**National University of Computer and Emerging Sciences**

Logo

Description automatically generated with medium confidence

**Lab Manual # 08**

Object Oriented Programming

|  |  |
| --- | --- |
| Course Instructor | Mr. Waqas Manzoor |
| Lab Instructor (s) | Mr. Dilawar Shabbir |
| Section | BSE-2C |
| Semester | Spring 2022 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

## **Objectives**

* Part-Whole (Has-a) relationships, specifically focusing on Composition, Aggregation and Association concepts and implementations.

**Composition**

### Exercise 1:

Make a new application called Lab\_<your roll number>. 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 Triangle in files Triangle.h and Triangle.cpp. This class should provide:

* Three private data members x, y and z (Point type) which will be indicating the three corners of the triangle.
* A constructor which takes six parameters (x and y coordinates of the three corners) and initializes the data members accordingly and prints Triangle() called on the screen.
* A destructor which prints ~Triangle() called on the screen.
* A function print() which prints out the information (i.e. the coordinates of its three corners) of the triangle object on the screen

**Aggregation**

### Exercise 5:

Define and implement a class Style in files Style.h and Style.cpp. This class should include:

* A private data member char color[10] which stores the color of the object
* A private boolean data member isFilled
* Setters for these data members (i.e. SetColor and SetFilled)
* A constructor which takes the color and a boolean value and initializes the data members. The constructor should also print Style() called on the screen.
* A destructor which prints ~Style() called on the screen.
* A function print() which prints the data members of the style on the screen

### Exercise 6:

Now modify the Circle class to include a pointer of type Style called st. Note that you will have to modify the constructor and print function of Circle class accordingly. Initially, a newly constructed Circle will have no style, so in the constructor you should point st to NULL. At this stage, you will have to add three more methods to the Circle class:

* Method SetStyle (Style\*) will take a pointer of type Style and set st to the pointer passed as an argument
* Method SetColor (char[]) will update the color of the circle. You will call the SetColor method of Style class inside this method.
* Method SetFilled (bool) will update the filled property of the circle. You will call the SetFilled method of Style class inside this method.

Modify the Lab.cpp file to instantiate a pointer of type Style called style1. The color of this new style should be blue and isFilled set to true. Instantiate an object of Circle called circle1. Now set the Style of circle1 to style1. Now call the print function of circle1. Note down the output and put it in comments inside the code.

**Association**

**Exercise 7:**

Make a class **Student** that has following data members:

char name[50];

char rollNo[8];

float cgpa;

Students class provides a constructor with default arguments and a function **Print** that prints students name and roll number on screen in following format:

StudentName (RollNo.)

For Example, Aslam Baig (12L9356)

Do we need a Destructor for this class?

**Exercise 8:**

Write the following piece of code in your main function: it should create six students with information provided.

|  |
| --- |
| Student s1("12L1111", "Hashim Amla", 3.99);  Student s2("13L1121", "Virat Kohli", 3.45);  Student s3("13L1126", "Quinton de Kock", 2.98);  Student s4("14L1361", "Joe Root", 2.99);  Student s5("14L1124", "Martin Guptil", 3.09);  Student s6("15L1314", "Rohit Sharma", 3.19); |

**Exercise 9:**

A Student **Society** has a president and five members from students. Make a class Society that has following private data members:

char name[50];

Student\* president;

Student\* members[5];

The Society class has a constructor with default arguments that takes the name of society as parameter. Why are we keeping Student pointers in Society class and what should the constructor do?

**Exercise 10:**

Write a member function of Society class **PrintInfo** that prints name of society and details of its members and president using the **Print** function of Student class. What should function do if some member does not exist?

**Exercise 11:**

Add following lines in your main function it should give following output:

|  |
| --- |
| Society sports ("Sports");  sports.PrintInfo(); |

**Output:**

|  |
| --- |
| Society Name: Sports  President: Not Available  Member 1: Not Available  Member 2: Not Available  Member 3: Not Available  Member 4: Not Available  Member 5: Not Available  Press any key to continue . . . |

Why is it displaying Not Available in members’ information? Because president and members pointers are currently pointing to NULL. We need to point these pointers to students’ objects in order to create association between sports society and students.

**Exercise 12:**

Make a member function **AppointPresident** in Society class that takes a student object by reference and appoints it to president’s position if the position is vacant and the cgpa of student is above 3.00. Display appropriate error message otherwise. Do you need to add Getters in Students class to accomplish this task?

**Exercise 13:**

Add following lines in your main function and verify the output:

|  |
| --- |
| sports.AppointPresident(s3);  sports.AppointPresident(s1);  sports.AppointPresident(s2); |

**Output:**

|  |
| --- |
| ...  Quinton de Kock cannot be appointed as President. CGPA criteria not met.  Hashim Amla has been appointed as President.  Virat Kohli cannot be appointed as President. President position is NOT vacant.  Press any key to continue . . . |

Note: AppointPresident need to call GetName of student in order to print this message.

**Exercise 14:**

Make a member function **AddMember** in Society class that takes a student pointer and adds it in the list of members if there is any position vacant and displays the error message otherwise. Also the president cannot be in the list of society members. Also a student cannot be added as a member more than once.

**Exercise 15:**

Add following lines in your main function and verify the output:

|  |
| --- |
| cout << endl << endl << endl;  Student s7("15L1334", "Robert Elen", 2.19);  sports.AddMember(s3);  sports.AddMember(s2);  sports.AddMember(s3);  sports.AddMember(s1);  sports.AddMember(s4);  sports.AddMember(s5);  sports.AddMember(s6);  sports.AddMember(s7);  sports.PrintInfo(); |

**Output:**

|  |
| --- |
| ...  Quinton de Kock has been added to members list successfully.  Virat Kohli has been added to members list successfully.  Quinton de Kock already exists in Members list.  President cannot be added in Members list.  Joe Root has been added to members list successfully.  Martin Guptil has been added to members list successfully.  Rohit Sharma has been added to members list successfully.  Robert Elen cannot be added to members list. Member position is NOT vacant.  Society Name: Sports  President: Hashim Amla (12L1111)  Member 1: Quinton de Kock (Roll No here)  Member 2: Virat Kohli (Roll No here)  Member 3: Joe Root (Roll No here)  Member 4: Martin Guptil (Roll No here)  Member 5: Rohit Sharma (Roll No here)  Press any key to continue . . . |

You may keep currentMembersCount in your Society class. Should it be a static data member or non-static?

Note: This is responsibility of Student Class to print its name and roll number, use Print of Student class here.

**Exercise 16:**

Make a member function **SuspendMember** in Society class that takes a char\* memberNameas parameter searches that member from the list of members and removes that member from the list if found otherwise displays the error message. Should SuspendMember delete the Student?

**Exercise 17:**

Add following lines in your main function and verify the output:

|  |
| --- |
| cout << endl << endl << endl;  sports.SuspendMember("abc");  sports.SuspendMember("Martin Guptil"); |

**Output:**

|  |
| --- |
| …  abc NOT found in the Members list.  Martin Guptil has been suspended from Sports society.  Press any key to continue . . . |

Print Society Information again to check if the student has been removed from the members list.

**Exercise 18:**

Add following lines of code in your main function and test the result.

|  |
| --- |
| cout << endl << endl << endl;  Society ieee("IEEE");  ieee.AppointPresident(s5);  ieee.PrintInfo(); |