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

Bachelor of Computer Applications (BCA)
Course Name: Computer Programming Lab
Course Code: BCA-1-01P

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)

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

PROGRAMME CO-COORDINATOR :
Dr. Gaurav Dhiman
Assistant Professor, School of Sciences and Emerging Technologies
Jagat Guru Nanak Dev Punjab State Open University, Patiala

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

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

Bachelor of Computer Applications (BCA)
Semester I
BCA-1-01P: Computer Programming Lab

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

Course: Computer Programming Lab	
Course Code: BCA-1-01P	
Course Outcomes (COs) After the completion of this course, the students will be able to:	
CO1	Develop C programs to solve simple mathematical and decision making problems.
CO2	Develop, Debug and Execute programs to demonstrate the applications of arrays in C
CO3	Develop, Debug and Execute programs to demonstrate decision making and looping constructs in C
CO4	Develop, Debug and Execute programs to demonstrate the basic concepts of pointers in C
CO5	Implement programs to read from and write to files using C, including concepts such as file pointers and file I/O operations.

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.
P21	WAP that swaps the values of two numbers using pointers.
P22	WAP that reverses an array using pointers.
P23	WAP hat passes an array to a function and calculates the sum of its elements.
P24	WAP that reads data from a file and prints it to the console.
P25	WAP that appends data to an existing file.

Table of Contents


```
*
**
***
****
*****
```

```

```

```
*****
```

```
****
***
**
*
```

```
    *
   * *
  * * *
 * * * *
* * * * *
```

```
    *
   * *
  * * *
 * * * *
* * * * *
* * * * *
 * * * *
  * * *
   * *
    *
```

```
**********
****  ****
***    ***
**      **
*        *
```

```
    0
   01
  010
 0101
```

```
    01010
```

```
0       0
    01      01
    010    010
    0101  0101
    0101001010
```

```
    12345
    1234
    123
    12
    1
```

```
        1
       123
      12345
     1234567
    123456789
```

```
    1        1
    12      21
    123    321
    1234  4321
    1234554321
```

Practice Exercise:

- Write a C program to find area of circle.
- Write a C program to find Simple Interest.
- Write a C program to perform Arithmetic operations.
- Write a C program to find area of a triangle using three sides
- Write a C program to find the largest of two numbers.
- Write a C program to find the largest of five number.
- Write a C program to find smallest of two numbers using conditional operator.
- Write a C program to perform the Sum of N natural numbers.
- Write a C program to perform factorial of a given number.
- Write a C program to find the given number is palindrome or not.
- Write a C program to find the given number is Armstrong or not.
- Write a C program to reverse the given number.
- Write a C program to find Sum of digits.
- Write a C program to find day of the week.
- Write a C program to find the given character is vowel or not.
- Write a C program to display “This is my first C Program”.
- Write a C program to add two numbers (2 and 6) and display its sum.
- Write a C program to multiply two numbers (4 and 5) and display its product.
- Write a C program to calculate area and circumference of a circle.
- Write a C program to calculate simple and compound interest.
- Write a C program to swap values of two variables with and without using third variable.
- Write a C program to display the size of every data type using “sizeof” operator.
- Write a C program to input two numbers and display the maximum number.
- Write a C program to find the largest of three numbers using ternary operators.
- Write a C program to find the roots of quadratic equation.

- Write a C program to perform addition, subtraction, division and multiplication of two numbers.
- Write a C program to illustrate the use of unary prefix and postfix increment and decrement operators.

Exercise 1: Solutions

1. C Hello World Program

```
#include <stdio.h>

int main()

{

 printf(“Hello, World!\n”);

 return 0;

}
```

2. C Program to Print Your Own Name

```
#include <stdio.h>

int main() {

 printf(“Your Name\n”);

 return 0;

}
```

3. C Program to Print an Integer Entered By the User

```
#include <stdio.h>

int main() {

 int num;

 printf(“Enter an integer: “);
```

```
 scanf(“%d”, &num);
 printf(“You entered: %d\n”, num);
 return 0;
}
```

4. C Program to Check Whether a Number is Prime or Not

```
#include <stdio.h>
int main() {
 int num, i, flag = 0;
 printf(“Enter a number: “);
 scanf(“%d”, &num);
 for (i = 2; i <= num / 2; ++i) {
 if (num % i == 0) {
 flag = 1;
 break;
 }
 }
 if (flag == 0)
 printf(“%d is a prime number.\n”, num);
 else
 printf(“%d is not a prime number.\n”, num);
 return 0;
 }
```

Exercise 2: Solutions

1. C Program to Multiply two Floating-Point Numbers

```
#include <stdio.h>
int main() {
```

```
 float num1, num2, product;
 printf("Enter two floating-point numbers: ");
 scanf("%f %f", &num1, &num2);
 product = num1 * num2;
 printf("Product: %f\n", product);
 return 0;
}
```

2. C Program to Print the ASCII Value of a Character

```
#include <stdio.h>
int main() {
 char ch;
 printf("Enter a character: ");
 scanf("%c", &ch);
 printf("ASCII value of %c = %d\n", ch, ch);
 return 0;
}
```

3. C Program to Swap Two Numbers

```
#include <stdio.h>
int main() {
 int num1, num2, temp;
 printf("Enter two numbers: ");
 scanf("%d %d", &num1, &num2);
 temp = num1;
 num1 = num2;
 num2 = temp;
 printf("After swapping: num1 = %d, num2 = %d\n", num1, num2);
```

```
 return 0;
}
```

4. C Program to Calculate Fahrenheit to Celsius

```
#include <stdio.h>
int main() {
 float fahrenheit, celsius;
 printf(“Enter temperature in Fahrenheit: “);
 scanf(“%f”, &fahrenheit);
 celsius = (fahrenheit – 32) * 5 / 9;
 printf(“Temperature in Celsius: %f\n”, celsius);
 return 0;
}
```

5. C Program to Find the Size of int, float, double, and char

```
#include <stdio.h>
int main() {
 printf(“Size of int: %d bytes\n”, sizeof(int));
 printf(“Size of float: %d bytes\n”, sizeof(float));
 printf(“Size of double: %d bytes\n”, sizeof(double));
 printf(“Size of char: %d byte\n”, sizeof(char));
 return 0;
}
```

6. C Program to Print Prime Numbers From 1 to N

```
#include <stdio.h>
int main() {
 int i, j, n;
 printf(“Enter a number (N): “);
```

```
 scanf(“%d”, &n);
 printf(“Prime numbers between 1 and %d are: “, n);
 for (i = 2; i <= n; ++i) {
 int isPrime = 1;
 for (j = 2; j <= i / 2; ++j) {
 if (i % j == 0) {
 isPrime = 0;
 break;
 }
 }
 if (isPrime)
 printf(“%d “, i);
 }
 return 0;
}
```

Exercise 3: Solutions

1. C Program to Check Whether a Number is Positive, Negative, or Zero

```
#include <stdio.h>
int main() {
 int num;
 printf(“Enter a number: “);
 scanf(“%d”, &num);
 if (num > 0)
 printf(“Positive number\n”);
 else if (num < 0)
 printf(“Negative number\n”);
 else
 printf(“Zero\n”);
 return 0;
}
```

Output:

```
Enter a number: 7
Positive number
```

2. C Program to Check Whether Number is Even or Odd

```
#include <stdio.h>
int main() {
 int num;
 printf(“Enter a number: “);
 scanf(“%d”, &num);
 if (num % 2 == 0)
 printf(“Even number\n”);
```

```
 else
 printf(“Odd number\n”);
 return 0;
}
```

Output:

```
Enter a number: 15
Odd number
```

3. C Program to Calculate Sum of Natural Numbers

```
#include <stdio.h>
int main() {
 int n, sum = 0;
 printf(“Enter a positive integer: “);
 scanf(“%d”, &n);
 for (int i = 1; i <= n; ++i) {
 sum += i;
 }
 printf(“Sum of natural numbers from 1 to %d: %d\n”, n, sum);
 return 0;
}
```

Output:

```
Enter a positive integer: 5
Sum of natural numbers from 1 to 5: 15
```

4. C Program to Print Alphabets From A to Z Using Loop

```
#include <stdio.h>
int main() {
 char ch;
```

```
 printf(“Alphabets from A to Z:\n”);
 for (ch = ‘A’; ch <= ‘Z’; ++ch) {
 printf(“%c “, ch);
 }
 return 0;
}
```

Output:

Alphabets from A to Z:

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

Exercise 4: Pattern Printing in C

SOLUTION: 1. PATTERN PROGRAMS IN C

Half, Full, Incremented and Decrement **Stars Series**, **Pyramid Pattern programs**

Program - 1

```
/*C program to print following Pyramid:
*
**
***
****
*****
*/

#include<stdio.h>

#define MAX 5

int main()
{
    int i,j;

    for(i=0; i< MAX; i++)
    {
        for(j=0;j<=i;j++)
        {
            printf("*");
        }
        printf("\n");
    }
    return 0;
}
```

Output

```
    *
```

```
    **
    ***
    ****
    *****
```

Program - 2

```c
/*C program to print following Pyramid:
*****
****
***
**
*
*/

#include<stdio.h>

#define MAX   5

int main()
{
       int i,j;

       for(i=MAX; i>=0; i--)
       {
              for(j=0;j<=i;j++)
              {
                     printf("*");
              }
              printf("\n");
       }
    return 0;
}
```

Output

```
    *****
    ****
```

```
    ***
    **
    *
```

Program - 3

```c
/*C program to print following Pyramid:
        *
       * *
      * * *
     * * * *
    * * * * *
*/

#include<stdio.h>

#define MAX  5

int main()
{
    int i,j;
    int space=4;
    /*run loop (parent loop) till number of rows*/
    for(i=0;i< MAX;i++)
    {
            /*loop for initially space, before star printing*/
            for(j=0;j< space;j++)
            {
                    printf(" ");
            }
            for(j=0;j<=i;j++)
            {
                    printf("* ");
            }

            printf("\n");
            space--;  /* decrement one space after one row*/
    }
```

```
    return 0;
}
```

Output

```
          *
        * *
       * * *
      * * * *
     * * * * *
```

Program - 4

```
/*C program to print following Pyramid:
          *
        * *
       * * *
      * * * *
     * * * * *
     * * * * *
      * * * *
       * * *
        * *
          *
*/

#include<stdio.h>

#define MAX   5

int main()
{
       int i,j;
       int space=4;
       /*run loop (parent loop) till number of rows*/
       for(i=0;i< MAX;i++)
       {
              /*loop for initially space, before star printing*/
```

```
            for(j=0;j< space;j++)
            {
                  printf(" ");
            }
            for(j=0;j<=i;j++)
            {
                  printf("* ");
            }

            printf("\n");
            space--;
      }
      /*repeat it again*/
      space=0;
      /*run loop (parent loop) till number of rows*/
      for(i=MAX;i>0;i--)
      {
            /*loop for initially space, before star printing*/
            for(j=0;j< space;j++)
            {
                  printf(" ");
            }
            for(j=0;j< i;j++)
            {
                  printf("* ");
            }

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

Output

```
          *
         * *
        * * *
```

```
     * * * *
    * * * * *
    * * * * *
     * * * *
      * * *
       * *
        *
```

Program - 5

```c
/*C program to print following Pyramid:
    **********
    ****  ****
    ***    ***
    **      **
    *        *
*/

#include<stdio.h>

#define MAX  5

int main()
{
       int i,j;
       int space=0;
       /*run loop (parent loop) till number of rows*/
       for(i=MAX;i>0;i--)
       {
              /*print first set of stars*/
              for(j=0;j< i;j++)
              {
                     printf("*");
              }
              for(j=0;j< space;j++)
              {
                     printf(" ");
              }
```

```
            /*print second set of stars*/
            for(j=0;j< i;j++)
            {
                  printf("*");
            }

            printf("\n");
            space+=2;
      }
    return 0;
}
```

Output

```
**********
****  ****
***    ***
**      **
*        *
```

NUMBER PYRAMID PROGRAMS - HALF, FULL INCREMENTED AND DECREMENT SERIES PROGRAMS

Program - 6

```
/*C program to print following Pyramid:
        0
       01
      010
     0101
    01010
*/

#include<stdio.h>

int main()
{
      int i,j,k;
```

```
        for(i=0 ; i<=4 ; i++)
        {
                for(j=4 ; j>i ; j--)
                        printf(" ");

                for(k=0 ; k<=i; k++)
                {
                        if(k%2==0)
                                printf("0");
                        else
                                printf("1");
                }
                printf("\n");
        }

    return 0;
}
```

Output

```
        0
       01
      010
     0101
    01010
```

Program - 7

```
/*C program to print following Pyramid:
    0         0
    01       01
    010     010
    0101   0101
    0101001010
*/

#include<stdio.h>
```

```c
int main()
{
	int i,j,k,l=8,m,n,o,p;
	for(i=0; i<=4; i++)
	{
		for(j=0; j<=i; j++)
		{
			if(j%2==0)
				printf("0");
			else
				printf("1");
		}
		for(k=1; k<=l; k++)
		{
			printf(" ");
		}
			l = l-2;
		for(m=0; m<=i; m++)
		{
			if(m%2==0)
			printf("0");
			else
			printf("1");
		}

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

Output

```
0         0
01       01
010     010
0101   0101
0101001010
```

Program - 8

```
/*C program to print following Pyramid:
    12345
    1234
    123
    12
    1
*/

#include<stdio.h>

int main()
{
       int i,j;
       for(i=5; i>=1; i--)
       {
              for(j=1; j<=i; j++)
              {
                     printf("%d", j);
              }
           printf("\n");
       }
    return 0;
}
```

Output

```
    12345
    1234
    123
    12
    1
```

Program - 9

```
/*C program to print following Pyramid:
```

```
        1
       123
      12345
     1234567
    123456789
*/

#include<stdio.h>

int main()
{
       int i,j,k,l=1;
       for(i=1; i<=5; i++)
       {
              for(j=4; j>=i; j--)
              {
                     printf(" ");
              }

              for(k=1; k<=l; k++)
              {
                     printf("%d",k);
              }
                     l = l+2;
           printf("\n");
       }
    return 0;
}
```

Output

```
        1
       123
      12345
     1234567
    123456789
```

Program - 10

```
/*C program to print following Pyramid:
    1         1
    12       21
    123     321
    1234   4321
    1234554321
*/

#include<stdio.h>

int main()
{
       int i,j,k,l,m=8,n=1;
       for(i=1; i<=5; i++)
       {
              for(j=1; j<=i; j++)
              {
                     printf("%d",j);
              }
              for(k=m; k>=1; k--)
              {
                     printf(" ");
              }
                     m = m-2;
              for(l=n; l>=1; l--)
              {
                     printf("%d",l);
         }
              n = n+1;
           printf("\n");
       }

       return 0;
}
```

Output

```
1         1
12       21
123     321
1234   4321
1234554321
```