**OBJECT ORIENTED PROGRAMMING WITH C++**

*Submitted in partial fulfilment of the requirements for the degree of BCA*

***Submitted By***

**[Your Name]**

**[Enrolment Number]**

***Submitted to***

**[Name of Subject Faculty]**

****

**Department of CS & IT**

**Avviare Educational Hub**

**Noida, UP – 201301, India**

**PRACTICAL LAB FILE**

**OBJECT ORIENTED PROGRAMMING WITH C++**

|  |  |  |
| --- | --- | --- |
| **NAME** | **:** |  |
| **ENROLLMENT NO** | **:** |  |
| **SEMESTER/SECTION** | **:** |  |
| **BRANCH** | **:** |  |
| **BATCH** | **:** |  |

**INDEX**

|  |  |  |  |
| --- | --- | --- | --- |
| **S. No.** | **Contents** | **Page No.** | **Faculty Signature** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**GUIDELINES FOR SUBMISSION OF OOP C++ LAB MANUAL**

1. The lab manual hardcopy must be submitted by **20th November 2024 (Wednesday)** to your respective class coordinator. Late submissions may not be accepted unless prior approval is given.
   * The file should be **handwritten** only. No typed or printed submissions are permitted.
   * Use A4 size sheets only with one side lined and other side plain sheet to write or paste the output.
2. **Cover Page: Include essential information on the cover page:**
   * Name, Enrolment Number, Semester, Section and Branch
   * Submitted By/ Submitted To
3. **Content Page:** Fill the table of contents with page numbers for each lab question and answer.
4. **Question and Answer Layout:** Clearly write each question before providing the answer, and start each new question on a fresh page.
5. **Manual Structure**
6. For printed submissions, include screenshots of outputs if the question requires program execution. Handwritten submissions can include drawn representations of the expected output.
7. Securely **bind** the lab manual for easy handling. Spiral binding is preferred.

**Practical Questions**

**CHAPTER 1**

Q1- Implement a program that calculates the factorial of a given number using a for loop. Take the number as input from the user and display the factorial result.

Q2- Write a C++ program to create a function that converts Celsius to Fahrenheit and returns the result. Demonstrate the use of type conversions in the calculation.

Q3- Write a program to display a table of numbers from 1 to 10 with their squares and cubes, aligning the output using the setw manipulator.

**CHAPTER 2**

Q1- Write a class Rectangle with a constructor that accepts length and width as parameters and a destructor that outputs a message. Calculate and display the area of the rectangle.

Q2- Implement a class Book that has a copy constructor. Initialize one Book object and create another by copying the first one. Display the details of both objects to verify the copy constructor’s functionality.

Q3- Write a class Employee that has a static data member count to keep track of the number of employees created. Display the count each time a new employee object is created.

**CHAPTER 3**

Q1- Create a Matrix class that includes a two-dimensional array as a data member. Implement member functions to input and display matrix elements.

Q2- Write a program to use the C++ Standard Library string class. Take a string as input, convert it to uppercase, and display the length of the string.

Q3- Write a C++ program to open a text file in ofstream mode, write some text, and close it. Reopen it in ifstream mode to read and display the text on the console.

**CHAPTER 4**

Q1 -Write a program that overloads a function add() to handle the addition of two integers, two floating-point numbers, and two strings.

Q2 -Implement a Complex class to represent complex numbers. Overload the + operator to add two complex numbers.

Q3 – Write down the difference between Compile time and Runtime Polymorphism. Explain its types with the programmatic examples of each.

**CHAPTER 5**

Q1- Create a base class Person with attributes like name and age. Create a derived class Student that adds a grade attribute. Demonstrate single inheritance by creating a Student object.

Q2- Write a program that demonstrates multilevel inheritance by creating a base class Shape, a derived class Rectangle, and another derived class ColoredRectangle that adds a color attribute.

Q3- Create a base class Animal with a virtual function sound(). Derive classes Dog and Cat that override the sound() function. Create objects of Dog and Cat using Animal pointers and call sound().

Q4- Write a class Box and implement a friend function that calculates and returns the volume of the box.