# Lab Manual – Association

**Exercise 1:**

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 2:**

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 3:**

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;

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 4:**

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

**Exercise 5:**

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

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

**Output:**

|  |
| --- |
| Society Name: Sports  President: Not Available  Press any key to continue . . . |

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

**Exercise 6:**

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 7:**

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.

# Lab Manual – Composition and Aggregation

## Objectives

After completing this lab, you will be able to:

* Identify and implement the “Composition” relationship between classes
* Identify and implement the “Aggregation” relationship between classes

**Definitions of Aggregation and Composition**

**Aggregation:**

The aggregation relationship is used to represent the ownership or a whole/part relationship between classes. The aggregate object has one or more parts which may be shared with other objects of the same class or other classes. The objects that make up the parts are created and destroyed independently of the aggregate object.

**Composition:**

Composition is used to represent a stronger kind of ownership than the aggregation relationship. In composition, the composed object has full responsibility for the disposition of its parts in terms of creation and destruction. Talking in terms of implementation, the composite has the responsibility for memory allocation and de-allocation of its parts. Moreover, the parts of a composite object cannot be shared with other objects.

**Composition**

### Exercise 1:

Make a new application called Lab\_<your roll number>. Define and implement a class Point. 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 . This class should contain:

Class Circle{

Point center;

Float radius;

};

* 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 the main. 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.

void main()

{

Circle c (3,4,2.5);

c.print();

}

**Aggregation**

### Exercise 3:

Define and implement a class Style in file 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 4:

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 main 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.

**END**