National University of Computer and Emerging Sciences



Lab Exercise 11

For

Object Oriented Programming Lab

|  |  |
| --- | --- |
| Course Instructor(s) | Dr. Danish |
| Lab Instructor(s) | Mr. Mughees Ismail |
| Semester | Spring 2020 |

**FAST School of Computing**

# Instructions:

1. Make a word document with the naming convention “SECTION\_ LAB#\_ROLLNO” and put all your source code and snapshots of its output in it. Make sure your word file is formatted properly.
2. Plagiarism is strictly prohibited.
3. Do not discuss solutions with one another.

# Useful links

|  |
| --- |
| **Question#1** |

Develop a class hierarchy of shapes and write a program that finds area of different objects. The hierarchy will consist of a parent class Shape with three derived classes - Sphere, Rectangle, and Cylinder. For the purposes of this exercise, the only attribute a shape will have is a name and the method of interest will be one that computes the area of the shape (surface area in the case of three-dimensional shapes). Do the following.

1. Write an abstract class Shape with the following properties: An instance variable shapeName of type String An abstract method area() A toString() method that returns the name of the shape.

2. Sphere class which is a descendant of Shape. A sphere has a radius and its area (surface area) is given by the formula 4\*PI\*radius^2. Define similar classes for a rectangle and a cylinder. Both the Rectangle class and the Cylinder class are descendants of the Shape class. A rectangle is defined by its length and width and its area is length times width. A cylinder is defined by a radius and height and its area (surface area) is PI\*radius^2\*height. Define the toString method in a way similar to that for the Sphere class.

Instantiate the three shape objects: deck to be a 20 by 35 foot rectangle, bigBall to be a sphere of radius 15, and tank to be a cylinder of radius 10 and height 30. Make the appropriate method calls to assign the correct values to the three amount variables. Run the program and test it. You should see polymorphism in action while computing values.

|  |
| --- |
| **Question#2** |

Write a class Rational having two attributes a numerator and denominator with functions input(), print(), and some operator overloading functions defined below:

1. ++ prefix (that will increment the numerator and denominator and return the updated value)
2. ++ postfix (that will increment the numerator and denominator and return the value before updation)
3. - - prefix (that will decrement the numerator and denominator and return the updated value)
4. - - postfix (that will decrement the numerator and denominator and return the value before updation)
5. ! operator (that changes the numerator to denominator and denominator to numerator)

Sample Output:

Enter Rational Number

Numerator: 5

Denominator: 6

You inputted: 5/6

**Choose operation you want to apply: (Suppose user chooses 1)**

**The rational number is: 6/7.**

//The output in bold letters will continue until user presses -1.

**Note:** Display error message if denominator is zero.