

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.