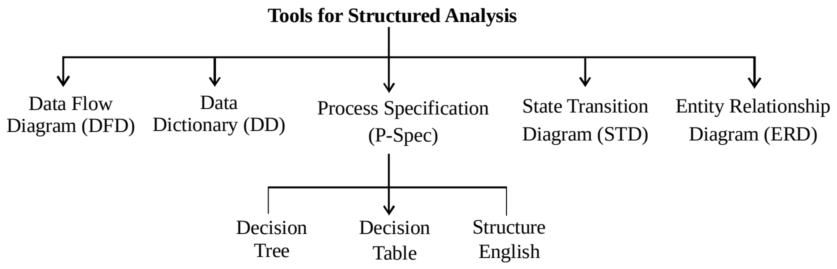

Tools for Structured Analysis
Data Flow Diagram (DFD)
Data Dictionary (DD)
Process Specification (P-Spec)
State Transition Diagram (STD)
Entity Relationship Diagram (ERD)
Decision Tree
Decision Table
Structure English

Fig. 4.6 Structured Analysis Tools

I. **Data Flow Diagram (DFD):**

1. It was developed by Larry Constantine. It is also known as bubble chart.
2. It is a graphic representation of the flow of data or information through a system
3. It represents the logical data flow rather than how they are processed. It only show the flow of data through the system.
4. It consists of a series of bubbles joined by lines.The bubbles represent data transformations and lines represent data flow.
5. DFD can be partitioned into levels. Each level has more information flow and data functional details than the previous level.

6. ***Symbols of the DFD:*** There are different types of symbols used to construct DFDs. The meaning of eachsymbol is explained below:

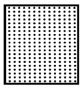
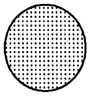
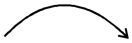
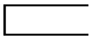
Sr. No.	Symbol	Meaning
1.		Source or destination of data
2.		Process which tranforms data flow
3.		Data flow
4.		Data Store

Table 4.1

7. ***Rules for the construction of DFD:***

(i) The direction of flow of data is from top to bottom and from left to right.

(ii) The names of sources, data stores and destinations are written in capital letters.

(iii) The processes are named and numbered for easy reference. When a process is exploded into lower level details, it is also numbered.

8. ***Level of DFD:***

(i) ***Level 0:*** It is also known as context level DFDs or highest abstraction level DFDs. It depicts the entire information system as one diagram covering all the essential details. It represents scope of the system and identifies external entities and related inputs and outputs

(ii) ***Level 1:*** It depicts basic modules and flow of data among various modules The Level 0 DFD is broken down into more specific, Level 1 DFD. It provides overview of full system. It identifies major processes and data stores.

(iii) ***Level 2:*** At this level, DFD shows how data flows inside the modules mentioned in Level 1. It describes the deeper level of understanding unless the desired level of specification is achieved. Level 1 process is expanded into more detail.

Advantages of DFD:

1. Data flow diagrams are easy to understand check and change data.
2. It give a very clear and simple look at the organization of the interfaces between an application and the people or other applications that use it.
3. It is an effective and efficient method in the absence of required design.
4. It is not limited to software.

Disadvantages of DFD:

1. Modification to a data layout in data flow diagram may cause the entire layout to be changed.
2. Maintenance of data flow diagram is harder, more costly and error prone because in the large application, number of units in a DFD are high and changes are impractical to be made on DFDs.

II. **Data Dictionary (DD):**

1. It is also known as Meta data. A data dictionary is a structured repository of data about data.
2. It is a centralized collection of information.
3. It stores meaning and origin of data, its relationship with other data, data format for usage etc.
4. It removes any chance of ambiguity and helps to synchronize the working of programmers and designers.
5. It is an important and essential step in building a database.
6. Data dictionary contains following information:

(i) ***Data Element:*** It is a smallest unit of data which has no further decomposition. It provides the description of data like: name, source, date of origin, users etc.

(ii) ***Data Structure:*** A data structure is a systematic way of organizing and accessing data. It explains how data are stored in a computer.

(iii) ***Data Flow:*** Data flows are data structures in motion.

(iv) ***Data Stores***: Data flows are data structures at rest. It includes files and tables which stores the information from where the data enters into the system and exists out of the system.

III. Process Specification (P-Spec):

1. It is used to define all the processes which appear in the DFD.
2. All bottom level processes are specified.
3. The tools of process specification are:

Decision Tree	• It is a way of breaking down the complicated situation into easier to understand scenario. • Graphical technique representing decisions using a series of nodes and branches • Each node is a decision point - a choice has to be made • Each branch has a corresponding value to the decision choice • Subsequent action is the result. • It is easy to understand, no need for special training.
Decision Table	• Representation of logic that is part of the processing • Based on a set of conditions, different actions will be performed • Can be simplified by removing impossible actions • Used when the process result is based around several different variables and the logic gets too complex for other methods
Structure English	• In Structured English, decisions are made through IF, THEN and ELSE statements. • It can use CASE construction as well as LOOP construction. • It is basically a subset of English with restrictions on usage. • It can use some acceptable verbs such as FIND, ADD, SUBTRACT, MULTIPLY, DELETE etc.

Table : 4.2

IV. State Transition Diagram (STD):

- It represent a process specification for a control bubble in DFD.

- It is a tool for representing the design.
- It has one initial state and may have multiple final states.
- **State transition diagrams have 4 components :**

States	It is some behavior of a system that is observable and that lasts for some period of time like doing, waiting etc.
Transitions	It is an instantaneous change in state/behavior.
Conditions	A condition is typically some kind of event. It is an event in the external environment which triggers a transition to a new state.
Actions	An action is the appropriate output or response to the event. It is a response sent back to the external environment whose result is stored by the system.

Table : 4.3

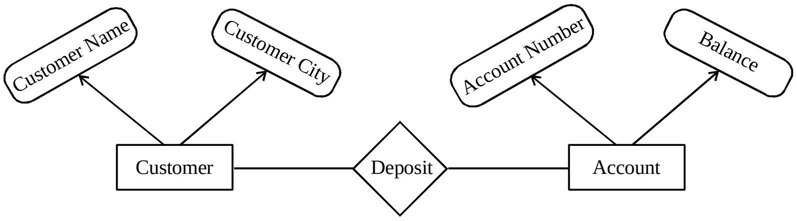
V. Entity Relationship Diagram (ERD)

1. Entity relationship diagram is based on real world. It is a collection of entities (objects) and relationship among entities.
2. Entity relationship model has mainly three components:

(i) Entity

(ii) Attributes

(iii) Relationship



E-R Model

Fig.4.7 E-R Model

(i) **Entity:**

- Entity is a person, place or thing which can be identifies.
- It is represented by rectangle ▭
- It is of two types:

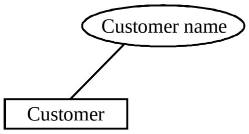
Weak entity: It depends upon some other entity. It is represented by ⧈

Strong entity: It is not dependent on some other entity. It is represente ▭

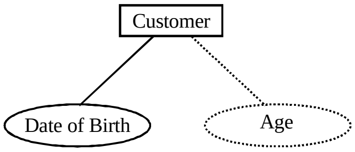
(ii) **Attributes:**

- Attributes are the properties of entity.
- It is represented by ellipse ⬭
- It is of four types:

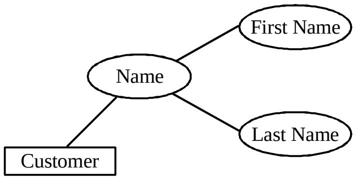
***Single attribute*:** These attributes cannot divided into subparts.



Derived attribute: The value of these attributes are derived from another attribute.



Multivalued attribute: These attributes have more than one value.

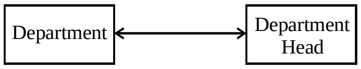


Composite attribute: The value of these attributes can be divided into another attributes.

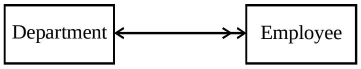
(iii) **Relationship:**

- It is used to connect entities. The entity involved in relationship is known as participants.
- The number of participants in a given relationship is known as degree of relationship.
- It is of four types:

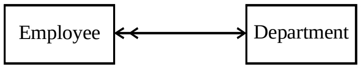
One to one relationship:In one to one relationship, for one record in entity A, there is exactly one record in entity B.



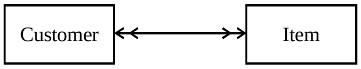
One to many relationship: In one to many relationship, for one record in entity A, there is more than one record in entity B.



Many to one relationship: In many to one relationship, for many records in entity A, there is only one record in entity B.



Many to many relationship: In many to many relationship, for many records in entity A, there are many records in entity B.



M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

SECTION- B

UNIT 5: SOFTWARE DESIGN

5.1 INTRODUCTION

Software design is a process of transforming the customer requirements into a form that is implementable by using a programming language. It is a very creative process. It is the basis of effective engineering. It should be practiced and learnt by experience and study of the already existing systems. The design phase begins when the requirements document is available. This is the first phase of transforming the problem into a solution. The design process is made up of a set of principles, concepts and practices which help a software engineer to model the system or product that is to be developed.

According to the Stevens in 1991, "Software Design is the process of inventing and selecting programs that meet the objectives for a software system."

In the words of Coad and Yourdon, "Software Design is the practice of taking a specification of externally observable behavior and adding details needed for actual computer system implementation, including human interaction, task management and data management detail."

5.1.1 **Outcomes of a Design Process**

The Design Process is an activity in which software requirements are studied and analyzed to produce a description of the internal structure and organization of the system. During the software design phase, many serious, strategic, technical decisions are taken to achieve the required functional and quality requirements of a system. These decisions help in developing the software in a successful manner and carry out its maintenance in a well-defined manner to improve the quality of the Final Product. Software design transforms the problem into a solution. In this phase the customer requirements, the business requirements and the technical needs, all come together to formulate a product or a system.

5.1.2 **Classification of Design Activities**

A good software design can never be achieved by using a single step procedure but requires several steps. Software Design is an activity in which software requirements are studied analyze and then a systematic procedure is adopted to produce a description of the internal structure and organization of the system that becomes the basis of its construction. Broadly speaking thus are two activities.

1. Software Architectural Design
2. Software Implementation Design

The meaning and scope of these two activities tend to vary considerably. In software architectural design, the top level structure and organization of the system is described and different components are identified. The system is decomposed and organized into components and interfaces between these components are described. The outcome of this architectural design is known as Program Structure or Software Architecture.

In the second activity i.e. the software implementation design, the data structure and the algorithms of different modules are designed and its outcome is usually known as Module Specification Document. In this activity each component is sufficiently described to allow for its coding. After studying different design approaches; seven steps are necessary to obtain a good design these steps are as follows:

(i) Function decomposition, (ii) Interface definition, (iii) Operational time line development, (iv) Data Definition, (v) Concurrency and real time consideration, (vi) Consolidation, (vii) Test procedure development.

5.1.3 **Classification of Design Methodologies**

Software Design is the central activity of software engineering. It is also the integrative activity at the core of software engineering. The designer should be methodical, disciplined, communicative and self-analytical. Various design methods are used today. The design principles and concepts lay a foundation for the creation of a Design Model that encircles around representation of Data, architecture, interface and components. The major software Design Methods include – (i) Function Oriented Design Methods, (ii) Data structure leased Design methods, (iii) Object oriented Design Methods, (iv) Reuse Based Design Methods.

Design Methodology is a very broader area and it is a problem solving activity. Design Methodology focuses on:

1. **Divergence:** It applies critical thinking through qualitative and quantitative approach to create better designs solutions.

2. **Transformation:** It includes better redefining of specifications of design solutions that leads to better guidelines for traditional and contemporary design activities.
3. **Convergence:** Prototypes probable scenarios for better design solutions.
4. **Sustainability:** It controls systematically the process of exploring, redefining and prototyping of designs solutions continually overtime.
5. **Articulation:** It articulates the visual relationship between the parts and the whole.

The true Goal of Design Methodology is to gain key insights to obtain more holistic solutions to achieve better experiences for users with products, services, environments and systems.

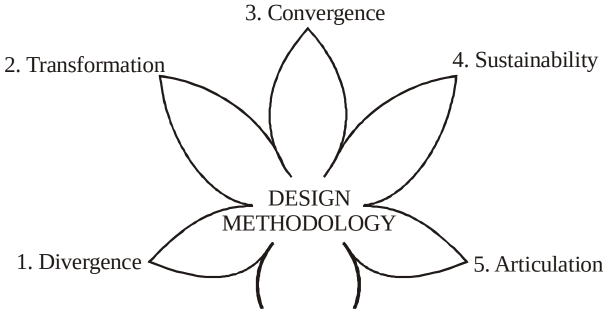


Fig. 5.1: Software Design Methodology

5.1.3.1 **Procedural Approach**

Procedural design techniques are very common and popular and at present they are being used in many software organizations. It is the result of focusing full orientation to the function of the program. This is an approach to software design where the design is decomposed into a set of interacting units where each unit of module performs a specific function Niklans Wirth who is the creator of Pascal and a number of other languages are one of the best known advocates of this method. His special variety to known as step-wise refinement. This stepwise refinement is a top-down strategy where a program is refined as a hierarchy of increasing levels of detail (Mills 1988). Refinement is actually a process of elaboration.

Procedural design is also named as component level design. It occurs after data, architectural and inter face designs have been developed. The aim is to translate the design model into an operational software.

5.1.3.2 **Object Oriented Approach**

Object-oriented design is a design strategy of planning a system of interacting objects for the purpose of solving a reaching a predefined software generated result. These Objects manage their own private state and offer services to other objects and hide representation of the state to limit access to it.

Object-oriented concepts

The five basic concepts of object-oriented design are:-

1. Object/Class
2. Information hiding
3. Inheritance
4. Interface (object-oriented programming)
5. Polymorphism

5.2 OBJECTIVE OF SOFTWARE DESIGN

Software Design is expected to deliver the requirement as specified in the Feasibility Report. The quality of the design is assessed with a series of technical reviews or design walk throughs. The main software objectives which a software design should provide are as follows:

Basic Objectives

1. To produce such models that can be analysed and evaluated to find whether they allow the various requirements to be fulfilled.
2. To study and evaluate different alternative solutions and trade-offs.
3. To plan the following development activities.

Major Objectives

1. To understand the requirements and to generate the solution to accomplish that requirements.
2. To transform the problem space of the implementation.
3. To solve complex problems by dividing them into a set of sub-problems and then finding partial solutions easily in a better way.
4. To identify the proper level of detail when design should stop and implementation should start.
5. To define the software architecture and describe if through step by step details.
6. To make the module functionally independent.
7. To implement all the explicit requirements of the Analysis Model.
8. To define the relationship between major structural element of the software.
9. To produce a design that is readable and understandable.
10. To provide a complete picture of the software that can be assessed for quality from an implementation prospective.
11. To translate the customs requirements into a well-furnished software product or system.
12. To follow a simple design approach so that designs are easily understood, easily built and easily tested.
13. To accommodate the changes that may be required during its life time.

5.3 CRITERIA OF EFFECTIVE SOFTWARE DESIGN

Most researchers and software engineers agree on certain characteristics that every Good Software Design must possess. The Goodness of a design depends on the targeted application. The definition of a Good Software Design can vary depending on the application on which it is based. The characteristics of a Good and effective Software Design are as under:

1. Correctness—A Good Design must implement correctly all the functionalities of the system.
2. A Good design should be easy to understand.
3. It should be efficient and should accommodate all the implicit requirements desired by the user.
4. It should not be difficult to change.
5. A good design should exhibit on architectural structure with recognizable design patterns.
6. An effective software design should be modular. It should be logically divided into elements that perform specific functions and sub functions.
7. A good design always possesses will written documents.
8. A design is said to be good and effective if it separates data, procedures and tuning considerations as far as possible.
9. In a good design, every requirement is testable. If a design cannot be tested easily against its requirements, then it is unacceptable.
10. Each and every point about the design should be easily communicated to all concerned through proper abstractions and representations.
11. A good design is liable to be repeated or re-used.

Each of the above characteristics is actually a goal of the design process. The design should be readable, understandable and must provide a complete picture of the software and if should fulfill the requirements desired by the customer.

5.4 DESIGN PRINCIPLES

Fundamental Principles for any discipline remain the same throughout. They provide the underline basis for development and evaluation of techniques. Fundamental concepts of software design include abstraction, structure, information hiding, modularity, concurrency, verification, testability, discreteness, and reusability and design aesthetics.

Some factors that help in making good software are creative skill, past experience and an overall commitment to quality. Davis has suggested a set of principles for software design which have been adapted and extended in the following list :

1. The Designer Process should not suffer from tunnel "Vision". A Good designer applies alternative approaches judging each according to the requirements of the problem and also takes into consideration the resources available to do the job.
2. The design should be traceable to the analysis model — A Good design must have means to track how the requirements have been satisfied.
3. The design should not re-invent the wheel — A set of design patterns should be chosen as an alternative to re invention. Design time should be used to represent truly new ideas and integrate all preexisting patterns.
4. The design should "minimize the intellectual distance" between the software and the problem as if exists in the real world.
5. The design should exhibit uniformity and integration.
6. The design should be structured to accommodate change.
7. The design should be structured to degrade gently, even when data, events or operating conditions are encountered. If should be designed to adjust according to the unusual circumstances.
8. Design is not coding and coding is not design — even when detailed procedural designs are produced, the level of abstraction of the design model is higher than source code design decisions at the coding level address the small implementation details which enables the procedural design to be coded.
9. The design should be assessed for quality as if being created not after the fact.
10. The design should be reviewed to minimize conceptual errors.

By following the design principles, Software Engineer produces a design of high quality which shows both external and internal quality factors. The user readily observes the external qualities (Speed, correctness, usability and reliability etc.). The internal quality factors are of immense significance to the Software Engineer. The designer must understand basic principles to create a high quality design from the technical perspective.

5.5 DESIGN STEPS

By critically examining and evaluating different design approaches, seven steps have been devised which should be followed to obtain a good design. These steps are as follows:

1. **Function Decomposition:** In the step software product is partitioned into smaller components. Rules are observed to assure that recombining and reassembling the components to intelligent testing scenarios which helps the management in maintaining control over the process. Block diagrams, structure charts and tools are the software development tools. Some characteristics and constraints for partitioning are as follows:

(a) Unit Testing Improvement
(b) System testing enhancement
(c) Cohesion and coupling
(d) Managerial Visibility
(e) Managerial control
(f) Information hiding
(g) Contractual an organizational compatibility
(h) Data, time dependency and function separation within limits.

The components are the building blocks. The partitioning step is the realization of a structure or architecture for the final design of the software product.

2. **Interface Definition:** Component to component and component to external interface are identified and defined. Many tools like object diagrams, data flow diagrams, timing chart and data dictionaries support this step. Another requirement of this step is to provide interface information that a designer, coder and tester needs for using other design components in developing a software product.

3. **Operational time line Development:** In this step realistic operational timeline or scenario to exercise are developed for each design step. These timelines should have operator interactions with the system.

4. **Data Definition:** This step is linked to develop data structures and this step serves the purpose of defining the files, records, fields, access methods, global and local data structure and the procedure to maintain files. Transactions are chalked out and the file size estimates are made. Tools for this step are data dictionary, information modeling and entity relation diagrams.

5. **Concurrency and Real Time Considerations:** The fifth step is linked to concurrence and real-time requirement. It is strongly desired to decouple timing requirements and their design and correctness tests from function, performance and data correctness. This step is used to establish the timing for the design component. The tools used for this step are timeline extensions to data flow diagrams and McCabe's analysis approach.

6. **Consolidation:** Information gathered from in steps one to five and a realization of the consolidated design information is used as a direction to coders in the form of diagrams, pseudo code structured English or any other form of program design languages.

7. **Test procedure Development:** This step studies those ways and means by which the design can be tested as a design component and finds how the design component contributes to confirm a system level acceptance criteria. This slip helps in verification that all requirements in the software requirements specifications have been met and traces if to implementation in the system design specifications.

Control hierarchy and data structure are too important aspects of the overall software products Architecture.

Control Hierarchy: The control relationship among various components of the product is sometime, referred to as visible. Each component has access to many other components and connectivity is done only to those components directly involved by the given component.

Data Structure: It is a representation of the relationship among data items in a software product design organization, access methods and controls, definitions, degree of association etc.

Data structure is consistent with overall architecture and if stays connected with control and procedure design throughout the design process. Formal design reviews can be inserted at proper interval to ensure that design efforts are fully coordinated.

5.6 DESIGN CONCEPTS

By following a rigorous approach in any engineering activity, satisfactory products can be produced. We can control. The costs of the products and increase their reliability. This does not mean to constrain creativity rather if improves the engineer's confidence in precision and accuracy. He brings creative results after critically analyzing in the light of a rigorous assessment.

Some basic steps have been devised over the past four decades which provides a basis to the software designer to apply more sophisticated design methods. He becomes capable of knowing different criteria to partition software into individual components; he comes to know how data structure detail can be separated from a conceptual representation of the software.

In the words of M.A. Jackson, "The beginning of Wisdom for a software engineer is to recognize the difference between getting a program to work and getting if right". The basic concepts for the Software Design are given below:

1. Abstraction

2. Refinement
3. Modularity
4. Software Architecture
5. Control Hierarchy
6. Structural Partitioning
7. Data Structure
8. Software Procedure
9. Information Hiding

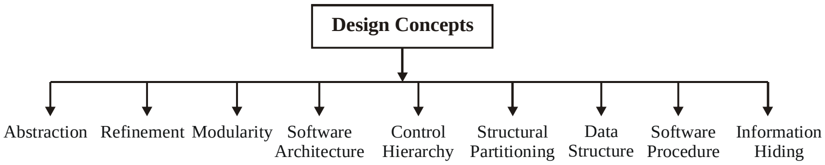


Fig. 5.2: Design Concepts

1. **Abstraction:** It is a very powerful concept and an intellectual tool used in all engineering disciplines which permits a designer to consider a component at an abstract level without worrying about the details of the implementation of the component. It is a special case of separation of concerns where in concern of the important aspect is separated from the concern of the unimportant details. Abstraction is an indispensable part of the design process and is essential for problem partitioning. Abstracts is used to existing components as well as components that are being designed. Abstraction of existing components plays a vital role in the maintenance of the system and we become able to modify the system in a step wise manner. Abstraction reduces the information content of a concept or an observable phenomenon, typically to retain relevant information for a particular purpose. Abstracting happiness to an emotional state of mind reduces the amount of information conveyed about the emotional state. There are different levels of abstractions. These levels are as follows:

(i) **Data Abstraction:** In Data Abstraction; the component is specified by the data types or data object and operations that performed on data object. Like any data object, the data abstraction for door would comprise of a set of characteristics (e.g. door type, swing direction, opening mechanism, weight, length, breadth, height etc.) Many modern programming languages provide mechanisms for creating abstract data types.

(ii) **Procedural Abstraction:** It is a named sequence of instructions that has a specific and limited function. In procedural abstraction component is specified by the function if performs e.g. Auto sum function in Excel. OR the word open for a door that implies a long series of procedural steps (i.e. walk to the door, reach out and grasp knob, turn knob end pull door, step away from moving door etc.)

(iii) **Control Abstraction:** It means a program control mechanism without specifying internal details. If describes the external behavior of that component. In example of a control abstraction is the Synchronization Semaphore used to co-ordinate activities in an operating system.

(iv) **Cluster Abstraction:** In cluster abstraction the component in a group of related classes that work together. These are also called framework e.g. for net frame work.

2. **Refinement:** Stepwise refinement is a top down strategy where in a program is refined in the form of hierarchy. In this technique decomposing of a system is done from high level specifications into more elementary levels. In each step, one part of high level description is taken and refined. It is actually a process of elaboration. Refinement is also known as “Stepwise Program Development and Successive Refinement” Niklans Wirth suggested following activities:

(i) Decomposing design decisions to elementary levels.
(ii) Isolating design aspects those are not truly independent.
(iii) Postponing decisions concerning representation details as long as possible.

(iv) Carefully demonstrating that each successive step in the refinement process is a faithful expansion of previous steps.

The following steps should be revised at each level in the design:

(a) Study and understand the problem.
(b) Identify gross features of at least one possible solution.
(c) Describe each abstraction used in the solution in some design description language.

Refinement begins with the specifications which are derived during requirements analysis and external design. It is a very effective method for describing small sized programs but fails in the large sized programs refinement helps the designer to know low-level details as design progresses.

3. **Modularity:** The real power of partitioning comes if a system is partitioned into modules, so that these modules are solvable and modifiable separately. It is a program unit. A module can be a macro, a function, a procedure a process or a package. Some criteria is followed to select modules so that modules support well defined abstractions e.g. coupling and cohesion are two modularization criteria used in Functional Abstraction.

Modularity promotes design clarity which cases implementation, debugging, testing, documenting and maintenance of the software product. A modular system has the following characteristics:

(i) Each Processing Abstraction is a well-defined sub system which can be effectively used in other applications.
(ii) Each function has a single well defined motto.
(iii) Each function manipulates only one major data structure.
(iv) Functions share global data selectively. It is convenient to identify all solutions that share a major data structure.
(v) Functions that control instances of Abstract Data types are encapsulated with the data structure being manipulates.

4. **Software Architecture:** Shaw and Garlan (1996) suggested that Software Architecture is the initial step in producing a software design. They gave three design levels namely Architecture Design, Code Design and Executable design.

Software Architecture is the hierarchal structure of Program Components (Modules) and expresses the way in which these components interact and structure of data used by the components.

Architecture Design: It links the system capabilities observed in the requirements specification with the system components that will implement them. The architecture describes the inter connections among different modules. He also defines operators who create systems from sub systems.

Code Design: If involves algorithms and data structures the components such as programming languages primitives such as numbers, characters, pointers and control threads. There as primitive operators which involves the language's arithmetic and data manipulation primitives and composition mechanisms (arrays, files, procedures)

Executable Design: The code design is studied of a lower level of detail. It discusses memory allocation, data formats, bit patterns and so on.

Keeping all the above mentioned specifications the architectural design can be represented using any number of the following models in the figure of system I and system II, it is to be noted how one component is connected to the other only when the first component can invoke the other. For a given component, the set of other components to which arrows are drawn is called scope of control of the components. The components invoked by the given component are collectively referred to a scope of effect e.g. in System I-II, are two possible designs for the same system. It should be noted that no component should be in the scope of effect; if it is not in the scope of control. If the scope of the effect is wider than the scope of its control, if is almost impossible to give assurance that a change to the component will not damage the whole design.

Fan in: It is the number of components controlling a particular component.

Fan out: It is the number of components controlled by a component. Component "A" has a fan-out of 3 in system I but a fan-out 5 in system II.

Fan in for component 'C' is 1 in both systems.

Generally a system with high fan-out tries to minimize the number of components. The controlling component performs more than one function. Comparing in this way system I with low fan out may be better than System II.

Studying from another angle, when number of levels are increased a particular component is used more than once. If we design one general purpose component to do that task, the results design is more efficient and easier to test. In such cases design with a large number of levels is considered better one of the goals in designing systems is creating such components which have high fan in and low fan-out.

(a) **Structural Models:** It puts forth Architecture as a systematized collection of program components.

(b) **Frame Work Models:** These models enhance the level of design abstraction. If tries to identify architectural design frameworks that appear in similar types of applications.

(c) **Dynamic Models:** They address the behavioral aspects by indicating how the system configuration may be changed as a function of external events.

(d) **Process Models:** They lay emphasis on the design of the business or the technical process.

(e) **Functional Models:** These models can be used to represent the functional hierarchy of a system.

5. **Control Hierarchy:** It represents the organization of program components and if shows a hierarchy of control. It does not represent procedural aspects of software. It does not apply to all architectural styles. Number of components under the control of a particular component can be represented diagrammatically as under.

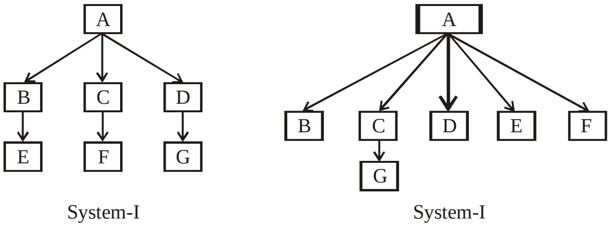


Fig. 5.3: One component controlling other components

6. **Structural Partitioning:** When the architectural style of a design follows a hierarchical nature, the program structure can be partitioned either horizontally or vertically.

Horizontal Partitioning: In this method the control modules (shown as shaded boxes) are used to co-ordinate communications between functions and execution of the functions. It has the following benefits:

- It results in software that is easier to best and maintain.
- It results propagation of fewer side effects.
- It results in software that is either to extend.

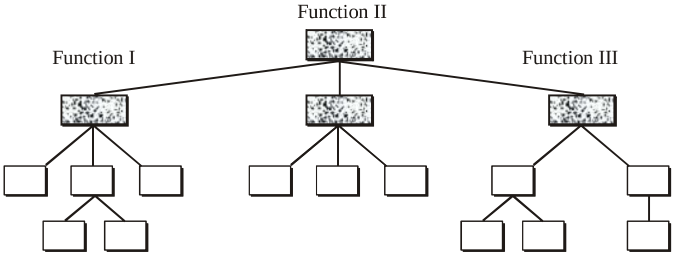


Fig. 5.4: Horizontal Partitioning

Disadvantage of Horizontal Partitioning: In this more data has to be passed across the module interface. This complicates the overall control flow of the problem, especially during speedy movement from one function to another.

Vertical Partitioning: This is also called factoring. In this the control (decision making) modules are located at the top and work is distributed in a top down manner in the program structure. Top level models perform control functions and do little processing. Modules that reside low in the structure perform all input, computation and output tasks.

The nature of change in the program structures requires vertical partitioning because a change in a control module (high in the structure) will have a higher probability of propagating side effects to modules that are subordinate to it. A change to a worker module (low level in the structure) is less likely to propagate side effects. Therefore vertically partitioned structures will be more maintainable a key quality factor.

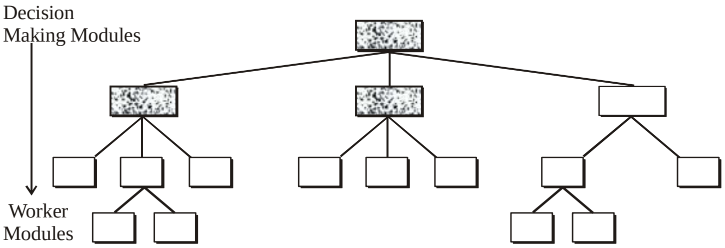


Fig. 5.5: Vertical Partitioning

7. **Data Structure:** Data structure is a representation of the logical relationship among individual elements of the data. As the structure of information has a great effect on the final procedural design. Data structure has an effect on the organization methods of access degree of associativity and processing alternatives for information. Although organization and complexity of a data structure depend on the thinking and skill of the designer, yet a limited number of classic data structures form the building blocks for more highly skilled and well-designed structures.

A scalar item represents a single element of information and it is the simplest form of data structure. Its access may be achieved by assigning a single address in the memory. The size and format of a scalar item are controlled by a programming language and they may very within limits. A scalar item may be a logical entity of one bit length, an integer or floating point number being 8 to 64 bits long or a character string that is hundreds or thousands of bytes long.

When scalar items are well organized in the form of a list or a group, sequential vector is formed. Vectors open the door to variable indexing of information.

When the sequential vector is extended to two, three and finally an arbitrary number of dimensions, an n-dimensional space is created. In many programming languages, an n-dimensional space is called an array. Two dimensional matrix is the commonest n-dimensional space.

Items, vectors and spaces may be organized in various formats & linked list is a data structure that organizes noncontiguous scalar items, vectors or spaces in a manner (called nodes) so that they may be processed as a list. Each node contains the appropriate data organization and one or more pointers that indicate the address in storage of the next node in the list. Nodes may be added at any point in the list. It is done by redefining pointers to accommodate the new list entry. Other data structure is constructed using the fundamental data structures. A hierarchical structure is commonly encountered in applications that require information categorization and associativity. Data structure can be represented at different levels of abstraction. For example, a stock is a conceptual model of a data structure which can be implemented as a vector or a linked list. The internal working of a stock depends on the level of design detail.

8. **Software Procedure:** Program structure defines control hierarchy without taking into consideration sequence of processing and decisions whereas software procedure focuses on the processing details of each module individually. Procedure must provide a precise specification of processing including sequence of events, exact decision points, repeatable operations and even data organization and

structure. There is a relationship between structure and procedure. A procedural representation of software is layered. The processing indicated for each module must include a reference to all modules subordinate to the module being described as is clear from the diagram given below:

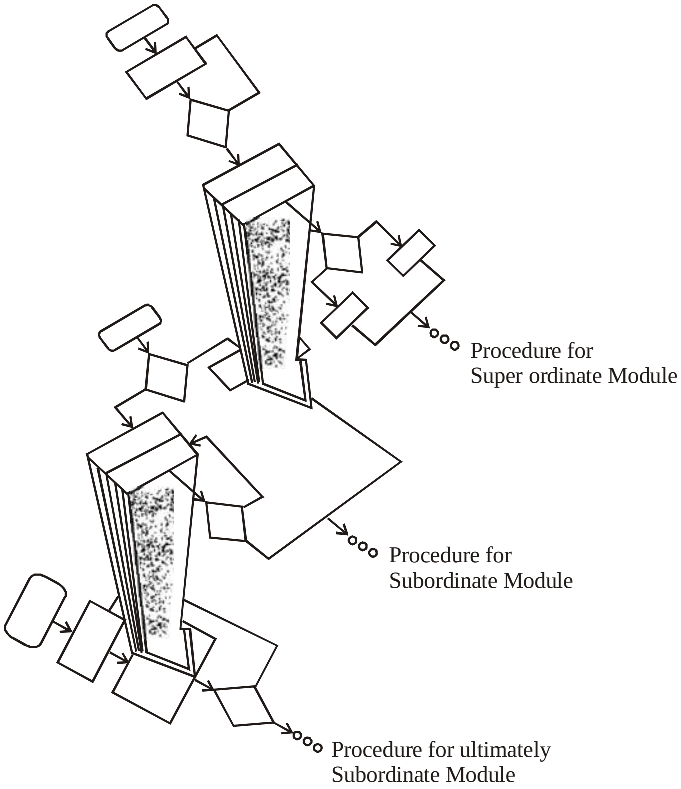


Fig. 5.6: Layered Software Procedure

9. **Information Hiding:** Modules should be designed and specified in such a way that each module in the system hides the internal details of its processing activities and modules may communicate through well-defined interfaces. The way of hiding in necessary details is known as Information Hiding. IEEE defines it as "the technique of encapsulating software design decisions in modules in such a way that the modules interfaces reveal as little as possible about the module's inner workings; this each module is a 'black box' to the other modules in the system."

According to Parnas, design should begin with a list of difficult design decisions and design decisions that are likely to change. Each module is designed to hide difficult and changeable design decisions because these design decisions transcend. Execution time, design modules may not correspond to processing steps in the implementation of the system.

Information Hiding is of much significance when modifications are required during the testing and maintenance phase. In object oriented design it gives rise to the concepts of encapsulation, modularity and is also linked to the concept of abstraction. Information hiding also hides the following information's:

(i) A data structure, its internal linkage and the implementation details of the procedure.
(ii) The format of control blocks such as those for queues in an operating system.
(iii) Character codes, ordering of character sets, shifting masking and other machine dependent details.

Advantages of Information Hiding

(i) It reduces the likelihood of adverse effects.
(ii) Leads to low coupling.
(iii) Attaches importance to communication through controlled interfaces.

(iv) Limits the global impact of local design decisions.

(v) Helps in producing higher quality software.

5.7 DESIGN QUALITY METRICS

Design quality metrics is a methodology of determining the quality of a design. A design is supposed to be good if it implements a specification correctly and allows efficient code to be produced. A good design is one which is maintainable and understandable. A maintainable design can be adapted to modify existing functions and add new functionality. The design components should be cohesive as they should have close logical relationship. They should be loosely coupled and should not be tightly integrated. Design quality metrics are used to assess the quality of design. Coupling & cohesion measures have been developed in conjunction with functional.

5.7.1 Cohesion

The internal cohesion of a module is measured in terms of the close of the relationship between its components or in other words as the strength of binding elements within the module. Cohesion occurs on the scale of weakest elements (least desirable) to strongest elements (most desirable). It shows how tightly the elements of the module are attached to one another. A component should implement a single logical function or a single logical entity. Contribution to implementation should be from parts of components. There is no need to modify many components if a change is desired. Cohesion gives the designer an idea about whether or not the different elements of a module belong together in the same module.

Constantive and Yourdon identify seven different levels of cohesion in order of increasing strength:

1. Coincidental
2. Logical
3. Temporal
4. Procedural
5. Communicational
6. Sequential
7. Functional

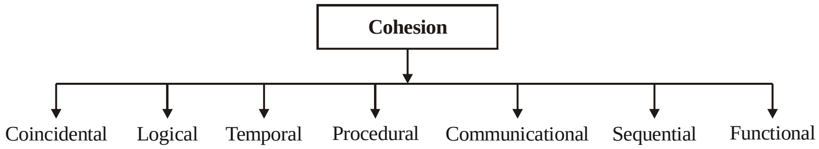


Fig. 5.7: Cohesion

1. **Coincidental:** If occurs when there is no meaningful relationship among the elements of a Module. Unrelated functions, processes or Data are present in the same component for the reasons of easiness serendipity. Coincidental cohesion can happen if an existing program is ‘modularized’ by cutting if into pieces and creating different pie modules. If a module is created to save duplicate code by combination some part of the code that occurs at many different places that module may have coincidental cohesion. In General, it is not wise to create a most merely to avoid duplicate code or chop a module to reduce the module size.

2. **Logical Cohesion:** A module is said to have logical cohesion if there is logical relationship between the elements of a module. Several logically related functions or data elements are kept in the same component. A typical example of this type of cohesion is a module that performs all the inputs or all the outputs. One component may read all types of input (from tape, disc or telecommunications port). It does not matter where the input is coming from or how it will be used. Although it is more reasonable than coincidental cohesion, the elements of a logically cohesive component are not related functionally. In general, logically cohesive modules should be avoided, if possible.

3. **Temporal Cohesion:** Temporal cohesion and logical cohesion are similar except that in temporal cohesion the elements a related in time and are executed together. Modules that perform activities like “Initialization” “clean up” and “Termination” a generally temporally bound. This avoids the problem of passing the flag and the code is generally simpler.

4. **Procedural Cohesion:** A procedurally cohesive module contains elements that belong to a common procedural unit. Sometimes it is needed that functions must be performed in a certain order e.g.

Data must be entered before they can be checked and then manipulated; all these three functions in a specific order. When functions are grouped together in a component to maintain a proper procedure, the component is procedurally cohesive. Such modules occur when modular structure is determined from same form of flow chart. Procedural cohesion, often, cuts across functional times.

5. **Communicational Cohesion:** In this type of cohesion, elements are related to each other by a reference to same input or output data because they operate on or produce the same data set. Sometime, unrelated data are fetched together because if can be done with only one disc or tape access. Communicably cohesive modules can do more than one function. e.g. “Print and Punch record”. It is to be noted that sometimes communicational cohesion may destroy the modularity and functional independence of the design.

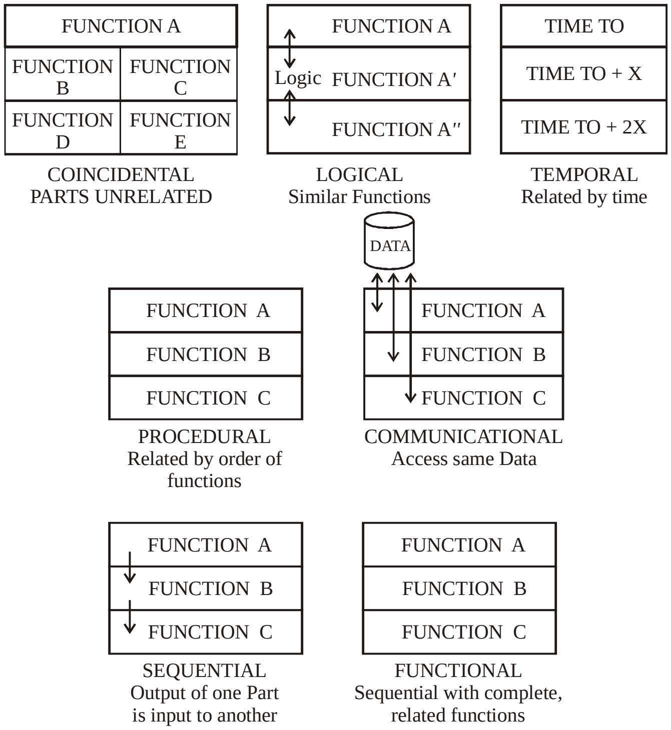


Fig. 5.8: Examples of Cohesion

6. **Sequential Cohesion:** When the output from one part of a component forms the input to the next part, we get sequential cohesion. Sequential cohesion does not provide any directions on how to combine them into modules. Different possibilities exist:

(a) Combine all in one module.
(b) Put the first half in one and the second half in another.
(c) The first third in one and the rest in the other and so on sequentially cohesive modules bear a close resemblance to the problem structure however they are still considered to be for from the ideal; which is function cohesion.

7. **Functional Cohesion:** Functional cohesion is the strongest cohesion. In this type of cohesion, every processing element is necessary for the performance of a single function and all the desired elements are contained in one component. A functionally cohesive component performs only and only that function for which it is designed and nothing else.

A cohesive object is one in which a single entity is represented and all the operations on that entity are included with the object. Such a cohesion called object cohesion is a new class of cohesion in which each operation provides functionality which permits the characteristics of the object to be modified, inspected or used as a basis for service provision.

5.7.2 **Coupling**

Coupling is the degree to which each program module relies on each one of the other modules. Coupling is a property of a collection of modules. It is an indication of the strength of the inter connections between the components in a design. Highly coupled systems have strong inter-connections with program units dependent on each other. Loosely coupled systems on the other hand are made up of components which are independent or almost independent. Loose coupling is desirable for good software engineering but tight coupling may be necessary for maximum performance. Uncoupled components have no inter connections at all coupling depends on several things:

(i) The references made from one component to another.
(ii) The amount of data passed from one component to another.
(iii) The amount of control one component has over the other.
(iv) The degree of complexity in the interface between components.

The following diagram showing component coupling explains everything. Dependency between modules has been shown by taking four modules A, B, C and D.

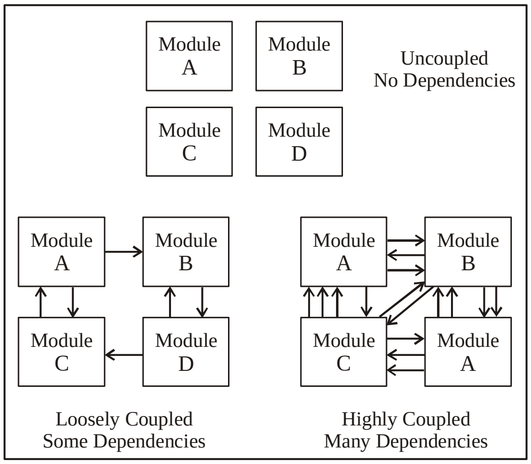


Fig. 5.9: Coupling

Coupling can be measured from complete dependence to complete independence. Actually a system can not be built of completely uncoupled components. When coupling is high, large parts of the system may be perturbed by the change. Low coupling helps in minimizing the number of components requiring revision.

5.7.2.1 **Types of Coupling**

Some types of coupling, in order of highest to lowest coupling are as follows: 1. Content coupling, 2. Common coupling, 3. External coupling, 4. control coupling, 5. Stamp coupling, 6. Data coupling message coupling

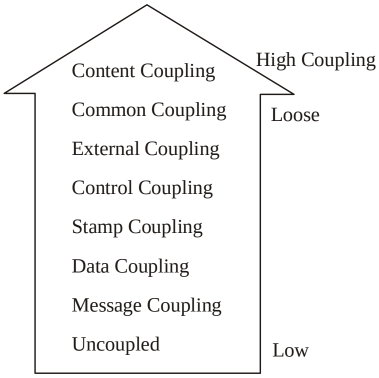


Fig. 5.10: The Range of Coupling Measures

1. **Data Coupling:** Two modules are data coupled, if they communicate by using an elementary data item that is passed as a parameter between the two, e.g. an integer, a float, a character etc. Data coupling is simpler and leaves less room for error. It is most convenient to trace data and to make changes. If coupling must exist between components data coupling is the most desirable.

2. **Message Coupling:** This is the loosest type of coupling. Modules do not depend on each other; instead they use a public interface to exchange parameter less messages.

3. **Stamp Coupling:** Stamp coupling is when modules share a composite data structure and use only a part of it, possibly a different part (e.g. whole record is passed to a function which needs one field of it).

4. **Control Coupling:** Two modules are control coupled if they pass a piece of information intended to control the internal logic using at least one ‘control flag’. The control flag is a variable that controls decisions in subordinate or superior modules. The advantage of control coupling is that each component controls only one function or executes one process. This restriction is necessary to minimize the amount of controlling information which must be passed from one component to another and localizes control to a fixed and identifiable set of parameters forming a well-defined interface.

5. **External Coupling:** External coupling occurs when two modules share an externally forced data format, communication protocol or device interface.

6. **Common Coupling:** Two modules are common coupled if they both are the same global data area. Changing the shared resource implies changing all the modules using it. The amount of coupling can be reduced by organizing the design so that data are accessible from a common data store but dependence is skill there because making a change to the common data means tracing back to all components that access the data to calculate the effect of that change. This kind of coupling is known as common coupling. With common coupling it becomes difficult to find which component is responsible for having set a variable to a particular value.

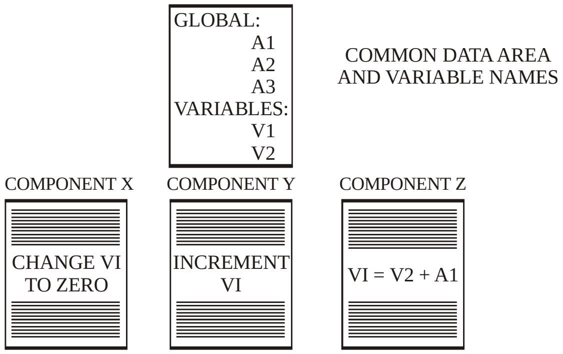


Fig. 5.11: Example of Common Coupling

7. **Content Coupling:** Content coupling exists between two modules if these module changes a statement in another module, one module references or changes data contained inside another module, or one module branches into another module. One module actually modifies the procedural contents of another module. In content coupling, modules are inter dependent on each other.

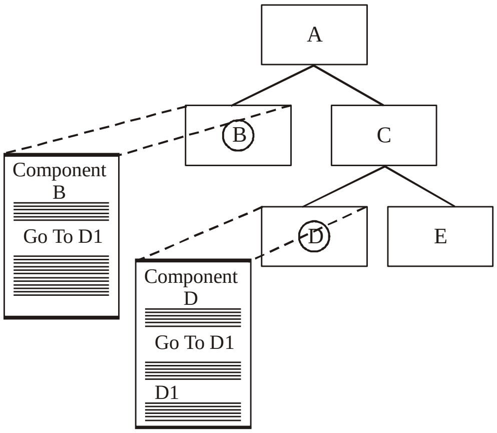


Fig. 5.12: Example of Content Coupling

5.8 DATA DESIGN

Data design is the first of all the activities in a software design process. It leads to a better program structure, effective modularity and decreased complexity. Data Design is developed by the transformation of the Data dictionary and entity relationship diagram into Data Structure which are necessary to implement the software. The whole process includes identifying the data defining specific data types and storage mechanisms and guarantees data integrity by using business rules and other mechanisms.

Data design creates a model of data information that is represented at a high level of abstraction i.e. the customer or user's views of data. The data model is refined into progressively more implementation specific representations which can be easily processed by the computer based system. In many software applications, the architecture of the data will put great influence on the architecture of the software that must process it.

The selection process may take the help of algorithmic analysis of alternative structures in order to determine the most efficient design or it can involve the use of a set of modules that provides operations on some representation of objects. Some principles are observed while specifying the data are given below:

1. The Systematic analysis principle applied to function and behavior should also be applied to data. Lot of June and energy is spent on deriving, reviewing and specifying functional requirements and preliminary design. In addition to it representations of data flow and content should be developed and reviewed; data objects should be identified, alternative data organizations should be considered and the impact of data modeling on software design should be evaluated.
2. All data structures and the operation to be performed on each should be identified. The design of an efficient data structure depends on operations to be performed on data structure upon evaluation of the operations; an abstract data type is defined for use in subsequent software design which simplifies software design considerably.
3. A data dictionary should be established and used to define both data and program design. The specific relationship among data objects and the constraints on the elements of a date structure can be more easily defined by dictionary like data specifications.
4. Low level data design decisions should be deferred until late in the design process. Overall data organization can be defined during requirements analysis, refined during data design work and specified fully during component level design.\
5. The representation of the data structure should be known only to those modules that must make direct use of the data contained within the structure. The quality of a software design is studied by the concept of information hiding and concept of coupling.
6. A library of useful data structures and the operations that may be applied to them should be developed which can reduce both specification and design effort for data.

7. A software design and programming language should support the specification and realization of abstract data types.

There are three levels to view the structure of data:

1. Program component level 2. Application Level 3. Business Level

At the program component level, design of data structures and the algorithms needed to manipulate them are essential for the development of a high quality design.

At the application level, the translation of a data model into a data base is necessary to get the specified business objectives. At the business level, the collection of information stored in different data bases should be reorganized into data ware house which enables data mining that has an effect on the business.

5.9 ARCHITECTURAL DESIGN

An architectural design acts as a preliminary “blue print” from which software is developed. Architecture describes the software top-level structure which identifies its components. IEEE architectural design as ‘the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system.’

First of all software requirement is examined to establish a framework and then building a physical model using recognized software engineering methods. Shaw and Garlan (1996) suggest that software architecture is also the first step in producing a software design levels; architecture, code design and executable design.

It is very useful to work from the top-down designing an architecture, then the code design and finally the executable design but Krasner, Curtis and Iscoe (1987) have studied the habit of developers on 19 projects; they present a report and on the basis of other evidences confirm that actually the designers more back and forth from one level to other as they get extra knowledge about the solution and its implication. As the designers explore other aspects of the system, the designers may interact with testers or programmers bringing changes in the design to step up implementation, testability or maintainability.

In Architectural design performs the following functions:

1. The software designers can specify the system behavior (function and performance) after the architectural design provides a level of abstraction.
2. It is really the consciences for a system, a good architectural design guides the process of system enhancement indicating what changes can be made without compromising the integrity of the system.
3. It evaluates all top level designs.
4. It develops and documents top level design for the external and internal interfaces.
5. It develops preliminary versions of user documentation.
6. It defines and documents preliminary test requirements and the schedule for software integration.

5.10 PROCEDURAL DESIGN (FUNCTION ORIENTED DESIGN)

Procedural design techniques are very common and popular and at present they are being used in many software organizations. It is the result of focusing full orientation to the function of the program. This is an approach to software design where the design is decomposed into a set of interacting units where each unit of module performs a specific function Niklans Wirth who is the creator of Pascal and a number of other languages are one of the best known advocates of this method. His special variety to known as step-wise refinement. This stepwise refinement is a top-down strategy where a program is refined as a hierarchy of increasing levels of detail (Mills 1988). Refinement is actually a process of elaboration.

Procedural design is also named as component level design. It occurs after data, architectural and inter face designs have been developed. The aim is to translate the design model into an operational software. The translation from high level of abstraction of existing design into low level of abstraction of operational program is risky because if can bring in subtle errors that are difficult to locate and correct in later shapes of the software process. During this translation, certain design principles must be followed. Regardless of the methodology adopted to represent the procedural design, the date structures, interfaces and algorithms defined should follow to a variety of well-established procedural guiding principles to avoid errors as the procedural design is developed.

1. **Structured Programming:** In the late 1960s, Dijkstra and others gave the idea of using a set of constructs from which any program could be formed. Each construct had a predictable logical structure which was entered at the top and exited at the bottom for the convenience of the reader to follow procedural flow move easily. These constructs are Sequence, Condition and Repetition. The sequence implements processing steps, condition provides facilities for selected processing and Repetition allows for looping. These structured constructs are meant to limit the Procedural Design to a small number of pre-determined operations this contributes more to a human understanding the process called chunking by the psychologists. Any program needs three constructs only for designing and implementation.

5.11 **STRUCTURED DESIGN METHODOLOGY (SDM)**

Structured Design Methodology helps the designer during process. Software is a transformation function that converts given inputs to desired outputs. The main objective of this strategy is to specify functional modules and connections. SDM controls and puts an influence on the final design so that the programs implementing design would have a nice hierarchical structure that has functionally cohesive modules and few inter connections between modules as for as possible. SDM aims to achieve high cohesion but low coupling.

Steps: SDM follows the following procedure:

1. **Restate the problem as Data Flow Diagram (DFD):** The Designer finds full freedom in creating a DFD because the system does not exist yet. DFD will solve the problems stated in SRS.

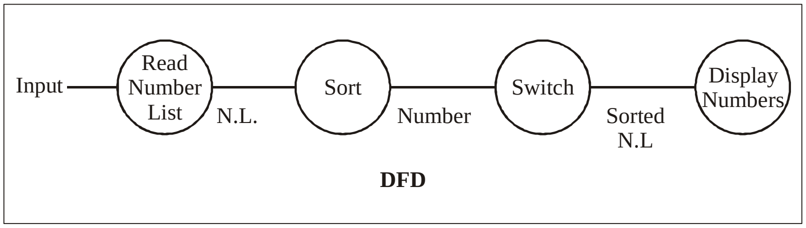


Fig. 5.13: Identification of DFD

Let us take the example of simple sort program that sort the 'n' numbers in ascending order. The problem has only one input file i.e. Read Number List. If we want to transform the input to the desired output, all the numbers are stored in a list. Then all numbers list goes to the sort transformation. If numbers are not sorted, Switch Transform changes the place order of number. Sorted Number are then passed to display transformation for printing numbers in ascending order on the screen.

2. **Identify most Abstract Input (MAI) and Most Abstract Out (MAO) data elements:** White going from outputs to inputs, the output of the major transformation is known as MAO and the input of major transformation is known as MAI. In this example of sort program, Number is MAI to switch transform and sorted N.L. is MAC.

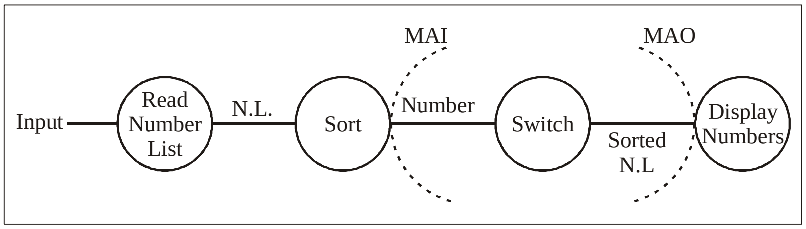
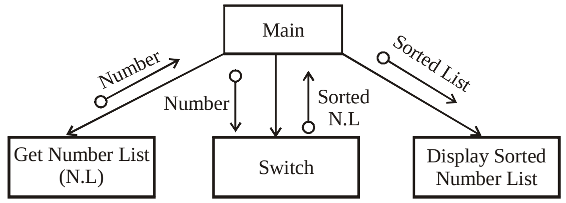


Fig. 5.14: Identification of MAI & MAO

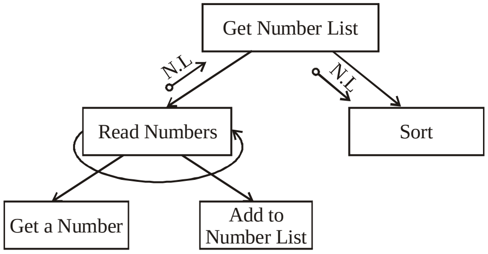
3. **First Level Factoring:** After the identification of central transformation MAI and MAO; the next step is the analysis of these information's. Main Module is specified first of all whose aim is to invoke the subordinates. For each MAI data item, an immediate subordinate module to the main module is specified. Each module is an input module which aims to provide to the main module the most abstract data item for which it is created. Same procedure is adopted for MAO data item, a subordinate module that is an output module that accepts data from the main module is specified.



First Level Factoring

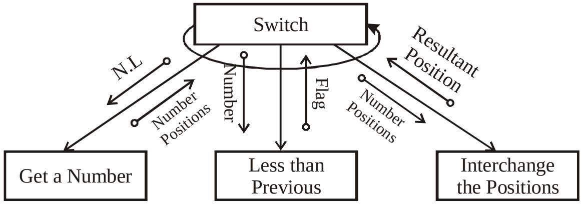
4. **Factoring the input, output and transform branches:** Each module in DFD is factored to move details:

(a) **Factoring the input Module:** The objective of input module is to produce some data to the main module. Do factor as input module, the transform in the DFD that produces number list is treated as a central transform as is shown in the following diagram.



(b) **Factoring the output Module:** It resembles the factoring of input module. The aim is to reach the final output. During the factoring of output module, usually there will be no input module because this task is performed by MAI.

(c) **Factoring the Central Module:** To factor a central module into its subordinate modules; there are no standard guidelines available. It depends on the designer's experience:



Graphical Design Notation

In this a flow chart is quite simple in the form of pictures.

A box indicates processing steps, a diamond represents a logical condition and arrows show the flow of control. Condition (if then else) is represented by decision diamond – if true – it causes 'then part processing' and if false – it invokes 'else part processing'.

Repetition is depicted by two slightly different forms. The do-while tests the condition and executes a loop task again and again as long as the condition holds true. A repeat until executes the loop task first and then tests a condition and performs the task again until the condition fails. The selection construct is the extension if then else successive decisions are taken to test a parameter until a true condition occurs and a case part processing path is executed. The structured constructs may be nested within one another. By nesting constructs in this way, a complex logical scheme may be developed.

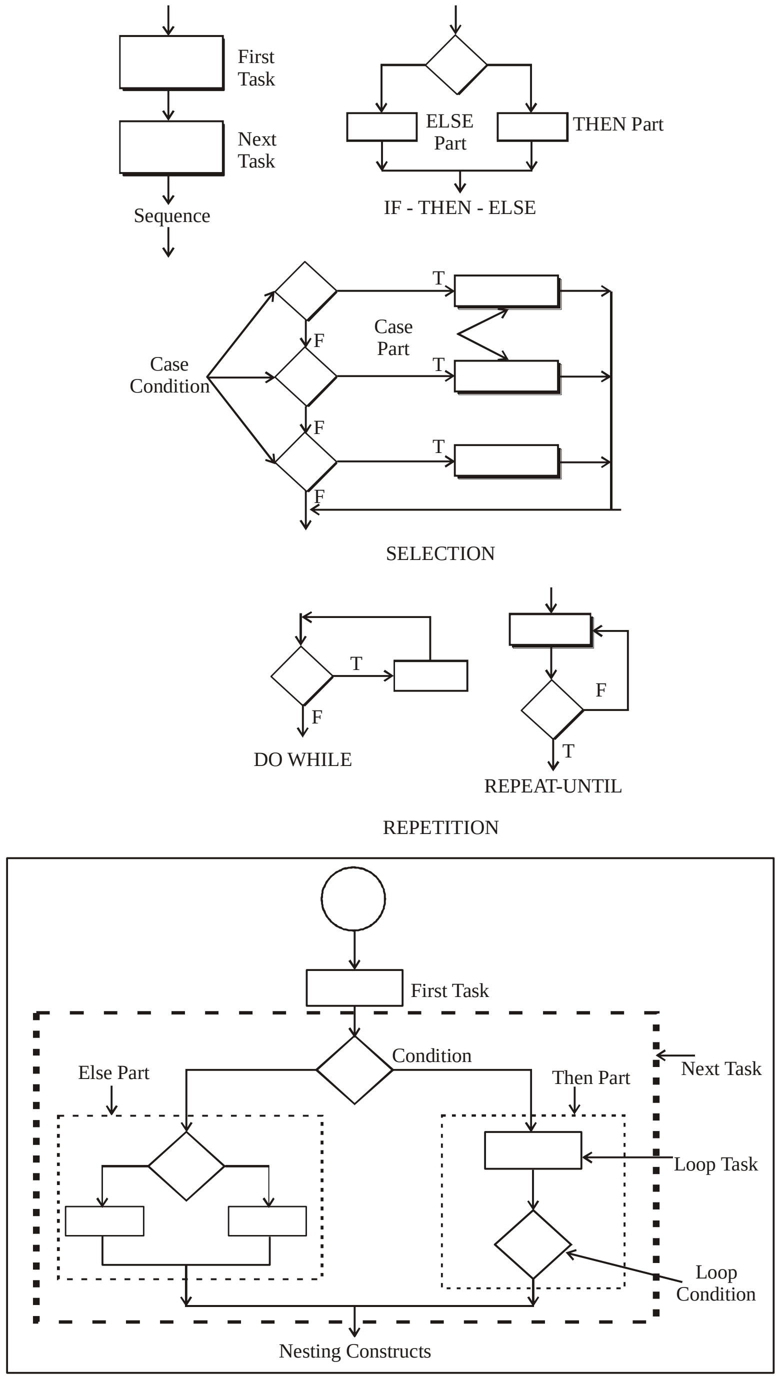


Fig. 5.15: Nesting of Different Structured Constructs

Tabular Design Notation: A module may be required to evaluate a complex combination of conditions and select proper actions based on these conditions. Decision tables provide criteria that translate actions and conditions in a tabular form. The table is not easy to misinterpret and if is possible to be used as a machine readable input to a table driven algorithm. Decision table is divided into four sections. As is clear from the following diagram; the upper left quadrant contains list of all conditions and the lower left hand quadrant contains a list of all actions that are possible depending on combinations of conditions. The right hand quadrants indicate condition combinations and the related actions that will happen for a specific combination.

CONDITIONS	1	2	3	4					n
Conditions # 1	✓			✓	✓				
Conditions # 2		✓		✓					
Conditions # 3			✓		✓				
ACTIONS									
Action # 1	✓			✓	✓				
Action # 2		✓		✓					
Action # 3			✓	✓					
Action # 4			✓	✓	✓				

Fig. 5.16: Decision Table Nomenclature

3. **Program Design Language (PDL):** PDL is also known as structured English or pseudo code. This is a language that uses the vocabulary of a natural language (English) and the overall syntax of a programming language. IEE defines PDL as a specification language with special const and sometimes with special constructs and sometimes verification protocols used to develop, analyze and document a program design. A team of designers and programmers use PDL as a way to ensure that actual programming is likely to match design requirements PDL is mainly used during the detailed design phase and uses to describe algorithms and specify interfaces between modules P&L enables the programmer to concentrate on the algorithms without worrying about all the syntactic details of a particular programming language.

Advantages of using PDL

- It uses English like statements that describe specific operator very precisely.
- It supports the idea of iterative refinement.
- It is expressed in easy-to-read format.
- Reviews became very easy because the source code is no examined.
- The language uses flow chart replacements, program documentation and technical communication at all level.
- Continuous use and refinement of the PDL has established if as the medium of choice for either creating refining a detailed program.

5.12 DESIGN VERIFICATION

Software design verification or review is necessary to review or check the design, implementation way, testing and maintenance, members of Development team. These reviews are well documented, comprehensive and systematic evaluation of a design to ensure the adequacy of the design requirements. It is done to evaluate the capability of the design and to identity problems IEEE defines if as a formal meeting at which a system's preliminary or detailed design is presented to the user customer or other parties for comment and approval.

The main aim behind the design review is to detect the errors in the design process which will reflect in code and also in the final system. These verifications are done to check if the design is of good quality or not. These reviews are taken at the end of the design phase to find solutions and resolve issues regarding software related design decisions, architectural design and detailed design. The reviews include study of the development plans, requirements and design specifications, testing plans and procedures, verification results for each stage of the design.

Types of Reviews: The review step generally involves three steps:

1. Preliminary Design Review
2. Critical Design Review

3. Program Design Review

1. **Preliminary Design Review:** It is a formal inspection of the high level architectural design to satisfy whether the design satisfies the functional as well as non-functional requirements of the customers and users. The review team includes customers, moderator, secretary, system designers, other state holders not involved in the project. The main aim is to:

(i) To specify whether effective modularity is achieved or not.
(ii) To define interfaces for modules and external system elements.
(iii) To ensure maintain ability has been considered or not.
(iv) To assess quality factors.
(v) To check discrepancies if any and to resolve it by the review team.

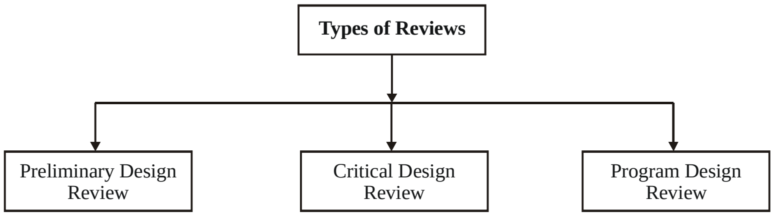


Fig. 5.17: Types of Reviews

2. **Critical Design Review:** Critical Design Review is taken after the successful completion of the preliminary design review. After the customer is satisfied, a critical design review is conducted with the following purpose in mind:

(i) To ensure that designs (conceptual and technical) are without defects.
(ii) To determine whether the design satisfies the requirements specifications or not.
(iii) To assess the functionality and maturity of the final design.
(iv) To justify that the design has clarity, effectiveness and easy to understand. A review is formed to check the design specifications using diagrams and data. Other individuals in addition to the members of the review team are:

(a) System tester: A technical expert.
(b) Analyst: Responsible for writing system documentation.
(c) Program Designer: Who understands the design in detail.

If a minor fault is found, if is resolved by the review team. In case of a major fault, the review team may take decision to revise the proposed technical design.

3. **Program Design Review:** The designers and the developers conduct this review after the completion of Critical Design Review. This is done to get feedback on the designs before implementation begins. The main purpose is to:

(i) Ensure the feasibility of the design.
(ii) Ensure that the interface is consistent with the architectural design.
(iii) Specify whether the design is amenable to implementation language.
(iv) Ensure that structured programming constructs are used throughout Software Design Process.
(v) Ensure that members of the implementation are able to understand the proposed design.

The team of Program Design Review includes program designers, developers, a system tester, moderator, secretary and analyst. A successful Program Design Review presents all plans related to coding plans before coding begins.

M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 6: CODING

6.1 INTRODUCTION

The software coding deals with design of data structure, information hiding, implementation. It is that phase of software development where actual implementation is carried out. The coding is done in a team manner which requires standardization of coding style, coding guidelines and coding documentations. The information hiding is another important aspect which is handled in this phase. In this chapter, we will go though the different coding styles and guidelines.

6.2 CODING

1. After the completion of design phase, coding is undertaken. The objective of the coding phase is to transform the design of a system into high level language code and then to perform unit testing at this code.
2. For adequate and effective coding, programmers have to mandatorily follow the coding standards. Before the testing phase, coding standards should be verified.
3. Coding guidelines provide some general suggestions regarding the coding style to be followed.
4. After the coding of a module, code reviews are carried out. This code review ensures that coding standards are followed during coding of a module. This code review also detects as many errors possible before testing.
5. It is important that the code should be written in such a manner that it could be modified later, if the need be.

6.3 CODING STANDARDS

A coding standard is a list of several rules to be followed during coding like variables should be named, name of module, exception handling mechanism etc. Many organizations develop their own coding standards and guidelines according to their need and type of software product they are going to develop. The some commonly known coding standards are as follows:

1. The organizations should follow the exact format of the header information which is as follows:

- Module's Name
- Author's Name
- Module's Creation Date
- Module's Synopsis

- Input/output parameters with different functions

2. There are some rules which limits the use of global. These rules explains what type of data can be declared global and what type of data cannot declared global.
3. The method of reporting error conditions and exception conditions should be standardized within an organization.
4. Global variable’s name should always start with a capital letter whereas local variable’s name should always start with small letter.

6.4 CODING GUIDELINES

Coding guidelines provides some general suggestions regarding the coding style to be followed. Some commonly known coding guidelines are as follows:

1. The function should be small not lengthy because length function is difficult to understand and arises large number of bugs.
2. goto statements should be avoided because it makes program unstructured.
3. The code should be well documented. It is only possible by using comment lines which explain it appropriately.
4. The code should be easy to understand. The complex coding is difficult to understand and maintain.
5. Avoid to use same identifies for multiple purposes. Same identifier for multiple purposes makes future enhancements very difficult and lead to confusion.
6. Variable names should be clear and simple. Variable names should be mnemonic procedure and function name should also be mnemonic.

6.5 PRINCIPLES OF PROGRAMMING/CODING

Programming is the process of writing, testing, debugging and maintaining of a computer program. The following are some basic principles of good programming practice.

1. Before start the programming, developer should understand the problem. He should choose a programming language which meets the needs of the software to be built and the environment in which it will operate.
2. Developer should create a visual layout that aids code understanding.
3. Avoid repetition in programming.
4. Coding should be simple because it takes less time to write, easier to modify and has fewer bugs.
5. Coding should improves code reliability and decrease development time
6. Developer should create a set of unit tests that will be applied once the coding is complete.Perform unit tests and correct the uncovered errors.
7. Developer should conduct a code walkthrough and refactor the code.
8. Developer should write the code for maintainer because any code that is worth writing is worth maintaining in the future, either by him(developer) or by someone else.

6.6 CODE REVIEW/VERIFICATION

1. After the coding of a module, code reviews are carried out. This code review ensures that the coding standards are followed during coding of a module. This code review also detects as many errors possible before testing.
2. After the successful compilation of module, code review takes place. Code review is a cost effective strategy which produce high quality code by eliminating coding errors.

3. Eliminating an error from code involves three main activities—testing, debugging and then correcting errors. Debugging is very laborious and time consuming activity than testing.

4. There are four types of code review which are as follows:

(i) Code Walk through

(ii) Code Inspection

(iii) Clean Room Testing

(iv) Static Analysis

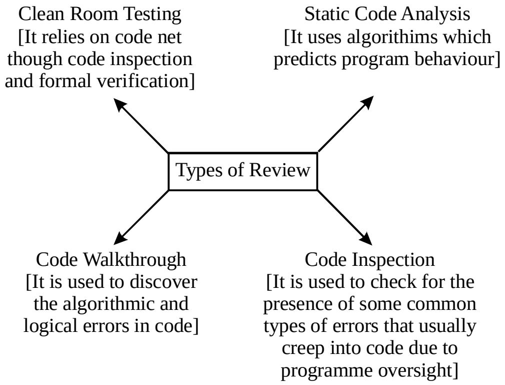

Clean Room Testing
[It relies on code net though code inspection and formal verification]
Static Code Analysis
[It uses algorithims which predicts program behaviour]
Types of Review
Code Walkthrough
[It is used to discover the algorithmic and logical errors in code]
Code Inspection
[It is used to check for the presence of some common types of errors that usually creep into code due to programme oversight]

Fig. 6.1 Types of Code Review/Verification

1. **Code Walkthrough**

- Code walkthrough is an informal code analysis technique.
- Its aim is to discover the logical and algorithmic errors in the code.
- It does not tell how to fix the discovered errors.
- Before the code walkthrough meeting, the members note down their findings to discuss.
- The success of code walkthrough are based on personal experience, common sense and several subjective factors.
- The walkthrough team should consist of three to seven members. The walkthrough team should not be too big or too small.

2. **Code Inspection**

- Code inspection is a formal code analysis technique.
- The aim of the code inspection is to discover common types of errors which are caused due to improper programming.
- Code inspection detects the commonly made errors along with the coding standards.
- It is also used to check whether coding standards are followed during the coding or not.
- Code inspection checks some programming errors like incompatible assignments, improper storage allocation and de-allocation, improper modification of loop variables, non-terminating loops etc.

3. **Clean Room Testing**

- It relies on code walkthrough code inspection and found verification.
- The aim of clean room testing is to produce documentation and code which is more maintainable and reliable.
- Clean room testing involves walkthroughs, inspection and formal verification which is time consuming and hence testing effort is increased.

4. **Static Analysis/Static Code Analysis**

- . It is also known as automated code review.
- Static code analysis is Excellent for enforcing compliance to standards

- It Identifies bugs in code and the design and implementation problems
- It uses algorithms which predict program behaviour.
- Static analysis explain the purpose of possible executions of a program. It gives assurance about any execution but this tool spend a lot of effort dealing with developer.
- Static analysis can be performed on modules or unfinished code

M.Sc. (Computer Science)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 7: SOFTWARE TESTING

7.1 INTRODUCTION

Software testing is an activity that has significant impact on the software development process. It is carefully designed process and must be performed effectively. Since allocated resource is limited and time is another constraint, so it is not possible to test any system with all possible inputs. It is the primary quality-control measure used during software development. The testing provides methods and techniques to test a system in reasonable time with assurance to give quality performance. The testing is broadly categorized as Black Box testing and White Testing. There is another way to cauterize the testing method called level testing discussed in detail in this chapter. The chapter includes detail discussion on testing along with advantages and limitations.

7.2 SOFTWARE TESTING

1. Testing is an important and critical activity of software development life cycle. It is the primary quality control measurement activity used during software development.
2. Testing is a set of systematic activities that can be planned in advance and conducted with the intent of finding errors. It make us feel confident that the software works well**.**

3. The goal of testing is to uncover requirement, design, and coding errors in the program. It is critical, expensive and time-consuming activity.
4. The testing process requires proper planning. The testing cannot show the absence of defects. It can show only software errors present.
5. During the testing phase, emphasis should be on the following:
   - Tests should be planned long before testing begins.
   - All tests should be traceable to customer requirements.
   - Tracing should begin “in the small” and progress toward testing “in the large.”
   - Testing must be conducted by the independent third parties.

7.2.1 **Objectives of Software Testing**

The testing objective is to test the code with aim to discover errors. It also expresses that the software functions are working according to software requirements specification with regard to functionality, features, facilities and performance. It is here noticeable, that testing will detect errors in the written code but it will not give you an idea about an error if the code does not address a specific requirement set in the SRS.

The primary objectives of testing are:

1. Testing is a process of executing a program with the intent of finding an error.
2. A good test case is one that has a high probability of finding the undiscovered error.
3. A successful test is one that uncovers an as-yet-undiscovered error.
4. It makes us feel confident that the software works well.

7.2.2 **Principles of Software Testing**

The testing principle is a guide book which helps software engineer to design effective test cases. There are many principles that a software engineer must understand before starting the testing process.

The following are the main principles of testing:

Sr. No.	Principles	Description
1	**All tests should be traceable to customer requirements**	It states that any defects that might cause the program or system to fail to meet the client’s requirements must be uncovered.
2	**The Pareto principle applies to software testing.**	It is also known as 80/20 principle. It states that 20% of the problems lead to 80% of other problems. One should concentrate to 20% of the problems and 80% of the problems will automatically remove.
3	**Tests should be planned long before testing begins.**	According to principle testing phase should be started soon after the requirements model is completed. The detailed test cases can begin as soon as the design model is designed.
4	**Testing should begin “in the small” and progress toward testing “in the large.”**	It states that the first test may be focused on individual components. As testing progresses, focus shifts in an attempt to find errors in integrated components and later on full system.
5	**Testing should be conducted by an independent third party**	The principle states that the testing should be conducted by an independent third party. The software engineer who has developed the system is not the best person to conduct testing
6	**Exhaustive testing is not possible.**	It states that It is possible, however, to adequately cover program logic and to ensure that all conditions in the component-level design have been exercised for testing.

Table 7.1 Principles of Software Testing

7.3 BASIC CONCEPTS

In software testing, there are few terms which generally create confusion. These terms are explained as follows:

1. ***Errors:*** *An error is a discrepancy between the actual value of the output given by the software and the specified (correct) value of the output for that given input.*

Errors occur when any aspect of software product is incomplete or incorrect. It is a mistake committed by the development team during development process. Errors can be classified into following categories:

- ***Implementation Errors:*** These errors made in translating design specifications into source code.
- ***Requirement Errors:*** These errors are due to incomplete and incorrect statements of user requirements. User fails to specify functional and nonfunctional requirements.
- ***Design Errors:*** These are introduced by failure to translate the requirements into complete and correct structures.
- ***Syntax Error:*** A syntax error is a program statement that violates one or more rules of the language in which it is written.
- ***Logic Error:*** A logic error deals with incorrect data fields, out-of-range terms and invalid combinations.

2. ***Fault:*** *A fault is a condition that causes a system to fail in performing its required function. A fault is the basic reason for software malfunction.*

It is also commonly called a bug. Sometime, it is possible that even providing correct input system give wrong results which due to some bug in the system. When it fails then we say that the system has a fault or a bug and it needs repair.

3**.** ***Failure:*** *Failure is the inability of the software to perform a required function to its specification.*

In other words, when software goes ahead in processing without showing error or fault even though certain input and process specification are violated, then it is called a software failure.

4. ***Quality Assurance:*** Quality assurance is the planned and systematic approach. It ensures that the standards, processes and methods are established and followed throughout the software development. Quality assurance is a set of activities which ensure the implementation of processes, procedures and standards in context to verification of developed software and intended requirements.

5. ***Quality Control:*** Quality control is used to verify the quality of the output. Quality control is set of activities which ensure the verification of developed software with respect to documented (or not in some cases) requirements.

6. ***Testing:*** Testing is the sequence of activities which ensure the identification of bugs in the software.

Difference between Testing, Quality Assurance and Quality Control

Quality Assurance	Quality Control	Testing
Quality assurance focuses on processes and procedures rather than conducting actual testing on the system.	Quality control focuses on actual testing by executing software with intend to identify bug/defect through implementation of procedures and process.	Testing focuses on actual testing of the system or unit with suitable test
Quality assurance is process oriented activities.	Quality control is a product oriented activities.	Testing is product oriented activities.
Quality assurance preventive activities.	Quality control is a corrective process.	Testing is a preventive process
It is a subset of Software Test Life Cycle (STLC)	QC can be considered as the subset of Quality Assurance.	Testing is the subset of Quality Control.

Table 7.2 Difference between Testing, Quality Assurance and Quality Control

7. ***Test Oracles:*** The test oracle defines a method to understand the determined and expected behavior of a system in order to test the test case for checking abnormality in behavior. In other words, a

test oracle is a mechanism, different from the program itself, which can be used to check the correctness of the output of the program for the test cases.

The following diagram shows how test oracle is useful to determine the expected behavior of the system

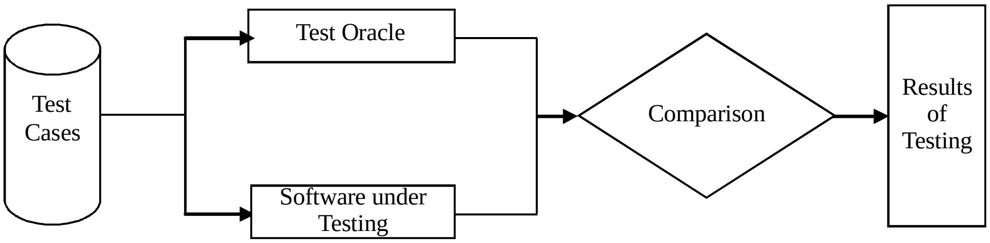


Fig 7.1 Working of Test Oracle

The above diagram shows testing process in which the test cases are given to the test oracle and the program under testing. The output of the two is compared to determine if the program behaved correctly for the test cases.

The human oracles generally use the specifications of the program to decide what the correct behavior of the system should be. To help the oracle to determine the correct behavior, it is important that the behavior of the system be unambiguously specified and the specification itself should be error free. Test oracles are human beings, so they may make mistakes when there is a discrepancy between the oracles and the results of a program. We need to verify the result produced by the oracle before declaring that there is a fault in the program.

8. ***Software Testability:***

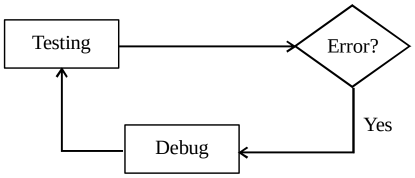
- Software testability is how easily a computer program can be tested.
- It is used to measure how adequately a particular set of tests will cover the product.
- The characteristics which leads to testable software are: operability, observability, decomposability, simplicity, stability and understandability.

9. ***Attributes of a Good Test***: The following are some attributes of good test:

- A good test has higher chances of finding errors.
- A good test is neither too simple nor too complex.
- A good test takes less time and resources.
- A good test is not redundant.

7.4 SOFTWARE DEBUGGING

Testing and debugging are different terms. Debugging is the process of locating and fixing errors in program code. Testing only determine the errors in the code. Testing does not fix those identified code. Whereas debugging is used to locate the errors and fix the located errors.



Difference between Testing and Debugging

The testing involves the identification of bugs in the program without correcting it. Normally people with a quality assurance background are involved in the identification of bugs. Testing is performed in the testing phase. The debugging involves fixing the bugs. The developers who coded the program conduct debugging.

Testing	Debugging
Testing is a process of finding and locating of a defect in a program	The debugging is process of fixing the defects in the program
Testing is done by the testing team	The debugging is done by the development team
Testing is done with intention to find as many defect as possible	The debugging is focused in removing the defects

Table 7.3 Difference between Testing and Debugging

7.5 PLANNING OF TEST

A test plan is a document consisting of different test cases designed for different testing objects and different testing attributes. The plan puts the tests in logical and sequential order per strategy chosen, top-down or bottom-up. The test plan is a matrix of test and test cases listed in order of its execution.

A test plan states:

- The items to be tested.
- At what level they will be tested at.
- The sequence they are to be tested in.
- How the test strategy will be applied to the testing of each item and the test environment.
- The software tools required and estimated hardware utilization.
- What will be the constraints which affect the testing process?

Test Case**:** A test case is a set of instructions designed to discover a particular type of error or defect in the software system by inducing a failure. The goal of selected test cases is to ensure that there is no error in the program and if there is it then should be immediately depicted. Ideal test casement should contain all inputs to the program. This is often called exhaustive testing.

There are two criteria for the selection of test cases:

- Specifying a criterion for evaluating a set of test cases.
- Generating a set of test cases that satisfy a given criterion.

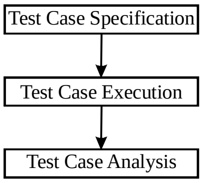


Fig. 7.2 Process of Test Case

1. **Test Case Specification:** Test plan focuses on approach; does not deal with details of testing a unit.Test case specification has to be done separately for each unit. Expected outcome also needs to be specified for each test case. Together the set of test cases should detect most of the defects. Would like the set of test cases to detect any defect, if it exists. The effectiveness and cost of testing depends on the set of test cases. Preparing test case specifications is challenging and time consuming. So for each testing, test case specifications are developed, reviewed, and executed.

2. **Test Case Execution:** Executing test cases may require drivers or stubs to be written; some tests can be automatic, others manual A separate test procedure document may be prepared

3. **Test Case Analysis:** Test summary report is often an output – gives a summary of test cases executed, effort, defects found, etc. the task of test case analysis are monitoring of testing effort is important to ensure that sufficient time is spent, computer time also is an indicator of how testing is proceeding, quality control focuses on removing defectsm and goal of defect prevention is to reduce the defect injection rate in future

10.6 SOFTWARE VERIFICATION AND VALIDATION

The validation and verification are two different functions in software engineering. We need to understand the difference between them.

Verification: All actions are taken at the end of software development phase to check whether we are building the product right. Verification is done at the end of each phase of software development cycle.

Validation: All actions are taken at the end of software development cycle to check whether we are building the right product. Validation is done at the end of complete development of the product.

These terms are often used interchangeably but following table shows how these terms are different:

Sr.No.	Verification	Validation
1	Verification is aimed to answer question like, Are we building the product right?	Validation is aimed to answer question like, Are we building the right product?
2	Verification ensures that the software system meets all the functionality.	Validation ensures that functionalities meet the intended behavior.
3	Verification takes place first and includes the checking for code, documentation etc.	Validation occurs after verification and mainly involves the checking of the overall product.
4	Verification is done by developers who developed that software.	Validation is done by expert team of testers.
5	Verification is a static activity as it includes the reviews, walkthroughs and inspections to verify that software is correct or not.	Validation has dynamic activities as it includes executing the software against the requirements.
6	It is an objective process and no subjective decision should be needed to verify the software.	It is a subjective process and involves subjective decisions on how well the software works.

Table 7.4 Difference between verification and validation

7.7 BLACK BOX TESTING

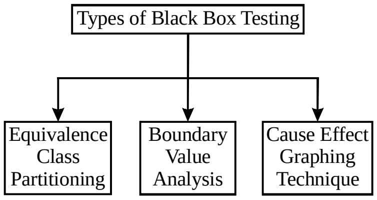
1. Black box testing is also called behavioral testing, focuses on the functional requirements of the software. Black box testing enables the software engineer to derive sets of input conditions that will fully exercise all functional requirements for a program.

2. In black-box testing, the tester only knows the inputs that can be given to the system and what output the system should give. In other words, the basis for deciding test cases in functional testing is the requirements or specifications of the system or module. This form of testing is also called functional or behavioral testing.

3. Black-box testing is not an alternative to white-box techniques. It is acomplementary approach that is likely to uncover a different class of errors thanwhite-box methods.It is applied at the later stages of testing. Black-box testing identifies the following kinds of errors:

- Incorrect or missing functions.
- Interface missing or erroneous.
- Errors in data model.
- Errors in access to external data source.

Types of Black-box Testing:


Types of Black Box Testing
Equivalence Class Partitioning
Boundary Value Analysis
Cause Effect Graphing Technique

7. 3 Types of Black-box Testing

Equivalence class partitioning and boundary value analysis both are test case design strategies in black box testing.

1. ***Equivalence Class Partitioning:***

- The domain of the input values partitioned into set of equivalence class. According to equivalence class partitioning, testing the code with one value from an equivalence class is as good as testing the code with any other value from the same equivalence class.
- In this method the input domain data is divided into different equivalence data classes. It is used to reduce the total number of test cases to a finite set of test cases which covers maximum requirements.
- It is a process of taking all possible test cases and placing them into different classes. One test value is picked from each class while testing.
- Example: If we test for an input box which accepts numbers from 1 to 1000 then there is no use in writing thousand test cases for all 1000 valid input numbers and test cases for invalid data. Using equivalence partitioning method, we can divide our test cases into three equivalence classes of some valid and invalid inputs.

2. ***Boundary Value Analysis:***

- Boundary value analysis is a next part of equivalence class partitioning for designing test cases where test cases are selected at the edges of the equivalence classes.
- This technique is used to identify errors at boundaries rather than finding those exist in center of input domain.More application errors occur at the boundaries of input domain.
- It is often called stress and negative testing.
- Example: If we test for an input box which accepts numbers from 1 to 1000 then test cases are: one test case for exact boundary values of input domains each means 1 and 1000, one test case for just below boundary value of input domains each means 0 and 999, one test case for just above boundary values of input domains each means 2 and 1001.

3. ***Cause Effect Graphing Techniques:***

- The Cause-Effect graph technique restates the requirements specification in terms of logical relationship between the input and output conditions. The causes are the input conditions and effects are the results of those input conditions.
- This technique is used to recognize the root causes, the cause for an exact effect, problem, or outcome.
- In this technique, first we recognize the input conditions (causes) and actions (effect). Then we build up a cause-effect graph and convert cause-effect graph into a decision table. Finally we convert decision table rules to test cases.

Advantages of Black-box Testing

Sr. No	Advantages
1	The test is unbiased because the designer and the tester are independent of each other.
2	The tester does not need knowledge of any specific programming languages.
3	The test is done from the point-of-view of the user, not the designer.
4	Test cases can be designed as soon as the specifications are complete.
5	Well suited and efficient for large code segments.
6	Code Access not required.
7	Clearly separates user's perspective from the developer's perspective through visibly defined roles.
8	Large numbers of moderately skilled testers can test the application with no knowledge of implementation, programming language or operating systems.

Table 7.5 Advantages of Black-box Testing

Disadvantages of Black-box Testing

Sr. No.	Disadvantage
1	Limited Coverage since only a selected number of test scenarios are actually performed.
2	Inefficient testing, due to the fact that the tester only has limited knowledge about an application.
3	Blind Coverage, since the tester cannot target specific code segments or error prone areas.
4	The test cases are difficult to design.

Table 7.6 Disadvantages of Black-box Testing

7.8 WHITE BOX TESTING

White-box testing is also known by other names, such as glass-box testing, structural testing, clear-box testing, open-box testing, logic-driven testing, and path-oriented testing. A complementary approach to functional or black-box testing is called structural or white-box testing. In this approach, test groups must have complete knowledge of the internal structure of the software. We can say structural testing is an approach to testing where the tests are derived from knowledge of the software's structure and implementation. Structural testing is usually applied to relatively small program units, such as subroutines, or the operations associated with an object. As the name implies, the tester can analyze the code and use knowledge about the structure of a component to derive test data. The analysis of the code can be used to find out how many test cases are needed to guarantee that all of the statements in the program are executed at least once during the testing process. It would not be advisable to release software that contains untested statements as the consequence might be disastrous.

The nature of software defects are:

Sr. No.	Nature of software defects in white box testing
1	Logical errors and incorrect assumptions are inversely proportional to the probability that a program path will be executed.
2	We often believe that a logical path is not to be executed when, in fact, it may be executed on a regular basis.
3	Typographical errors are random. When a program is translated into programming language source code, it is likely that some typing errors will occur.

Table 7.7 Nature of Software Defects in White Box Testing

Different forms of White Box Testing

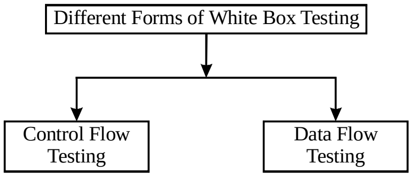


Fig. 7.4 Different forms of White Box Testing

- **Control Flow Testing:** Control flow testing is a structural testing strategy which uses the program's control flow. It is a directed graph where the nodes represent the processing statements like definition, computation and predicates while the edges represent the flow of control between processing statements. It is more effective for unstructured code rather than structured code. But it cannot catch initialization and specification mistakes.Toexamine the patterns, we need to construct a control flow graph of the code.
- **Data Flow Based Testing:** Data flow testing is a technique used to detect improper use of data in a program. It is a white box testing technique thatcan be used to detect improper use of data values due to coding errors. Data-flow testing uses the control flow graph to explore the unreasonable things that can happen to data. Data-flow testing is the set of test strategies based on selecting paths through the program's control flow in order to explore sequences of events related to the status of data objects. Data-

flow testing strategies span the gap between all paths and branch testing.Data-flow testing monitors the lifecycle of a piece ofdata and identifies potential bugs byexamining the patterns in which that piece of data is used.

Reasons Why White-box Testing is performed

Sr. No.	**Reasons**
1	All paths in a process are correctly operational.
2	All logical decisions are executed with true and false conditions.
3	All loops are executed with their limit values tested.

Table 7. 8 Reasons Why White-box Testing is Performed

Types of White Box Testing

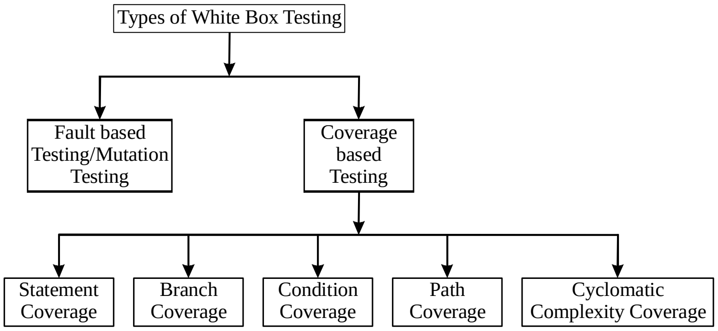

Types of White Box Testing
Fault based Testing/Mutation Testing
Coverage based Testing
Statement Coverage
Branch Coverage
Condition Coverage
Path Coverage
Cyclomatic Complexity Coverage

Fig. 7.5 Types of White Box Testing

1. **Fault Based Testing:** It is used to detect certain types of faults. Its goal is to determine the absence of faults rather than the goal of "finding errors" .The focus is on faults rather than errors. It is not a manual testing. The example of fault based testing is mutation testing.
2. **Coverage Based Testing:** A coverage based testingstrategy aims to execute certain program elements to discover failures. It is of four types:

- ***Statement Coverage:***It aims to design test cases so as to executeall the statements of a program. In this strategy, every statement in a program is to be executed at least once. The disadvantage of this strategy is that after executing a statement for one input value once, it assumes that it behave properly but there is no guarantee that it will behave correctly for all input values.
- ***Branch Coverage:***It is also known as edge testing. It is used to design test cases in such a way that it assume true and false value in return. Branch coverage testing is stronger than statement coverage testing because branch coverage ensures statement coverage but statement coverage does not ensure branch coverage.
- ***Condition Coverage:*** In this testing strategy, test cases are designed to make each component of a conditional expression to assume both true and false values. Condition coverage testing is a stronger testing strategy than branch coverage and Branch testing is a simplified condition testing.
- ***Path Coverage:***It design test cases in such a manner that all linearly independent paths in the program are executed at least once. To understand each linearly independent path, we need to understand the control flow graph (CFG). A control flow graph is a directed graph which consists of set of nodes and edges (N, E). It describes how the control flows through the program.

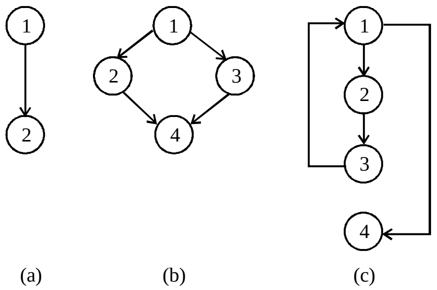


Fig. 7.6 Examples of Path Coverage

- ***Cyclomatic Complex Graph:*** *It tests the complexity of the program. It also estimates the amount of effort required to understand the code. The Cyclomatic complexity is defines as:*

CC=E-N+P
CC= Cyclomatic Complexity
E= Number of Edges
N= Number of Nodes
P= Number of Connected Components
Example:

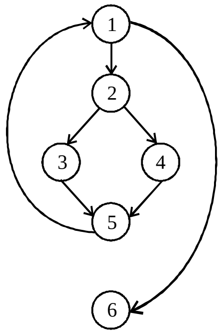


Fig. 7.7 Examples of Cyclomatic Complexity

If number of edges= 7 and number of nodes= 6, then cyclomatic complexity is:
CC= 7-6+2 => CC=3
Cyclomatic complexity helps in estimation of complexity of code, testing effort and program reliability.

Advantages of White-Box Testing:

Sr. No.	Advantages
1	As the tester has knowledge of the source code, it becomes very easy to find out which type of data can help in testing the application effectively.
2	It helps in optimizing the code.
3	Extra lines of code can be removed which can bring in hidden defects.
4	Due to the tester's knowledge about the code, maximum coverage is attained during test scenario writing.

Table 7.9 Advantages of White Box Testing

Disadvantages of White-Box Testing:

Sr. No.	Disadvantages
1	Due to the fact that a skilled tester is needed to perform white box testing, the costs are increased.
2	Sometimes it is impossible to look into every nook and corner to find out hidden errors that may create problems as many paths will go untested.
3	It is difficult to maintain white box testing as the use of specialized tools like code analyzers and debugging tools are required.

Table 7.10 Disadvantages of White Box Testing

7.9 GREY BOX TESTING

Grey Box testing is a technique to test the application with limited knowledge of the internal workings of an application. In software testing, the term “the more you know the better” carries a lot of weight when testing an application.

Mastering the domain of a system always gives the tester an edge over someone with limited domain knowledge. Unlike black box testing, where the tester only tests the application’s user interface, in grey box testing, the tester has access to design documents and the database. Having this knowledge, the tester is able to better prepare test data and test scenarios when making the test plan.

Advantages:

1. Offers combined benefits of black box and white box testing wherever possible.
2. Grey box testers don’t rely on the source code; instead they rely on interface definition and functional specifications.
3. Based on the limited information available, a grey box tester can design excellent test scenarios especially around communication protocols and data type handling.
4. The test is done from the point of view of the user and not the designer.

Disadvantages:

Since the access to source code is not available, the ability to go over the code and test coverage is limited. The tests can be redundant if the software designer has already run a test case. Testing every possible input stream is unrealistic because it would take an unreasonable amount of time; therefore, many program paths will go untested.

7.10 DIFFERENCE BETWEEN BLACK BOX TESTING, GREY BOX TESTING AND WHITE BOX TESTING

Sr. No.	Black Box Testing	Grey Box Testing	White Box Testing
1.	The Internal Workings of an application are not required to be known	Somewhat knowledge of the internal workings are known	Tester has full knowledge of the Internal workings of the application
2.	Also known as closed box testing, data driven testing and functional testing	Another term for grey box testing is translucent testing as the tester has limited knowledge of the insides of the application	Also known as clear box testing, structural testing or code based testing
3.	Performed by end users and also by testers and developers	Performed by end users and also by testers and developers	Normally done by testers and developers
4.	Testing is based on external expectations Internal behavior of the application is unknown	Testing is done on the basis of high level database diagrams and data flow diagrams	Internal workings are fully known and the tester can design test data accordingly
5.	This is the least time consuming and exhaustive	Partly time consuming and exhaustive	The most exhaustive and time consuming type of testing
6.	Not suited to algorithm testing	Not suited to algorithm testing	Suited for algorithm testing

Table 7.11 Difference Between Black Box Testing, Grey Box Testing and White Box Testing

7.11 LEVELS OF TESTING

There are four levels of testing, i.e., four individual modules in the entire software system.

1. Unit Testing
2. Integration Testing
3. System Testing
4. Acceptance testing

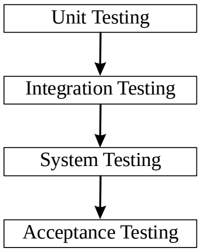


Fig 7. 8 Level of Testing

7.11.1 Unit Testing

The unit test is the lowest level of testing performed during software development, where individual units of software are tested in isolation from other parts of a program. The individual components are tested to ensure that they work correctly. It focuses on the smallest unit of software design where each component is tested independently without other system components. There are number of reasons to do unit testing rather than testing the entire product:

Reasons to perform Unit Testing:

- The size of a single unit is small that is easy to locate error in it.
- The unit is small enough that we can attempt as much as test cases.
- The interactions of multiple errors in it are eliminated.

Benefits of Unit testing:

The unit testing has the following benefits:

1. **Increases confidence:** The unit testing increases confidence in changing/maintaining code. If good unit tests are written and if they are run every time any code is changed, the likelihood of any defects due to the change being promptly caught is very high. If unit testing is not in place, the most one can do is hope for the best and wait till the test results at higher levels of testing are out. Also, if codes are already made less interdependent to make unit testing possible, the unintended impact of changes to any code is less.
2. **Reusability:** The codes are more reusable. In order to make unit testing possible, codes need to be modular.
3. **Fastest development:**The development is faster as the effort required to find and fix defects found during unit testing is easy as comparison to finding defects during system testing or acceptance testing.
4. **Fewer efforts:**The cost of fixing a defect detected during unit testing is lesser in comparison to that of defects detected at higher levels.
5. **Easy Debugging:**Due to unit testing the debugging is easy. When a test fails, only the latest changes need to be debugged.
6. **Error discovery:**Unit testing can discover error like Comparison of different data types, Incorrect logical operators or precedence, Expectation of equality when precision error makes equality unlikely, Incorrect comparison of variables, Improper loop termination, Failure to exit when divergent iteration is encountered, Improperly modified loop variables

Unit-test Procedure:

In unit testing, a module is tested under well-defined environment which includes all relevant code that is required to execute the module. The following diagram shows environment:

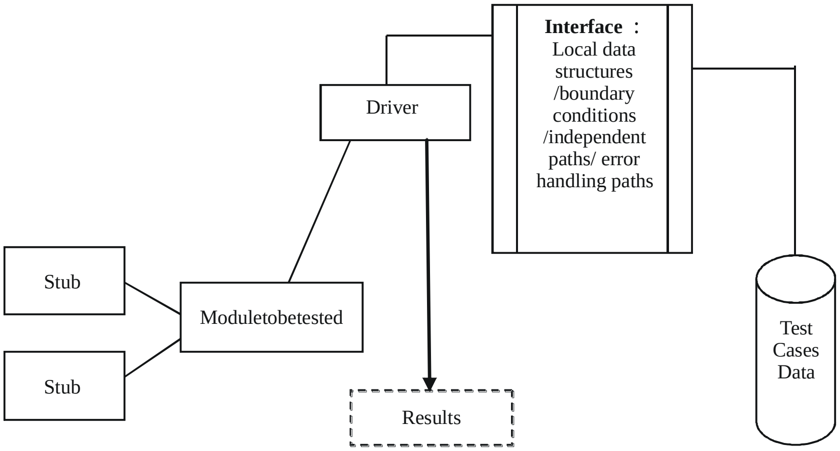

Interface :
Local data structures /boundary conditions /independent paths/ error handling paths
Driver
Stub
Moduletobetested
Stub
Results
Test Cases Data

Fig. 7.9 Unit-test Environment

In the unit test environment, there are four components namely stub, driver, interface, test cases.

Stubs: The Stubs serve to replace modules that are subordinate to the component to be tested.

Driver: A driver is nothing more than a "main program" that accepts test-case data, passes such data to the component (to be tested), and prints relevant results.

Interface: The interface manages takes like integration of local data structure, keep check on boundary condition and possible path to execute the code. The local data structure is examined to ensure that data stored temporarily maintains its integrity during all steps in an algorithm's execution. Boundary conditions are tested to ensure that the module operates properly at the boundaries established to limit. All independent paths through the control structure are exercised to ensure that all statements in a module have been executed at least once.

Test case data: The test case data is collection of all possible input that the can be observed during the testing phase of a particular unit.

7.11.2 **Integration Testing**

The second level of testing is called integration testing. Integration testing is a systematic technique for constructing the program structure while at the same time conducting tests to uncover errors associated with interfacing. In this testing many unit-tested modules are combined into subsystems, which are then tested. The goal here is to see if the modules can be integrated properly.

Objective of Integration Testing:

The primary objective of integration testing is to test the module interfaces in order to ensure that there are no errors in the parameter passing, when one module invokes another module. During integration testing, different modules of a system are integrated in a planned manner using an integration plan. The integration plan specifies the steps and the order in which modules are combined to realize the full system. After each integration step, the partially integrated system is tested.

Approaches to Integration Testing:

The various approaches used for integration testing are:

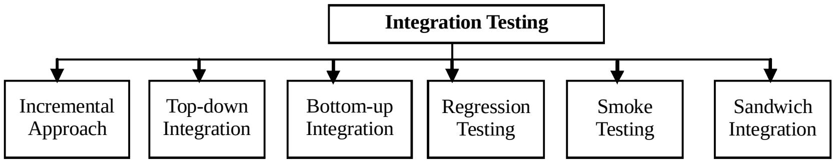


Fig. 7.10 Approaches to integration testing

- **Incremental Approach:** According to the incremental approach, testing starts with first combine only two components together and performs testing on them. If there are some error then removes them first, otherwise combine another component to it and then test again, and so on until the whole system is developed. The following logical diagram shows the same process:

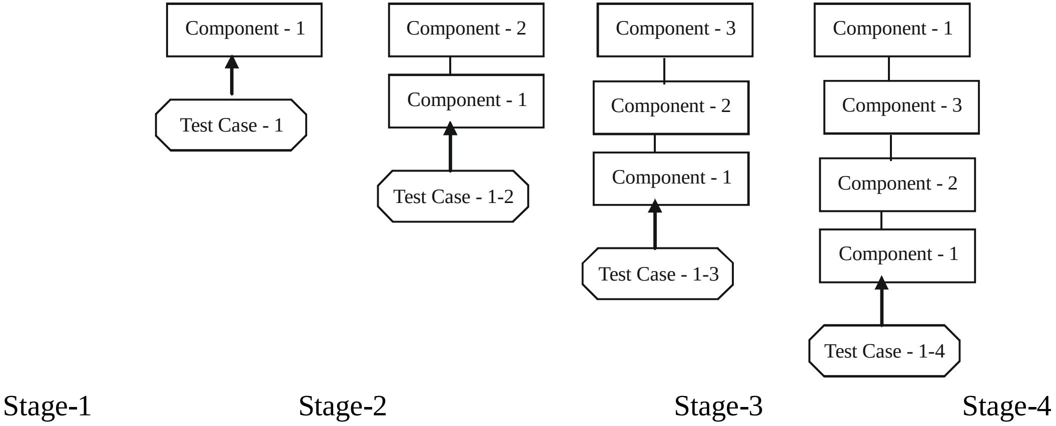


Fig. 7.11 Incremental approach for testing

According to Figure, in incremented testing is performed in four stages where in stage1, component1 is tested if it is working correctly then it is integrated with componets2 and gain test cases1 to 2 are used for testing. If these components are corrected or error free then module componets3 is integrated, i.e. stage3 and then tests case 1 to 3 applied on all the components. The process is repeated till whole system is not integrated and tested.

- **Top-Down Integration Testing**

Top-down integration testing is an incremental approach to construction of program structures. Modules are integrated by moving downward through the control hierarchy beginning with the main control module.

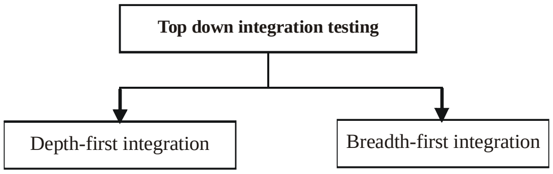
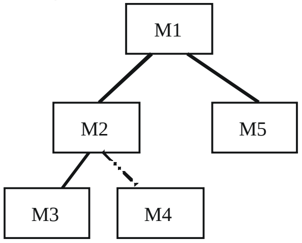


Fig. 7.12 Types of Top down Integration Testing

Depth-first integration: According to depth first integration, we integrate all components on a major control path of the structure. Let us consider example: For example, M1, M2, and M3 would be integrated first. Next, M4, M6 would be integrated. Then, the left side depth control paths are built.



Breadth-first integration: According to breadth first integration all components directly subordinate at each level, moving across the structure horizontally will be incorporated as shown in figure. The components M2, M5 will be integrated first. The next control level incorporate M3, M4.Figure: Top-down integration testing

- **Bottom up Integration Testing**. The bottom-up integration testing begins construction and testing with the components at the lowest level in the program structure. Then low level tested components are integrated and again tested as group. A bottom-up integration strategy may be implemented with the following steps:

1. Low-level components are combined into clusters that perform specific software sub functions.
2. A driver is written to coordinate test case input and output.
3. The cluster is tested.
4. Drivers are removed and clusters are combined moving upward in the program structure.

The bottom up integration is demonstrated in the following diagram:

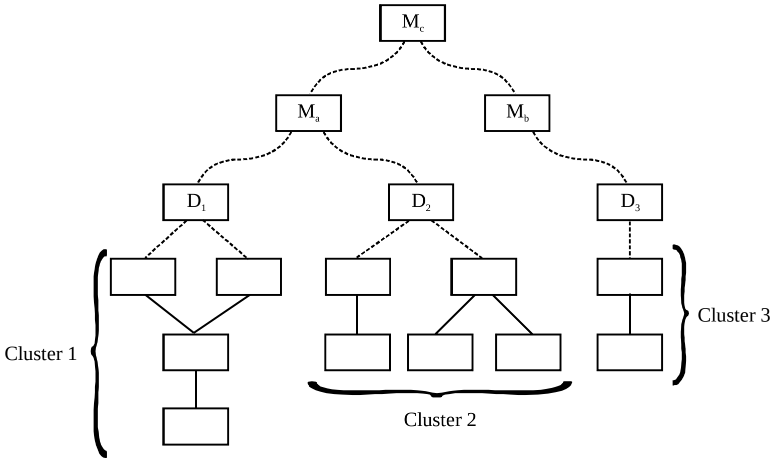


Fig. 7. 13 Bottom-up Integration Testing

The components are combined to form clusters 1, 2, 3. Every cluster is tested using a driver and components in clusters 1 and 2 are subordinate to M*a*. Drivers D1 and D2 are removed and the clusters are interfaced directly to M*a*. Similarly, driver D3 for cluster 3 is removed prior to integration with module M*b*. Both M*a* and M*b* will ultimately be integrated with component M*c*and so forth.

- **Regression Testing:** Regression testing is the activity that helps to ensure that changes (due to testing or for other reasons) do not introduce unintended behavior or additional errors. The regression test suite contains three different classes of test cases:

— Additional tests that focus on software functions.
— A representative sample of tests that will exercise all software functions.
— Tests that focus on the software components that have been changed.

- **Smoke Testing:** Smoke testing is an integration testing approach that is commonly used when “shrink-wrapped” software products are developed. Smoke testing is characterized as a rolling integration approach because the software is rebuilt with new components and testing. Smoke testing encompasses the following activities:

- Software components that have been translated into code are integrated into a “build.” A build includes all data files, libraries, reusable modules, and engineered components that are required to implement one or more product functions.

- A series of tests is designed to expose errors that will keep the build from properly performing its functions. The build is integrated with other builds and the entire product (in its current form) is smoke tested daily.

- **Sandwich Integration Testing.** Sandwich integration testing is the combination of both the top-down and bottom-up approach. So, it is also called mixed integration testing. In it, the whole system is divided into three layers, just like a sandwich: the target is in the middle and one layer is above the target and one is below the target.

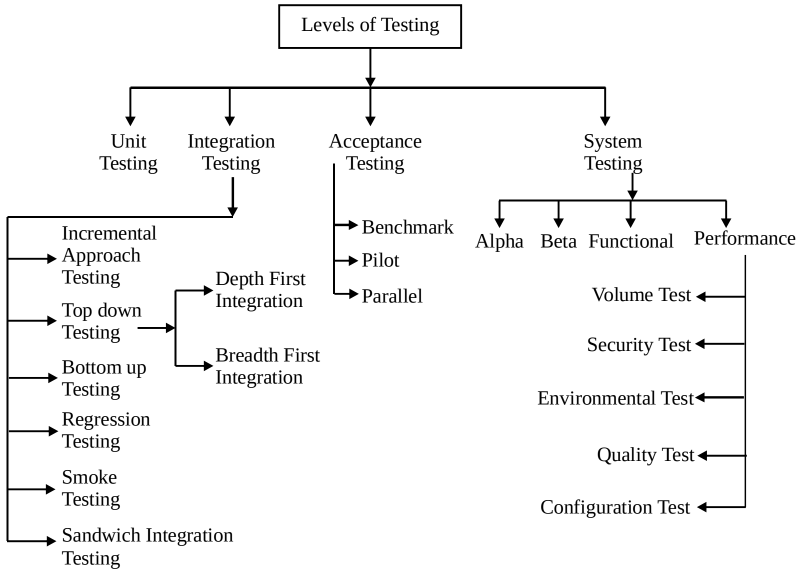


Fig 7.14 Levels of Testing in Detail

7.11.3 **System Testing**

Once all the components are integrated, the application as a whole is tested thoroughly to see that it meets Quality Standards. This type of testing is performed by a specialized testing team. It is also concerned with validating that the system meets its functional and non-functional requirements. There are essentially three main kinds of system testing:

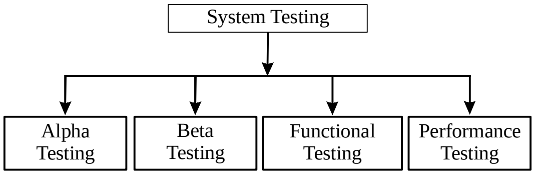


Fig 7.15 Classification of System Testing

- ***Alpha Testing***: *Alpha testing refers to the system testing carried out by the test team within the development organization.*

The alpha test is conducted at the developer’s site by the customer underthe project team’s guidance. In this test, users test the software on thedevelopment platform and point out errors for correction. However, thealpha test, because a few users on the development platform conduct it, has limited ability to expose errors and correct them. Alpha tests are conducted in a controlled environment. It is a simulation of real-life usage. Once the alpha test is complete, the software product is ready for transition to the customer site for implementation and development.

- ***Beta Testing***: *Beta testing is the system testing performed by a selected group of friendly customers.*

If the system is complex, the software is not taken for implementation directly. It is installed and all users are asked to use the software in testing mode; this is not live usage. This is called the beta test. Beta tests are conducted at the customer site in an environment where the software is exposed to a number of users.

The developer may or may not be present while the software is in use. So, beta testing is a real-life software experience without actual implementation. In this test, end users record their observations, mistakes, errors, and so on and report them periodically.

In a beta test, the user may suggest a modification, a major change, or a deviation. The development has to examine the proposed change and put it into the change management system for a smooth change from just developed software to a revised, better software. It is standard practice to put all such changes in subsequent version releases.

- ***Functional Testing:*** *Functional Testing of the software is conducted on a complete, integrated system to evaluate the system's compliance with its specified requirements.*

This is a type of black box testing that is based on the specifications of the software that is to be tested. The application is tested by providing input and then the results are examined that need to conform to the functionality it was intended for. There are five steps that are involved when testing an application for functionality.

1. The determination of the functionality that the intended application is meant to perform.
2. The creation of test data based on the specifications of the application.
3. The output based on the test data and the specifications of the application.
4. The writing of Test Scenarios and the execution of test cases.
5. The comparison of actual and expected results based on the executed test cases.

An effective testing practice will see the above steps applied to the testing policies of every organization and hence it will make sure that the organization maintains the strictest of standards when it comes to software quality.

- ***Performance Testing:*** Performance testing checks the performance objectives set by the customer. It is also known as nonfunctional testing. It measures the accessibility of data, accuracy of the output, speed of response etc. The following are the types of performance testing:

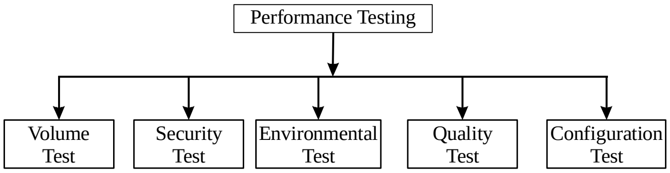


Fig. 7.16 Types of Performance Testing

Volume Test: It address the handling of large amount of data in the system.

Security Test: It ensures the security requirements of the system.

Environmental Test: This test ensures performance of system under the tolerance for heat, motion, portability, moisture and other environmental situations.

Quality Test:It ensures the reliability, availability and maintainability of the system.

Configuration Test: It ensures the various software and hardware configurations explained in the requirements.

7.11.4 **Acceptance Testing**

Acceptance testing is the system testing performed by the customer to determine whether to accept or reject the delivery of the system.

When customer software is built for one customer, a series of acceptancetests are conducted to enable the customer to validate all requirements. Conducted by the end-user rather than the software engineers, an acceptance test can range from an informal 'test drive' to a planned and systematically executed series of tests. In fact, acceptance testing can be conducted over a period of weeks or months, thereby uncovering cumulative errors that might degrade the system over time. Acceptance tests are written, conducted and evaluated by the customers.

Types of acceptance tests:

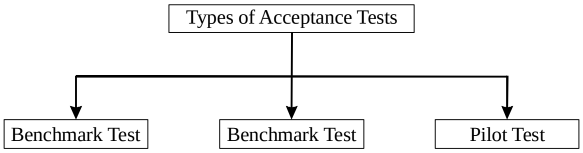


Fig. 7.17 Types of Acceptance Tests

- ***Benchmark Test:***These tests are performed when a user has special requirements. These tests are performed by actual user or a special team.
- ***Pilot Test:***Pilot tests rely on all the functions of everyday working of the system. User exercise the system as if it had been installed permanently.
- ***Parallel Test:*** In this test, the new system installs and operates in parallel with the previous version. It builds the confidence of user in new system with the old system.

M.SC. (COMPUTER SCIENCE)

SEMESTER-2

COURSE: SOFTWARE ENGINEERING

UNIT 8: SOFTWARE MAINTENANCE AND RELIABILTY

8.1 **INTRODUCTION**

Software maintenance is the activity associated with keeping an operationalcomputer system continuously in tune with the requirements of users and data processing operations. It process is expensive and risky and is very challenging.Maintenance may be classified into the four categories as follows: Corrective,Adaptive,Perfective,and Preventive. The term software reliability is defined as the probability of failure-free operation of a computer program in a specified environment for a specified time. Thedetaileddiscussion about categories of software maintenance along with different components of software maintenance procedures are discussed with appropriate examples.

8.2 **SOFTWARE MAINTENANCE**

1. After the complete development of software product, software is handed over to the client. It client requires any changes in the software product after the delivery then it is turned as software maintenance. Software maintenance is the most difficult aspect of software development.

Definition

According to ***IEEE***: ***“Software maintenance is a process of modification in a software product after delivery to correct faults, or to improve performance so that software product can adopt the modified environment.”***

2. Maximum time is spent on software maintenance as compared to the development of any phase of the software product. So it is very important to kept maintenance in mind throughout the software development.
3. Once an error has been located, the maintenance programmer resolves the errors without adding any error (regression errors). Detailed documentation helps the maintenance programmer in correcting the errors.
4. After receiving a maintenance request, the first step is to identify what type of error is there and what type of maintenance is required.
5. After correcting an error, the fix is tested again to ensure the correct functionality of the software. It also ensure that there should be no other side effects.
6. There are number of software life cycle models which ignore the software maintenance phase and do not provide any adequate information for integration of maintenance into development process.
7. Software development and software maintenance are two processes of software engineering. Software maintenance is based on the information gained from software development. Without this information, software maintenance is very difficult process designing, coding, testing etc.

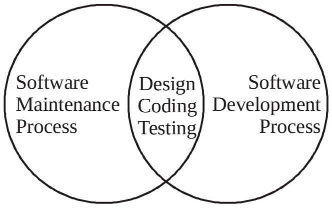


Fig. 8.1 Relationship between Software Maintanance Process & Software Development Process

8. Software maintenance costs are underestimated during the design and implementation phase. But in reality software maintenance costs are the greatest costs.
9. One of the issue regarding maintenance is that software engineers have very poor image of maintenance. They it is unimportant and easy task. That is why organizations allocated this maintenance task into very less skilled and inexperienced staff which results in very high maintenance cost and time.
10. The maintenance cost and time can be reduced by involving experienced and skilled staff early in the software process.
11. In short, software maintenance means modification in a software product after delivery. It mostly occurs at customer's site. Software maintenance include error detection, error correction and enhancement of capabilities.

8.2.1 **Objectives of Software Maintenance**

Software maintenance is a set of activities performed in a software product after delivery. Example of software maintenace are changes in code documentation manuals or any part of the product etc. Software maintenance has many objectives which are as follows:

1. Software maintenance enhances the performance of the software.
2. It correct various errors occur during the functionality of the software.
3. It makes changes in the software so that it can adopt the changes in the environment.
4. Software maintenance also update the software product to avoid future problems.

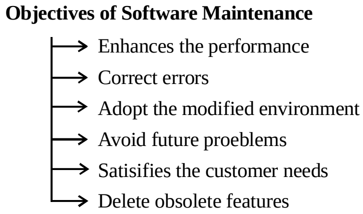

Objectives of Software Maintenance
Enhances the performance
Correct errors
Adopt the modified environment
Avoid future proeblems
Satisifies the customer needs
Delete obsolete features

Fig. 8.2 Objectives of Software Maintenance

5. Maintenance process satisfies the customer needs. After maintenance, software starts working according to user requirements.
6. Software maintenance is used to correct program errors, to add new capabilities into the software to delete obsolete features and to improve performance of the software.

8.2.2 Categories of Software Maintenance

Software maintenance is categorized into four types which are as follows:

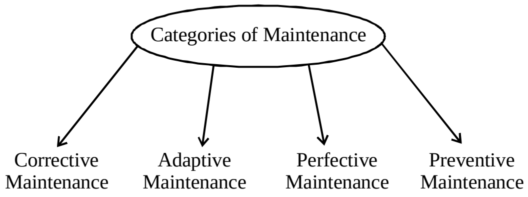


Fig. 8.3 Categories of Maintenance

I. Corrective Maintenance

1. Corrective maintenance is necessary to resolve the bugs occur during the usage of a software product. Bugs can occur at any stage during the development of software for example—bugs can occur during the specifications, design, coding and testing etc.
2. Corrective maintenance changes the software to correct the errors. Some errors are caused by data processing error and system performance errors.
3. An error can result from design errors, logic errors and coding errors which are as follows:

- **Designing errors:** These errors occurs when change are made in the software. These changes are incorrect, incomplete or misunderstood and arises errors.
- **Logic errors:** These errors occurs when invalid tests, incomplete test data of design specification and incorrect implementations are made in the software.
- **Coding errors:**These errors occurs when incorrect use of source code logic and wrong implementation of detailed logic design are made in the software.

II. Adaptive Maintenance

1. Adaptive maintenance makes changes in the software so that it can work in new environment (new platforms) without any error. For example—porting to a new compiler, operating system and/or hardware.
2. The term new environment refers to the totality of all conditions/influences that act from outside on the software.
3. Adaptive maintenance in the software is not the need of the client. It is the need of external environment in which software works.

III. Perfective Maintenance

1. Perfective maintenance needs time and money. It improves the efficiency, effectiveness and maintainability of the software.
2. In perfective maintenance, additions or enhancements are made in the software to improve its functionality and quality.
3. The perfective maintenance is requested by software engineer and the client. The software engineer needs perfective maintenance to improve the status of the product in market by improving its quality. The client needs perfective maintenance to meet the new requirements.

IV. **Preventive Maintenance**

1. Preventivemaintenance performs the activities/operations on the software to update it in anticipation of any feature problems.
2. It performs modifications in a software after delivery, to improve the effectiveness, efficiency and performance so that software does not face any problem (errors) in the future.
3. Preventive Maintenance is basically a modification in the software after the delivery to detect and correct the faults. It ensures the safety of the software.

8.2.3 Components **of Software Maintenance Process**

The components mean those parts which play important role in software maintenance. The important components of the software maintenance process are people, tasks and knowledge of the software.These components are inter-related and interdependent on each other.

1. **People:** It is the most important component of the software maintenance process. This component covers all those persons which are the part of the maintenance process directly or indirectly. It includes users, development staff and maintenance staff.

- Users are those people for whom software is developed and maintained.
- Development staff is a team of those people who takes responsibility for the successful development of the software product
- Maintenance staff includes those people who participated actively in the software product after the complete development of the software to make it up to date.

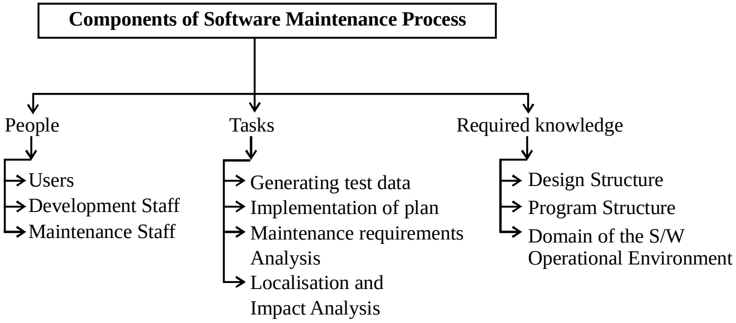

Components of Software Maintenance Process
People
Tasks
Required knowledge
Users
Development Staff
Maintenance Staff
Generating test data
Implementation of plan
Maintenance requirements Analysis
Localisation and Impact Analysis
Design Structure
Program Structure
Domain of the S/W Operational Environment

Fig. 8.4 Components of Software Maintenance Process

2. **Tasks:** Tasks means those activities or operations which are performed during successful maintenance process. During maintenance process, various tasks like implemation plan, maintenance requirements analysis, localization and impact analysis etc. are performed.

- Implementation plan provides the step by step maintenance requirement analysis and the need of maintenance requirement is checked.
- It is used to examine the requirements of the maintenance. Impact analysis include the detailed study of software product after the maintenance process. Performance, efficiency and effectiveness of software product is examined after the successful implementation of maintenance process.

3. **Required knowledge:** This component includes the structured and unstructured knowledge for the software maintenance. Maintenance staff should have the knowledge of the design and program structure of the software product. Maintenance staff should have complete knowledge of domain of the software alongwith the operational environments.

8.2.4 Software Maintenance Process

The maintenance process has five steps which are follows:

1. **Understand the existing programs:** In this step, before modification, existing programs are studied for complete understanding. Important attributes which are helpful in better understanding of existing programs are: less complexity and proper documentation. In this step, objectives are also studied in the detailed manner.

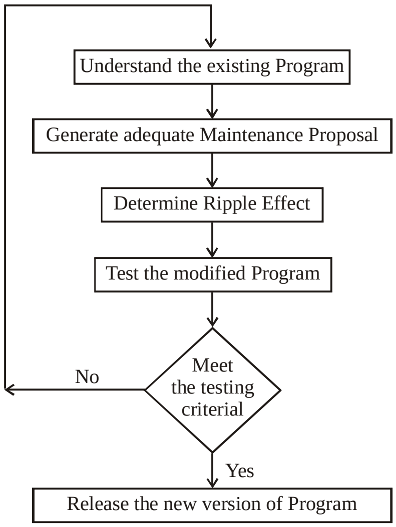

Understand the existing Program
Generate adequate Maintenance Proposal
Determine Ripple Effect
Test the modified Program
Meet the testing criterial
No
Yes
Release the new version of Program

Fig. 8.5 Steps of Software Maintenance Process

2. **Generate adequate maintenance proposal:** In this step, maintenance team generates and propose the adequate maintenance proposal to satisfy the maintenance objectives.

3. **Determine ripple effect:**

- Ripple effect determines the effect of the change (modification) on the system.
- Ripple effect is logical/functional in nature. It concerns the performance of the program.
- The ripple effect is affected by the consequence of a program modification i.e. the stability of the program.
- Program stability is basically a resistance to the amplification of changes in the program.
- Regression testing is performed to ensure that unchanged code is still working efficiently after making the new changes in the software product.

4. **Test the modified program:** In this step, modified program is tested to ensure the reliability level. Modified program should have at least same reliability level as before. Cost effective testing techniques are also performed (during the maintenance process) to measure the required effort to test the modified program, under well-defined testing conditions.

5. **Release the new version of program:** After meeting the testing conditions, modified programs with new versions are ready to install/release.

Maintenance Process Models

No single maintenance process model is suitable for all types of maintenance projects. Maintenance process activities are different for different maintenance projects.

There are two bread categories of maintenance process models are proposed.

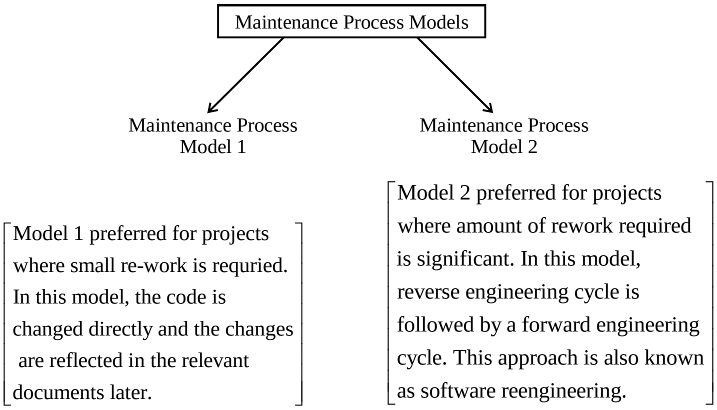

Maintenance Process Models
Maintenance Process Model 1
Maintenance Process Model 2
Model 1 preferred for projects where small re-work is requried. In this model, the code is changed directly and the changes are reflected in the relevant documents later.
Model 2 preferred for projects where amount of rework required is significant. In this model, reverse engineering cycle is followed by a forward engineering cycle. This approach is also known as software reengineering.

Fig. 8.6 Maintenance Process Models

1. **Maintenance Process Model 1:** The maintenance process model 1 starts with gathering the requirements for change. This is the first step of Model 1. In this step, all the change requirements are gathered in a detailed manners. In this step 2, these change requirements are analyzed and studied from different angles. In this step 3, after analyzing the change requirements, efficient strategies are formulated and adopted for code change. At this stage, few members of original development team goes a long way in reducing the cycle time, for the projects involving unstructured and inadequate documented code.

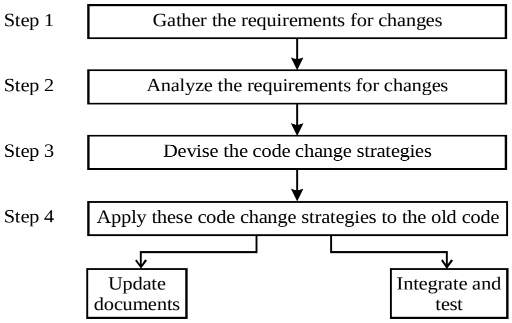

Step 1
Gather the requirements for changes
Step 2
Analyze the requirements for changes
Step 3
Devise the code change strategies
Step 4
Apply these code change strategies to the old code
Update documents
Integrate and test

Fig. 8.7 Maintenance Process Model 1

In this step 4, apply the formulated and adopted code strategies to the old code. Maintenance engineers study the working of the old system and compare the working of their modified system with the old system.

In the end, debugging of the new modified system become easy as after the comparison of the old and new modified system, bugs can be easily localize.In the model 1, the output is update documents and integrate and test the modified program.

2. **Maintenance Process Model 2**

- Maintenance Process Model 2 is basically suitable for those projects where amount of rework is required. In this model reverse engineering cycle is followed by a forward engineering cycle which is known as software re-engineering.
- During the reverse engineering, the old coding is analyzed to extract the module specification. The module specification is further analyzed for producing the designing part. The design is further analyzed to produce the original requirement specification. This process is known as reverse re-engineering because analyzing process is done in reverse order means from bottom to top stages.
- After analyzing the original requirement specifications, change requirements are applied to the original requirement specifications, to produce the new requirement specification. After arriving at the new requirement specification, forward engineering approach is used to produce a new code.

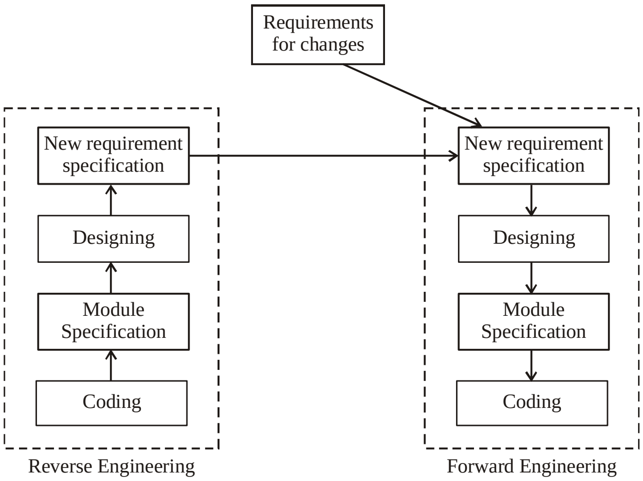


Fig. 8.8 Maintenance Process Model 2

- During the forward engineering approach, new requirement specifications are considered. At the designing, module specification and coding stages, a substantial reuse is made from the reverse engineered products. This approach produces good documentation, more structured design as compared to the original product.

Comparison between Model 1 and Model 2

1. The maintenance process model 2 gives efficient outputs, efficient designs and adequate documentation as compared to maintenance process model 1.
2. The maintenance process model 2 is cost lies than the process model 1.
3. The maintenance process model 2 is preferred only when amount of rework is very high. If amount of rework is not more than 20%, then we use model 1.

8.2.5 Factors affecting the Software Maintenance

There are broadly two categories of factors which affects the software maintenance.

I. **Technical Factors:** The technical factors of software maintenance are as follows:

1. **Programming Language:** Programming language plays important role in the software maintenance. Programs written in the high level language are easier to understand and maintained than the programs written in low level languages.
2. **Program Testing:** If a program is adequately validated and tested during the design phase, then there are less chances of occurring errors in the software product. If development staff spends maximum effort and time on the design testing then maintenance cost is reduced because of fewer errors in the program.

   Maximum time and effort in design phase = Less maintenance cost

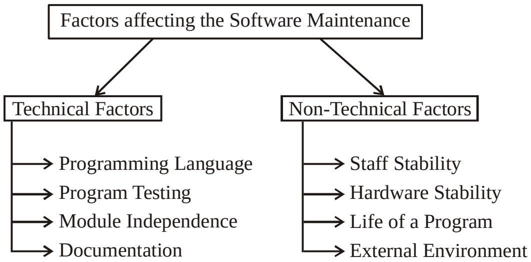


Fig. 8.9 Factors affecting the Software Maintenance

3. **Module Independence:** Module independence means modification of one program unit of a system without affecting any program unit of a system. If one module is being modified, then there should be no effect on other modules of software product.
4. **Documentation:** Documentation is another important factor of software maintenance. If a program is clearly, completely and adequately documented, then it provides the complete understanding of the program.

II. **Non-Technical Factors:** Then non-technical factors of software maintenance are as follows:

1. **Staff Stability:** The stability of staff effects the maintenance cost.

   More stable staff = less maintenance cost

   If an original writer of a program understands and changes a program, then it will be very easy for him to maintain that program and becomes very beneficial for the development company in tums of cost and time. It is very difficult for individual to understand and changes in a program written by another person (a new programmer).
2. **Hardware Stability:** The program must be modified to use new hardware. The new hardware replaces the old (obsolete) equipment. Sometimes a program is designed for a particular hardware configuration and the chances of the change of that particular hardware configuration is very less during the program’s whole life. In this situation there is no maintenance cost, but it is rarest case.
3. **Life of a Program:** As the life of a program increases, the maintenance cost increases. The life of a program depends upon the application used. When the application becomes obsolete, program becomes obsolete and maintenance cost increases very rapidly because of conversion of original obsolete hardware into new hardware.

   Life of Program = Maintenance cost and time
4. **External Environment:** Sometimes, program depends upon the external environments. According to the external environments, program needs changes and maintenance cost increases. Whole maintenance process and maintenance cost depends upon the external environment.

   Changes in external environment αHigher maintenance cost and time.

8.3 SOFTWARE RELIABILITY

1. Software reliability is an important concern for users. Software reliability is an attribute of software quality and it is independent from time.
2. More the software reliability, higher the software quality.
3. If software reliability increases, then the performance usability, functionality and stability of software increases.
4. According to ***IEEE: “Software reliability is defined as the probability of failure free software operation for a specified period of time in a specified environment.”***
5. Reliability of a software product denotes the twist-worthiness, software reliability refers to as the probability of the product working correctly over a given time duration.
6. Sometimes, it is very difficult to measure accurately the reliability of a software product because of following two reasons:

(i) Reliability is observer dependent. It depends upon the observer’s new point.

(ii) Reliability of a software product keeps changing as errors are detected and fixed. Because of frequent changing nature of software product, problems occur during the accurate measurement of the reliability of the software product.

8.3.1 Factors affecting Software Reliability

There are many factors which affects the growth of software reliability. These factors increases or decreases the growth of software reliability. These factors are as follows:

1. **Number of Errors:** Software reliability depends upon the number. of errors in the software product. Reliability is inversely proportional to the number of errors (defects) occur in the software product.

   $$\text{Reliability}\ \alpha \frac{1}{\text{No. of Errors}}$$

   Reliability increases when no. of errors decreases in a particular software product. Reliability decreases when no. of errors increases in a particular software product.
2. **Exact Location of Errors:** Software reliability increases when maintenance staff finds out the exact location of error. If exact location of error is finds out, then it becomes very easy for maintenance staff to remove that error and enhance the reliability and quality of a particular software product.
3. **How Product is used:** Software reliability also depends upon the way in which product is being used. If proper and correct way is used then it increases the reliability of a particular software product.
4. **Time:** Software reliability is not a function of time and different from hardware failure. With the passage of time, hardware components become old and obsolete. Hardware failure parts can be replaced with rate of hardware reliability increases. But in the case of software reliability, software cannot wear out. It can be upgraded intentionally by identifying and resolving errors and failure rate decreases. But when the software become obsolete, no more error correction occurs and failure rate remains unchanged.

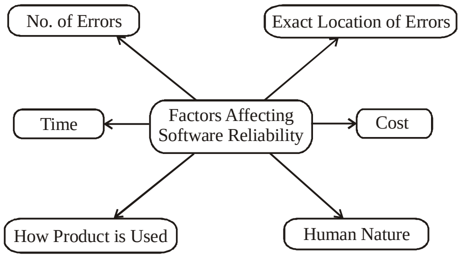


Fig. 8.10 Factors affecting Software Reliability

5. **Cost:** Software reliability depends on cost. Testing is a costly process. When more testing is performed on any software product to improve its reliability, cost automatically increases.
6. **Human Nature:** Software reliability also depends upon the human nature and observer's view point. Level of reliability various with the different nature of different people.

8.3.2 Comparison of Software Reliability and Hardware Reliability

Measuring the software reliability is more difficult as compared to measuring the hardware reliability. The difference between hardware reliability and software reliability are as follows:

Sr. No	Hardware Reliability	Software Reliability
1	Hardware reliability is concerned with stability.	Software reliability is concerned with reliability growth.
2	Hardware components fail because of wear and tear.	Software components fail because of bugs
3	To fix a hardware fault, one has to repair or replace the failed parts.	To fix a software fault, one has to fix the bug, code is changed to fix the bug.
4	When the hardware failure is repaired its reliability would be maintained at the level that existed before the failure occurred.	When the software failure is repaired, its reliability may increase or decrease.
5	Hardware reliability is a function of time and different from software failure.	Software reliability is not a function of time and different from hardware failure.

Table 8.1 Comparison of Software Reliability and Hardware Reliability

8.3.3 Uses of Reliability Studies

1. It estimate and predict the reliability behavior of software during its development and operation.
2. It answer the various question like what is the expected duration between successive failures.
3. It helps software engineering methodologies to develop the code.
4. It use formal methods during code development.
5. It compares the software engineering technologies in terms of cost and quality.
6. It aims at fault-free performance of software systems.
7. It concerns itself with how well the software functions to meet the requirements of the user.
8. It measures the progress of system testing.
9. It gives us a better insight into the development processes.

8.3.4 Software Reliability Metrics

Software reliability metrics are used to express quantitatively the reliability of the software product. No single metric is present which explains the reliability of all software products. Different software products may have different reliability requirements. The following are some reliability metrics:

1. **Rate of Occurrence of Failure (ROCOF):** It measures the frequency of the occurrences of failure. ROCOF can be obtained by observing the operational behavior of a software product over a specified time interval.

$$\text{ROCOF} = \frac{\text{Total number of observed failures}}{\text{Duration of observation}}$$

2. **Mean Time to Failure (MTTF):** MTTF is mean time between two successive failures. MTTF is measured by recording the failure data for n failures.

$$\text{MTTF} = \sum_{i-1}^{n} \frac{t_{i+1} - t_i}{(n-1)}$$

3. **Mean Time to Repair (MTTR):** After the occurrence of failure, some time is required to fix the error to solve that failure.

MTTF is the average time to track and fix the errors.

4. **Mean Time between Failure (MTBF):** MTBF metric is a combination of MTTF and MTTR

MTBF = MTTF + MTTR

5. **Availability (Avail):** It is a measure of how likely would the system be available for use over a given period of time. Availability is a measure of time during the system is available over long time periods.

$$\text{Availability} = \frac{\text{MTBF}}{(\text{MTBF} + \text{MTTR})}$$

8.3.5 Software Reliability Growth Models

A software reliability growth model is a mathematical model which helps in detection of errors, repair those errors and improves the software reliability. It also provides information when to stop testing to attain given reliability growth level.

The two software reliability growth models are as follows:

1. Jelinski and Moranda Model
2. Littlewood Model

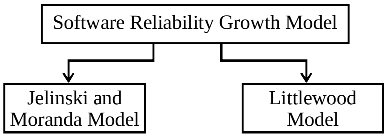


Fig. 8.11 Software Reliability Growth Models

1. **Jelinski and MorandaModel:**It is the simplest reliability growth model. This model assumes that reliability increases each time when an error is detected and repaired. Reliability is directly proportional to error detection and fixing.

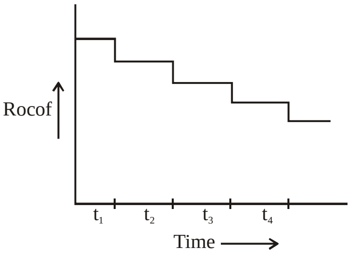


Fig. 8.12 Jelinski and Moranda Model

2. **LittlewoodModel:**It is the complex reliability growth model. This model assumes that there is a chance of introducing additional errors when an error is detected and repaired.

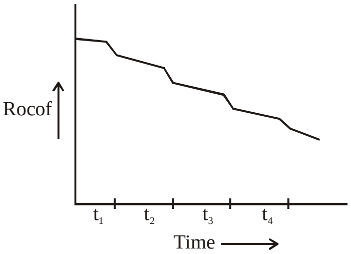


Fig. 8.13 Littlewood Model