University of Central Punjab

**Faculty of Information Technology**

**Object Oriented Programming**

**Fall 2025**

**Lab 03**

|  |  |
| --- | --- |
| **Topic** | Classes in C++, access specifiers, objects and methods. |
| **Objective** | The basic purpose of this lab is to revise some preliminary concepts of C++ that has been covered in the course of Introduction to Computing and Programming Fundamentals.  Its objective is to recall previously learned basic concepts like revision of arrays and functions |
| **CLO** | Apply object-oriented programming principles to implement real-world problems. |

**Instructions:**

• Indent your code.

• Comment your code.

• Use meaningful variable names.

• Plan your code carefully on a piece of paper before you implement it.

• Name of the program should be same as the task name. i.e. the first program should be Task\_1.cpp

**• void main() is not allowed. Use int main()**

**• You are not allowed to use any built-in functions**

**• You are required to follow the naming conventions as follow:**

**• Variables:** firstName; (no underscores allowed)

**• Function:** getName(); (no underscores allowed)

**• ClassName:** BankAccount (no underscores allowed)

**Students are required to complete the following tasks in lab timings.**

**Task 1:**

Create a class named as **Rectangle** having following private attributes:

**• length (double)**

**• width(double)**

Now write the following for the above-mentioned class:

• The program should include functions to assign user defined values to the above-mentioned variables of the Rectangle class.

**void setValue()**

• Write a **display** function to print the attributes of the class.

**void display()**

• Write a function **calculateArea** which calculates the area of the Rectangleand returns it.

**double calculateArea** **(double length, double width)**

**Introduction:**

This case study will be helpful in exploring the concepts of object-oriented programming (OOP) by developing a Rectangle class. Through this exercise, students will identify the class's features, attributes, and behaviors, to understand how to model real-world objects in code.

**Objective:**

The primary objective is to create a Rectangle class with specific attributes and methods. To perform calculations related to the Rectangle’s properties and data assigned to the attributes.

 **Identify Attributes**:

• What other attributes could be added to the Rectangle class (e.g., color, type, height)?

• How would you modify the class to include these attributes?

 **Define Behaviors**:

• Can you think of other methods that could enhance the functionality of the Rectangle class (e.g., perimeter calculation)?

• How would you implement those methods?

 **Implementation**:

• Implement the Rectangle class based on the identified attributes and behaviors.

**Task 2:**

Create a class named as **Student** having following private attributes:

**• studentName (char \*)**

**• studentID(int)**

**• GPA(double)**

**• degreeName(char\*) (e.g, BSCS, BSDS, BSAI)**

• The program should include setter function(s) to assign user defined values to the above-mentioned variable.

**void setValue()**

• Write a non-returning **display** function to print all the attributes of the class.

**void display()**

• Write a function ‘**calculatePercentage’** which calculates the student’s percentage for a specific course. The function will take 2 parameters for obtained marks and total marks as input

**double calculatePercentage** **(int total, int obtained)**

• Write a function ‘**AssignGrade’** which will assign the grade to that specific course for the student according to the percentage. The function will take the calculated percentage as parameter and then returns or displays the assigned grade.

**void AssignGrade** **(double percentage)**

**OR**

**char AssignGrade** **(double percentage)**

• Create multiple objects of ‘Student’ class with different data. And then display all of the object’s data.

**Use the table given below to assign the grade accordingly.**

**GRADE**

**Percentage**

A+

86-100%

A

82-85%

B+

78-81%

B

74-77%

B-

70-73%

C+

66-69%

C

62-65%

C-

58-61%

D+

54-57%

D

50-53%

F

Below 50%

**Task 3:**

Create a class named as **Calculator** having following private attributes

**• num1 (int)**

**• num2 (int)**

**• num3 (int)**

• The program should include setter function(s) to assign user defined values to the above-mentioned variable.

⁃ **void setValue()**

• Write a non-returning **display** function to print all the attributes of the class.

**⁃ void display()**

Implemnet below methods as a member function of the **Calculator** class

Methods:

• **add():** Adds all three numbers.

• **average():** Calculates the average of the three numbers.

• **multiply():** Multiplies all three numbers.

• **maximum():** Finds the largest value among the three numbers.

• **minimum():** Finds the smallest value among the three numbers.

**Main Function:**

• Creates an object calc of the Calculator class

• Demonstrates the use of these methods by displaying the results.

**Task 4:**

Create a class named as **Car** having following private attributes

• **make (char\*):** The brand or manufacturer of the car (e.g., "Toyota").

• **model (char\*):** The specific model of the car (e.g., "Corolla").

• **year (int):** The year the car was manufactured (e.g., 2000).

• The program should include setter function(s) to assign user defined values to the above-mentioned variable.

⁃ **void setValue()**

• Write a non-returning **display** function to print all the attributes of the class.

**⁃ void display()**

Implement below methods as a member function of the **Car** class

Methods:

• **calculateCarAge(int currentYear):** This method calculates the age of the car based on the current year passed as a parameter.

• **isVintage(int currentYear):** This method checks if the car is vintage. A car is considered vintage if it is older than 20 years.

• **displayCarDetails():** Displays the car's make, model, year, calculated age, and whether the car is vintage.

**Main Function:**

• Creates two Car objects (car1 and car2).

• Calls the displayCarDetails() method to show the details for each car, including their age and vintage status.

**Task 5:**

Now create a menu based program where user should select the function that needs to be tested from the above functions.

**For example:**

• Press 1 for Testing **‘Rectangle’** class.

• Press 2 for Testing **‘Student’** class.

• Press 3 for Testing **‘Calculator’** class.

• Press 4 for Testing **‘Car’** class.

• Press 0 for exit.