National University of Computer and Emerging Sciences

Lab Manual
Object Oriented Programming

Course Instructor	Ms. Hafsa Tariq
Lab Instructor (s)	Ms. Yusra Arshad Ms. Sonia Anum
Section	BSCS 2J
Semester	Spring 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Objectives

After performing this lab, students shall be able to:

- Composition
- Inheritance

TASK 1:

a) 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.

b) 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"
int main()
{
      Circle c (3,4,2.5);
      c.print();
}
```

TASK 2:

Consider the following hierarchy as it exists in a university:

- There are two types of persons in the university i.e. Student and Faculty
- Every **Person** has some basic information that is common to all persons i.e. the ***first_name*** and ***last_name*** stored as private attributes and ***age*** which is a protected attribute.
- A **student** can in turn be either an **Undergraduate** or a **Graduate** student, every student has a ***cgpa***

An **undergraduate** student has a ***fyp_name*** and **supervisor name** as his private attributes.

- A **graduate** student has a ***thesis topic*** and **supervisor name** as his private attributes.
- A student can in turn be either an Undergraduate or a Graduate student, every student has a **cgpa** and **rollNumber** as his private attributes.
- A **faculty member** has private attributes about the number of courses he is currently teaching, i.e. his ***course_count*** and a three digit telephone ***extension*** number.

- An **undergraduate** class should be inherited from the person class.
- A **graduate** class should be inherited from the person and undergraduate class.
- A **faculty** class should be inherited from the person class.
- A faculty class should have following functions.
  - Get_age
  - Get_course_count
  - Get_extension
  - Set_extension
  - Set_ course_count
- An undergraduate class should have following functions.
  - ✓ Get_age
  - ✓ Set_cgpa
  - ✓ Get_cgpa
  - ✓ Get_fyp_name
  - ✓ Set_fyp_name
  - ✓ Set_roll_number
  - ✓ Get_roll_number
  - ✓ Set_supervisor_name
  - ✓ Get_supervisor_name

- A graduate class should have following functions.
  - ✓ Get_age
  - ✓ Set_cgpa
  - ✓ Get_cgpa
  - ✓ Get_thesis_name
  - ✓ Set_thesis_name
  - ✓ Set_roll_number
  - ✓ Get_roll_number
  - ✓ Set_supervisor_name
  - ✓ Get_supervisor_name

- A person class should have following functions.

  - ✓ Set_age
  - ✓ Set_first_name
  - ✓ Get_first_name
  - ✓ Set_last_name
  - ✓ Get_last_name

Implement these classes i.e. define all the classes along with their attributes and their inheritance. Every class should be defined in a separate header file named according to the class name.