Lab Sheet-08

Functions and OOPs Concepts in C++

BCA I Semester (Session 2025-2026)

Objective:

- To be familiar with the syntax and structure of C++ programming.

- To learn problem-solving techniques using C++

Requirements:

- C++ programming environment

- Text editor or IDE

Program List:

1. Write a C++ function to find a prime number.

2. Write a C++ function to print the Fibonacci series.

3. Write a C++ function to apply the bubble shot technique on an array.

4. Write a C++ function to copy a string from another string.

5. Write a C++ function to find out the number of words in a string.

6. Write a C++ function to search an element using a binary search technique.

7. Write a C++ function to find the sum of two matrix.

8. Write a C++ function to find out the sum of diagonal numbers in matrix.

9. Write a C++ function to swap two numbers using call-by-reference properties.

10. Write a program to define a class Student with data members name, roll_no, and marks. Use member functions to input and display the details of the student.

11. Write a program to define a class Car with members brand and price. Create multiple objects of the class and display their information.

12. Write a program to demonstrate encapsulation using a class Bank Account. Keep balance as a private data member and provide public functions deposit () and withdraw () to modify it safely.

13. Write a program to demonstrate single inheritance. Create a base class Person with name and age, and a derived class Student with marks.

Display all details using functions.

14. Write a program to demonstrate function overloading (compile-time polymorphism). Create a class Calculator with overloaded functions add (int, int) and add (double, double) to perform addition.

15. Write a program to demonstrate abstraction using an abstract class Shape. Define a pure virtual function area (), and create derived classes Circle and Rectangle to calculate and display their areas.

Instructions:

- Write comment to make your programs readable.
- Use descriptive variables in your programs (Name of the variables should show their purposes)

Guidelines:

- Use Clear and Consistent Syntax
- Manage Variables and Data Types Correctly
- Handle Input/Output Effectively

Submission:

- Submit the source code files (.cpp) for each task along with a brief report documenting the implementation details and the results of the executions.