FAST School of Computing

Object Oriented Programming – Spring 2023

Cyber Security Department

LAB 06

Classes

Learning Outcomes

In this lab you are expected to learn the following:

- Basic Implementation of Classes in C++

Problem 1:

Create a class **Student**. The data members of the class are roll number (string), name (String), age (int), CGPA (double). Then Implement member functions to;

- ✧ Set the values of all data members.
- ✧ Display the values of all data members

Problem 2:

Write a class called **rectangle**. Your task is to store the length and width of the rectangle. Then implement member functions;

- ✧ A function to compute the area of the rectangle.
- ✧ A function that will display the length, width and area of the rectangle.
- ✧ A function to scale up the rectangle by adding 1 to length and width, and add the updated values to get the resultant.
- ✧ Create setter and getter methods.

Problem 3:

Create a class **point** that take two-point x, y and return the plan of the coordinates.

- ✧ Create default constructor
- ✧ Create parameterized constructors
- ✧ Setters getter of the member variables
- ✧ Function that will return the coordinate plane

Problem 4:

We have a real world scenario where we have to simulate a park ticket system. A park has some specific seating capacity, we also have ticket price, ticket number, and total amount we get for the day. Price of ticket per head is Rs. 20/- . Write a class called **park** with the following member functions

- ✧ Default and parameterized constructor
- ✧ Setters and getters
- ✧ A function that perform entrance in the park (function has one argument the number of persons going into the park)
- ✧ On exit updating of the capacity (one argument how many people left the park)
- ✧ Total amount we get for the day (return total amount of the day).
- ✧ Isfull function that will check is there no more capacity in the park

Problem 5:

Create a class '**Date**', with three private variables 'day', 'month', 'year'. Write a no argument constructor to iniatilize date to 01/01/1900. Also write a three argument constructor Date (int day, int month, int year) to show constructor overloading. Also create a destructor. Create two functions with following signatures:

- bool LeapYear (Date obj)
  - Checking if the date is within a leap year
- int SubstractDate (Date obj1, Date obj2)
  - Subtracting two dates to give a number of days

Submission Details:

1. Save single .cpp file with your roll no and lab number e.g. i22-XXXX_Lab2.cpp
2. Take screen shot of running test cases of tasks.
3. Zip the .cpp file and screen shots (Do not create .rar file) with roll no and lab no. e.g. i22-XXXX_Lab2.zip.
4. Submit the zip file on google class room.