**Object Oriented Programming in Java (Graded Lab 3)**

**Problem:**

Assuming we have a Circle class defined earlier which throws an appropriate exception when user tries to create invalid circle and extends GeometricObject. Assume a Colorable interface which contains a method double costToColor(double c) which returns the cost of painting where c is the cost to paint per unit area and Circle class implements Colorable.

Define another Class ConcentricCircles extending GeometricObject which has a set of two concentric circles as shown in figure below to show for example a circular pavement around a circular field which throws appropriate exception when object assigned illegal values. The newly defined class should implement Colorable, cost of painting should only include area of pavement (shaded region). For newly defined class.

A test program in another package which allows user to create as many objects of type circle or concentric circles as user wants in iterative loop as defined earlier. An arraylist, a version of array which allows size increasing automatically, of Geometricobjects has to be created initially to refer different objects being created.

User may ask during these iterations the total cost to paint or cost to paint any particular geometric

object. Define another generic method double ComprehensiveCost(<ArrayList of

GeometricObjects>, int position), such that when position is -1 the method returns cost of all

figures in the Array of objects and if valid number it returns the cost of painting a figure located at position in array.

The output which need to be pasted with the submission should include one example of illegal

object being created of both circle and concentric circle. One case for finding cumulative cost of

objects of different type of objects. Cost for an object of type Circle and ConcentricCircle.

**Structure:**

The package SNU.geometryUtil contains 5 classes: Circle, ConcentricCircle, IllegalCircleException, IllegalConcentricCircleException, GeometricObject and Colorable Interface.

Default Package contains the ConcentricCircleMain class which contains the main function for the program.

**Input:**

Program on each iteration asks what object to create and on creating the specified object asks whether to calculate the cost to paint any object from the list or the total cost or not. If not the program goes on until the user stops creating new objects.

**Output:**

Program returns the cost of painting a specified object or the total cost of painting all the objects currently in the arraylist depending on user choice.