**National University of Computer and Emerging Sciences**



**Lab Manual 08**

**Object Oriented Programming**

|  |  |
| --- | --- |
| Course Instructor | Ms. Arooj Khalil |
| Lab Instructor (s) | Ms. Fariha Maqbool  Ms. Amara Nasir |
| Section | BSE-2A |
| Semester | Spring 2023 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## Objectives

After performing this lab, students shall be able to:

* Understand Aggregation, Composition and Association

# **TASK 1: (Composition)**

### **Step 1:**

Define and implement a class Point - Point.cpp. 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.

### **Step 2:**

Now define and implement a class Circle - 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) { … };

main() function:

void main()

{

Circle c (3,4,2.5);

c.print();

}

### **Step 3:**

Define and implement a class Quadrilateral - 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.

### **Step 4:**

Modify the main() 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.

**TASK 2 (Association):**

Implement a class **Teacher** that has following members:  
int EmployeeID

char\* Name

Create another class **Student** with these data members:

Char \* roll number

Char \* name

Teacher\* TeacherList; //List of all teachers who are teaching this student

Create constructors, destructors, Display function of both the classes such that Teachers are associated with Students.

**Note:**

* Deallocate all dynamically allocated memory.
* Do not use any string class built-in functions except for strlen(), if required.
* Follow all the code indentation, naming conventions and code commenting guidelines.

**TASK 3 (Aggregation):**

Design a "Library" class that has a collection of "Book" objects. The "Library" class should allow adding books and displaying the details of all books in the library.

Instructions:

- Create a class "Book" with attributes "title", "author", and "publicationYear".

- Create a class "Library" that has a private attribute, a vector of type "Book".

- Implement methods "addBook()" and "displayBooks()" in the "Library" class.