

National University of Computer and Emerging Sciences



Lab Manual 09 Object Oriented Programming

Course Instructor	Mr. Bismillah Jan
Lab Instructor (s)	Hamna Waseem Dilawar Shabbir
Section	G
Semester	Fall 2020

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- ✓ Understand and implement Composition
- ✓ Understand and implement Inheritance.

TASK 1: (Composition)

Exercise 1:

Define and implement a class Point in files Point.h and Point.cpp, respectively. This class should provide:

- Two private integer data members x and y which will store the x and y coordinates of a point
- A default constructor which takes two parameters to initialize the x and y coordinates and prints "Point() called" on the screen.
- A function print() which prints out the point on the screen in the format (x,y)
- A destructor which prints "~Point() called" on the screen.

Exercise 2:

Now define and implement a class Circle in files Circle.h and Circle.cpp. This class should contain:

- A private data member center which will be an instance of the Point class
- A private float data member radius that will store the radius of the circle
- A constructor which takes three parameters (x and y coordinates of the center of the circle, and the radius) and initializes the data members accordingly and also prints "Circle() called" on the screen.
- A destructor which prints "~Circle() called" on the screen.
- A function print() which prints the information (center and radius) of the circle on the screen

To call the constructor of class Point from the constructor of class Circle, you can use the following syntax.

```
Circle::Circle(int x, int y, float r): center(x,y) { ... };
```

Add another file Lab.cpp in your project. Copy the following piece of code in that file, compile and then execute. Note down the output of the program and write it in comments in the code.

```
#include "Circle.h"

void main()
{
    Circle c (3,4,2.5);
    c.print();
}
```

Exercise 3:

Define and implement a class Quadrilateral in files Quadrilateral.h and Quadrilateral.cpp. This class should provide:

- Four private data members w, x, y and z (Point type) which will be indicating the four corners of the quadrilateral.
- A constructor which takes eight parameters (x and y coordinates of the four corners) and initializes the data members accordingly and prints “Quadrilateral () called” on the screen.
- A destructor which prints “~Quadrilateral called” on the screen.
- A function print () which prints out the information (i.e. the coordinates of its four corners) of the quadrilateral object on the screen.

Exercise 4:

Modify the Lab.cpp file to instantiate an object of class Quadrilateral called obj with parameters for points (1, 0) (0, 1), (1, 1) and (0, 0) and call its print function. Note down the output of the program and write it in comments in the code.

TASK 2: (Inheritance)

Note: Use initializer list for all classes in this task.

Exercise 1:

Define and implement a class Square in files Square.h and Square.cpp, respectively. This class should be inherited from the Quadrilateral class.

- A default constructor of Square should print “Square() called” on the screen.
- A display function should be there in Square Class which prints its coordinates.
- A destructor which prints “~Square () called” on the screen.

Exercise 2:

Define and implement another child class of Quadrilateral called “Rectangle”. Complete the class in Rectangle.h and Rectangle.cpp, respectively.

Rectangle class should also have the following features:

- A default constructor of Rectangle should print “Rectangle () called” on the screen.
- A display function should be there in Rectangle Class which prints its coordinates.
- A destructor which prints “~Rectangle () called” on the screen.