FC-1-01T: Fundamentals of Computer

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 6
Pass Percentage: 40%

Course: Fundamentals of Computer	
Course Code: FC-1-01T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Learn the basic knowledge of computer hardware and software
CO2	Get basic knowledge of number system
CO3	Gain knowledge of computer languages such as machine language, assembly language, high level language, 4GL.
CO4	Learn hands on experience with operating systems
CO5	Learn the computer networks, Information Technology and Society

Detailed Contents:

Module No.	**Module Name**	**Module Contents**
Unit 1	**Introduction of Computer**	Characteristics of the Computer, Block diagram of a Computer, Classification and Generations of Computer, *Input Devices:* Keyboard, Mouse, Trackball, Space ball, Joystick, Light pen, Touch screen, Digitizer, Data Glove, Scanner, Speech Recognition Devices, Optical Recognition Devices: OMR, OBR, OCR, MICR, Video Cameras, *Output Devices:* Monitors, Printers and its types, Plotters and its types, Speakers, Multimedia Projector.
Module II	**Number System**	Concept of Bit and Byte, Number System: Binary, Octal, Decimal and Hexadecimal System, Conversion from one system to the other. Binary Arithmetic: Addition, subtraction and multiplication. 1's compliment, 2's compliment, Subtraction using 1's compliment and 2's compliment.
Module III	**Computer Languages**	Machine language, assembly language, high level language, 4GL. *Language Translators:* Compiler, Interpreter, and Assembler. *Software:* Types of Software: System Software, Application Software, and Firmware. *Memories:* Memory Hierarchy, Memory Types: Magnetic core, RAM, ROM, Secondary, Cache, Overview

30/1/24

		of storage devices: floppy disk, hard disk, compact disk, tape.
Module IV	**Operating System**	Functions of Operating System, Types of Operating System, Turning on a computer, desktop, taskbar, start menu, booting up, Desktop, Shortcut, Icons, Recycle Bin, Start Menu, My Computer, Computer's Devices and Drives, Storage, Removable Storage, CD/DVD Drive, floppy drive, and USB flash drive, Hard drive, Control Panel, The Window, Parts of Window, File Explorer, Files, Folders, Directories, Command, Menus, Keyboard, Function Keys, Normal Keys, Special keys, Direction keys, Numeric Keypad, Numeric Keys, Mouse: Left button, Right Button, Windows Accessories, Sharing Information between Programs. Virus, Antivirus, Peripherals can use with your computer.
Module V	**Computer Networks**	Components of data communication, modes of communication, standards and organizations, Network Classification, Network Topologies; Network Types, Transmission media, network protocol; layered network architecture. Basic of Computer networks: LAN, MAN, WAN.
Module VI	**Information Technology and Society**	Applications of Information Technology in Business and Industry, Railway, Airline, Entertainment, Banking, Insurance, Inventory Control, Hotel Management, Education and Training, Mobile Phones, Information Kiosks, Weather Forecasting, Scientific Application.

Books

1. Pradeep K., Sinha, "Computer Fundamentals: Concepts, Systems & Applications", BPB
2. Rajaraman V, "Fundamentals of Computers", PHI
3. RS Salaria, "Computer Fundamentals", Kanna Publication, 1st Ed., 2017
4. E Balagurusamy, "Fundamentals of Computers", Mc Graw Hill, 2012
5. Glenn Brookshear, "Computer Science: An Overview", Pearson, 2012

30/1/24

ISE-1-01T: Introduction to Software Engineering

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 6
Pass Percentage: 40%

Course: Introduction to Software Engineering	
Course Code:ISE-1-01T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Learn the basic knowledge of software engineering
CO2	Get basic knowledge of software process and project planning
CO3	Gain knowledge of software requirements analysis
CO4	Understands the detailed knowledge of software design and coding.
CO5	Understand the software testing and maintenance that is relevant to the industry.

Detailed Contents:

Module	Module Name	Module Contents
Module 1	**Introduction to 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.
Module II	**Software Process and Project Planning**	Characteristics of software process, Project management process, Software configuration management process. Project Planning: Activities, COCOMO model. Software Metrics – Definition, Importance, Categories of metrics. Software Quality – Attributes, Cyclomatic complexity metric.
Module III	**Software Requirements Analysis**	Need for SRS, Data flow diagrams, Data Dictionary, entity relationship diagram, Characteristics and components of SRS, validation, metrics.
Module IV	**Software Design and Coding**	Design principles, Module-level concepts, Structure Chart and Structured Design methodology, verification, metrics: network metrics, information flow metrics. Coding: Programming Principles and Guidelines, Verification- code inspections, static analysis.
Module V	**Software Testing**	Testing fundamentals, Black Box Testing: Equivalence class partitioning, Boundary value analysis, cause-effect graphing; White Box

30/1/24

		Testing: Control flow and Data flow based testing, mutation testing; levels of testing, test plan, test case specification, test case execution and analysis.
Module VI	**Software Maintenance**	Categories of maintenance. Software Reliability – Definition, uses of reliability studies

Books

1. Pankaj Jalote, "An Integrated approach to Software Engineering", 3rd Ed., Narosa Publications, 2005.
2. K.K. Aggarwal, Yogesh Singh, "Software Engineering", Revised 2nd Ed., New Age International Publishers.
3. Roger. S. Pressman, "Software Engineering – A Practitioner's Approach", 5th Ed., Tata McGraw Hill
4. Henry F. Korth, A. Silberschhatz, "Database Concepts", Tata McGraw Hill.
5. C. J. Date, "An Introduction to Database Systems", Pearson Education

30/1/24

CP-1-02T: Computer Programming

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 6
Pass Percentage: 40%

Course: Computer Programming	
Course Code: CP-1-02T	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Develop the ability to analyze problems, design algorithms, and implement solutions using C/C++ programming, showcasing proficiency in algorithmic problem-solving skills.
CO2	Implement and manipulate fundamental data structures such as arrays, linked lists, stacks, queues, trees, and hash tables in C/C++, demonstrating competence in choosing and utilizing appropriate data structures for different scenarios.
CO3	Gain expertise in handling exceptions, debugging C/C++ code, and implementing error-handling strategies to create robust and reliable programs.
CO4	Understand and apply principles of multithreading and concurrency in C/C++, including synchronization mechanisms, thread communication, and concurrent programming, showcasing the ability to develop efficient and responsive applications.
CO5	Familiarize oneself with common C++ frameworks gaining an understanding of how frameworks can streamline development and improve code organization and maintainability.

Detailed Contents:

Module No.	Module Name	Module Contents
Module I	**Problem Solving with Computers**	**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, Operators in C: Types of Operator: Arithmetic, Relational, Logical, Comma, Conditional, Assignment, Operator Precedence and Associativity in C, Input and Output Statements, Assignment statements.
Module II	**Control Structure**	**Control Structure:** 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
Module III	**Arrays and Pointers in C**	**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.
Module IV	**Functions**	**Functions:** 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.
Module V	**Strings and User Defined Data Types**	**Strings:** C Strings, Difference between char array and string literal, Traversing String, Accepting string as the input, Pointers with strings, String Functions **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.
Module VI	**Object Oriented Programming**	**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.

Books

1. E. Balagurusamy, "Programming in C", Tata McGraw Hill.
2. Kamthane, "Programming with ANSI and Turbo C", Pearson Education
3. Rajaraman,V, "Fundamentals of Computers", PHI
4. Kanetkar, "Let Us C", BPB Publications.
5. Herbert Schildt, "The Complete Reference C++", Tata McGraw-Hill.
6. Deiteland Deitel, "C++ How to Program", Pearson Education.
7. Robert Lafore, "Object Oriented Programming in C++", Galgotia Publications.
8. Bjarne Strautrup, "The C++ Programming Language", Addition-Wesley Publication Co.
9. Stanley B. Lippman, Josee Lajoie, "C++ Primer", Pearson Education.
10. E. Balagurusamy, "Object Oriented Programming with C++", Tata McGraw-Hill

CP-1-02P: Computer Programming Lab

Total Marks: 50
External Marks: 35
Internal Marks: 15
Credits: 2
Pass Percentage: 40%

Course: Computer Programming Lab	
Course Code: CP-1-02P	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Demonstrate proficiency in C/C++ programming by successfully designing, coding, and debugging Java applications to solve a variety of programming problems.
CO2	Implement and manipulate fundamental data structures, such as arrays, linked lists, stacks, and queues, showcasing the ability to choose and apply appropriate data structures based on problem requirements.
CO3	Apply object-oriented programming principles effectively, demonstrating the ability to design and implement classes, encapsulate data, utilize inheritance and polymorphism, and create reusable and modular code.
CO4	Develop strong algorithmic problem-solving skills by designing and implementing efficient algorithms to solve programming challenges, demonstrating an understanding of algorithm complexity and optimization.
CO5	Gain competence in error handling and debugging C/C++ code, utilizing debugging tools and techniques to identify and resolve errors effectively, thereby producing robust and error-free programs.

Detailed List of Programs:

Programme No.	Name of Program
P1	Write a simple program that prints "Hello, World!" to the console.
P2	Take two numbers as input and display their sum.
P3	Generate and print the multiplication table for a given number.
P4	Compute the factorial of a given number.
P5	Check whether a given number is prime or not.
P6	Generate and display the Fibonacci series up to a specified number of terms.
P7	Determine if a given number or string is a palindrome.
P8	Reverse a given string without using library functions.

P9	Implement a sorting algorithm (e.g., bubble sort, selection sort) for an array of integers.
P10	Search for an element in an array using linear search.
P11	Implement binary search for a sorted array.
P12	Perform addition of two matrices.
P13	Find and display the transpose of a matrix.
P14	Implement a program to calculate the power of a number using recursion.
P15	Create a basic calculator program that performs addition, subtraction, multiplication, and division.
P16	Compute the factorial of a number using a recursive function.
P17	Check whether a given number is an Armstrong number.
P18	Calculate the GCD of two numbers using Euclidean Algorithm.
P19	Convert a decimal number to its binary equivalent.
P20	Reverse the words in a given sentence without using library functions.

DBMS-2-01T: Data Base Management System (DBMS)

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 6
Pass Percentage: 40%

Course: Data Base Management System (DBMS)	
Course Code: DBMS-2-01T	
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 to solve problems

Detailed Contents:

Module No.	Module Name	Module Contents
Module 1	**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. 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.
Module II	**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. Relational Data Model: Concepts and

		Terminology. Constraints: Integrity Constraints, Entity and Referential Integrity constraints, Keys,
Module III	**Relational Algebra & Relational Calculus**	**Relational Algebra:** Basic Operators, Additional Operators. **Relational Calculus:** Tuple Relational Calculus and Domain Relational Calculus, Difference between relational algebra and relational calculus.
Module IV	**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.
Module V	**Transaction Management & Concurrency Control**	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.
Module VI	**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.

Books

1. Elmasry Navathe, “Fundamentals of Database System”, Pearson Education.
2. James Groff, Paul Weinberg, Andy Oppel, “Oracle SQL Complete Reference”, Tata McGraw-Hill.
3. T.Connolly, C Begg,“Database Systems”, Pearson Education.
4. Jeffrey D. Ullman, “Principles of Database Systems”, Galgotia Publications.
5. Henry F. Korth, A. Silberschhatz, “Database Concepts”, Tata McGraw Hill.
6. C. J. Date, "An Introduction to Database Systems”, Pearson Education

DBMS-2-01P: Data Base Management System (DBMS) Lab

Total Marks: 50
External Marks: 35
Internal Marks: 15
Credits: 2
Pass Percentage: 40%

Course: Data Base Management System (DBMS) Lab	
Course Code: DBMS-2-01P	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Implement Basic DDL, DML and DCL commands.
CO2	Understand Data selection and operators used in queries and restrict data retrieval and control the display order.
CO3	Use Aggregate and group functions to summarize data.
CO4	Join multiple tables using different types of joins.
CO5	Implementation of different types of operators in SQL

Detailed List of Programs:

Programme No.	Name of Program
P1	Implementation of DDL commands of SQL with suitable examples • Create table • Alter table • Drop Table
P2	Implementation of DML commands of SQL with suitable examples • Insert • Update • Delete
P3	Implementation of different types of function with suitable examples • Number function • Aggregate Function • Character Function • Conversion Function • Date Function
P4	Implementation of different types of operators in SQL • Arithmetic Operators • Logical Operators • Comparison Operator • Special Operator • Set Operation

30/1/24

P5	Implementation of different types of Joins ● Inner Join ● Outer Join ● Natural Join etc.
P6	Implementation of ● Group by & having clause ● Order by clause ● Indexing
P7	Implementation of ● Sub queries ● Views
P8	Study & Implementation of different types of constraints.
P9	Study & Implementation of Database Backup & Recovery commands. Study & Implementation of Rollback, Commit, Savepoint.
P10	Creating Database /Table Space ● Managing Users: Create User, Delete User ● Managing roles:-Grant, Revoke

30/1/24

DSDP-2-02T: Data Structures

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

Course: Data Structures	
Course Code: DSDP-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	Understand non-linear data structures like trees and graphs.
CO3	Apply stack for evaluation of arithmetic expressions, and conversion from infix to post fix and recursion.
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.

Detailed Contents:

Module	Module Name	Module Contents
Module 1	**Introduction to Data Structures**	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 off algorithms and Big O notation.
Module II	**Arrays and Linked Lists**	**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. **Linked List:** Introduction; operation on linked list, circular linked list, doubly linked list, header linked list, implementation of linked list, application of linked lists.
Module III	**Stacks & Queues**	**Stacks:** Introduction; array representation of stacks, Operation on stacks; Linked representation of stacks, Application of stacks: matching parenthesis, evaluation of arithmetic expressions, and conversion from infix to post fix, recursion. **Queues:** Introduction, operation on queues,

30/1/24

		linked representation of queue, Applications of queues, circular queue, memory representation of queues, priority queues, Multiple queues.
Module IV	**Trees, Graphs and searching & Sorting Algorithms**	**Trees:** Introduction; Binary Tree; Complete Binary Trees, representation of binary trees in the memory, traversing a binary tree, Binary Search Tree, Operations on Binary Search Tree. **Graphs:** Introduction Graph: Graph terminology, Memory Representation of Graphs: adjacency matrix representation of graphs, adjacency list or linked representation of graphs, graph traversal algorithms. **Searching & Sorting Algorithms:** Linear Search, Binary Search, Bubble Sort, Selection Sort, Insertion Sort

Books

1. A. Tanenbaum, Y. Lanhgsamand A. J. Augenstein, "Data Structures Using C", PHI.
2. Loomis, Marry, "Data Management and File Structures", PHI
3. Seymour Lipschultz, "Theory and Practice of Data Structures", Tata McGraw-Hill.
4. E. Horowitz and S. Sahni, "Data Structures with Pascal", Galgotia.
5. M. J. Folk, B. Zoellick, G Riccardi, "File Structures", Pearson Education.

30/1/24

DSDP-2-02P: Data Structures Lab

Total Marks: 50
External Marks: 35
Internal Marks: 15
Credits: 2
Pass Percentage: 40%

Course: Data Structures Lab	
Course Code: DSDP-2-02P	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Implement basic data structures such as arrays and linked list.
CO2	Develop programs to demonstrate fundamental algorithmic problems including Tree Traversals and Graph traversals.
CO3	Implement various searching and sorting algorithms.
CO4	Develop programs to demonstrate the implementation of various operations on stack
CO5	Develop programs to demonstrate the implementation of various operations on queue

Detailed List of Programmes:

Programme No.	**Name of Programme**
P1	WAP to demonstrate the concept of one dimensional array finding the sum of array elements.
P2	WAP to insert an element in an array
P3	WAP to delete an element from an array.
P4	WAP to insert an element at the beginning of a linked list
P5	WAP to insert an element at the end of a linked list
P6	WAP to insert an element within the linked list
P7	WAP to demonstrate PUSH and POP operations of stack using array
P8	WAP to demonstrate the implementation of queue using linked list
P9	WAP to search an element from an linear array using linear search.
P10	WAP to Search an element using binary search.
P11	WAP to arrange the list of numbers in ascending order using Bubble Sort.
P12	WAP to arrange the list of numbers in ascending order using Insertion Sort.

30/1/24

P13	WAP to arrange the list of numbers in ascending order using Selection Sort.
P14	WAP to demonstrate the operation of Pre order Traversing technique of Tree
	WAP to demonstrate the operation of Post order Traversing technique of Tree
P15	WAP to implement a simple Depth-First Search (DFS) traversal in Graph .
P16	WAP to implement a simple Breadth-First Search (BFS) traversal in Graph.

30/1/24

DSDP-2-03P: Project

Total Marks: 100
External Marks: 70
Internal Marks: 30
Credits: 6
Pass Percentage: 40%

Course: Project	
Course Code: DSDP-2-03P	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Demonstrate a sound technical knowledge in the field of Software Development and Programming.
CO2	Gain ability to identify research gaps through literature survey, problem identification, formulation and solution.
CO3	Design solutions to problems utilizing a systems approach.
CO4	Gain ability of communication, management, leadership and entrepreneurship skills.
CO5	Obtain capability and enthusiasm for self-improvement through continuous professional development and life-long learning

Description

To develop project in the field of Software Development and Programming

Study projects can be individual or team projects. Team projects are limited to a maximum number of 3 students (and should be defined according to the complexity of the study).

At the beginning of the 2nd semester (deadline 15 March for Jan Admission Cycle and 15 October for July Admission Cycle) every student /group has to submit his/their application for the study project to the Programme Coordinator for the approval of topic and team members. Within 15 days after approval the topic, the students have to write a project synopsis. The project synopsis should follow a scientific structure and consists basically of the following parts:

1. INTRODUCTION (1 PARAGRAPH)
2 LITERATURE SURVEY (2-3 pages)

(reviews of 4-5 papers/journals/articles/techniques/wares/etc)

3. RESEARCH GAPS (1 paragraph)

4. PROBLEM FORMULATION (1 paragraph)

5. OBJECTIVES OFPROJECT

30/1/24

The major aim of this project is to ……………………….. To achieve the major aim, following objectives are proposed for the study

1.

2.

3.

6. METHODOLOGY/PLANNING OF WORK (1 page)

Research type, unit, methods, tools of data collection / analysis. Methodology will include the steps to be followed to achieve the objective of the project during the project development.

7. H/W AND S/W REQUIREMENTS (1 paragraph)

Software/Hardware required for the development of the project

8. EXPECTED OUTCOMES (1 paragraph)

At the end of the study project, the students have to write a project report. The project report should follow a scientific structure and consists basically of the following parts:

1. INTRODUCTION

2 LITERATURE SURVEY

3. PROBLEM FORMULATION

4. OBJECTIVES OF PROJECT WORK

The major aim of this project is to ……………………….. To achieve the major aim, following objectives are proposed for the study

1.

2.

3.

5. METHODOLOGY

6. MAJOR FINDINGS

7. CONCLUSIONS & FUTURE SCOPE

The length of the final project report should be about 30-40 A4 pages (about 9000-12000 words).

The due date for the final version of the report is at least one week before the official presentation. The duration of the presentation is 10-15 min with an additional 10 min discussion in English.

30/1/24