Lab Assignment 1

1. **Classes and Objects**

Implement a system that performs addition and multiplication of two Complex numbers using a class Complex which has real-part and imaginary-part as private data members and add and mul as two public member functions to add and multiply two complex numbers. Also use two member functions to read and display the complex numbers.

2. **Constructors and Destructors**

Create a class Rectangle. It has two private data members to hold the length and the breadth, and one function is to calculate the area of a rectangle. Calculate the area for three rectangles with three sets of values. Use default constructor, parameterized constructor, default copy constructor, and a destructor.

3. **Counter Problem**

Create a class called Counter. The counter class has one private data member to hold the counter value. It has three member functions. Init to initialize the data member, count to increment the counter whenever needed and display to display the value of the counter. The counter works until the user wants to count. Design a menu-driven program.

4. **Function Overloading**

Create a class called solid which will store the volume (float) of the solid in a private section, and four overloaded member functions that calculate the volume for four different solids like a cube(side-integer), sphere(radius-float), cylinder (radius-float, height-float), and rectangular box (length, breadth, and height all are integers) in the public section. Calculate the volume of a cube, sphere, cylinder, and rectangular box using only one object and the overloaded member function. Use a display function to display the volume of the respective solid.

5. **Toll Booth Program**

Imagine a tollbooth at a bridge. Cars passing by the booth are expected to pay a 50-cent toll. Mostly they do, but sometimes a car goes by without paying. The tollbooth keeps track of the number of cars that have gone by, and of the total amount of money collected. Model this tollbooth with a class called tollBooth. The two data items are a type unsigned int to hold the total number of cars, and a type of double to hold the total amount of money collected. A constructor initializes both to 0. A member function called paying Car () increments the car total and adds 0.50 to the cash total.

Another function, called nonpayCar(), increments the car total but adds nothing to the cash total. Finally, a member function called display () displays the two totals. Make appropriate member functions const. Include a program to test this class. This program should allow the user to push one key to count a paying car, and another to count a non-paying car. Pushing the [Escape] key should cause the program to print out the total cars, total cash, and then exit.

6. **Addition of times**

Create a class called time that has separated integer member data for hours, minutes and seconds. One constructor should initialize this data to 0, and another should initialize it to fixed values. Another member

function should display it, in 11:59:59 format. The final member function should add two objects of type time passed as arguments.

A main() program should create two initialized time objects and one that is not initialized. Then it should add the two initialized values together, leaving the result in the third time variable. Finally, it should display the value of this third variable. Make appropriate member functions const.

7. Inheritance

Assume that a bank maintains two kinds of accounts for customers, one called as savings account and other as current account. The savings account provides compound interest and withdrawal facilities but no cheque book facility. The current account provides cheque book facility but no interest. Current account holders should also maintain a minimum balance and if the balance falls below this level, a service charge is imposed

Create a class account that stores customer name, account number and type of the account. From this derive the classes cur_acct and sav_acct to make them more specific to their requirements. Include necessary member functions to achieve the following tasks.

Accept deposit from a customer and update the balance.

a. Display the balance.
b. Compute and deposit interest.
c. Permit withdrawal and update the balance.
d. Check for the minimum balance, impose penalty, and update the balance.
e. Modify the program to include constructors for all the three classes.

8. **Friend Function**

Create two classes DM and DF which store the value of distances. DM stores distances in meters and centimetres whereas DF in feet and inches. Write a program that can read values for the class objects and add one object of DM with another object of DB.

Use a friend function to carry out the addition operation. Write suitable functions such that you can perform the following operations.

DM2 = DM1 + 10.5 DM4 = 10.5 + DM3

DM5 = DM1 + DF1 DF2 = DF1 + DM6

9. Friend Class

Use the friend class concept to implement the following scenario.

Define a class incometax that has the functionality to calculate the income tax of a Doctor and an Engineer which are represented as two different classes and store the tax to a private data member of incometax class. Make incometax class as a friend to both Doctor class and Engineer class. Include a display function that displays salary and tax of an Engineer and a Doctor.

10. Practice the programs given in Readings session of Week 3