

Question 1

You are developing a system to manage a university's course enrollment. Each course can have multiple students enrolled in it, and each student can be enrolled in multiple courses. Design a system using C++ that represents this relationship using aggregation. Additionally, include a concept of struct to represent the student's information within the system. One course can have a maximum of two Students.

- Student should be a struct
- Course should be a class.

Closely, look at the following main() function.

```
int main() {  
  
    Student student1("Shahood", 1001);  
    Student student2("Hammad", 1002);  
    Student student3("Ali", 1003);  
  
    Course course1("Math", 2001);  
    Course course2("Physics", 2002);  
  
    course1.enrollStudent(&student1);  
    course1.enrollStudent(&student2);  
  
    course2.enrollStudent(&student2);  
    course2.enrollStudent(&student3);  
  
    cout << course1 << endl;  
    cout << course2 << endl;  
  
    return 0;  
  
}
```

Question 2

Write a superclass called Shape (as shown in the diagram), which contains:

- Two instance variables color (String) and filled (boolean).
- The required constructors: a default constructor that initializes the color to "green" and filled to true, and a constructor that initializes the color and filled to the given values.
- Getter and setter for all the instance variables.
- A toString() method that returns "A Shape with color of xxx and filled/Not filled".

Write two subclasses of Shape called Circle and Rectangle, as shown in the diagram.

The Circle class contains:

- An instance variable radius (double).
- The required constructors. The default constructor initializes the radius to 1.0.
- Getter and setter for the instance variable radius.
- Methods getArea() and getPerimeter().
- Redefine the toString() method inherited, to return "A Circle with radius=xxx, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.

The Rectangle class contains:

- Two instance variables width (double) and length (double).
- The required constructors. The default constructor initializes the width and length to 1.0.
- Getter and setter for all the instance variables.
- Methods getArea() and getPerimeter().
- Redefine toString() method inherited, to return "A Rectangle with width=xxx and length=zzz, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.

Write a class called Square, as a subclass of Rectangle. Convince yourself that Square can be modeled as a subclass of Rectangle.

Square has no instance variable, but inherits the instance variables width and length from its superclass Rectangle.

- Provide the appropriate constructors (as shown in the diagram).
- Redefine the toString() method to return "A Square with side=xxx, which is a subclass of yyy", where yyy is the output of the toString() method from the superclass.
- Redefine the setLength() and setWidth() to change both the width and length, so as to maintain the square geometry.

