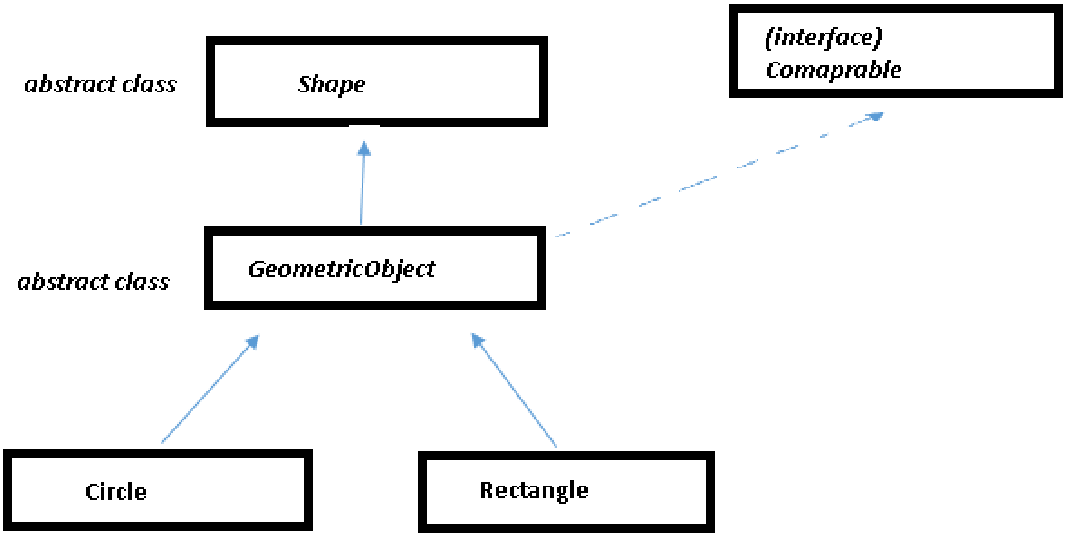
CSC253 – Assignment 1

Shape Project (Total: 100 pts)

This hierarchy begins with superclass **Shape**, which is extended by subclass **GeometricObject** – is a shape. Also, GeometricObject is a ***Comparable*** (*interface*) object. The third level of this hierarchy contains specific types of **Circle** and **Rectangle.**

As in the below figure, we can follow the arrows from the bottom of the diagram to the topmost superclass in this class hierarchy to identify several ***is-a*** relationships. For example, a **circle** is a geometric object and is a shape as well as is a comparable object.



Requirements:

Need to create the following **four (4)** classes and each class MUST include the **code documentations**

1. **Shape** class (20 pts) –abstract class

class	Variable	Constructor/method

Shape	-X: int	Shape()
		Shape(int, int)

	-Y: int	setX(int x): void and setY(int Y): void
		getX(): int and +getY(): int
		getName(): abstract String
		getArea(): abstract double
		getPerimeter(): abstract double

2. **GeometricObject** class (30 pts) – abstract class

class	Variable	Constructor/method
GeometricObject	-color: String	GeometricObject() GeometricObject(String color, boolean filled), GeometricObject(int x, int y, String color, boolean filled)
	-filled: boolean	getColor(): String and setColor(String):void
	-dateCreated: java.util.Date	isFilled():Boolean and setFilled(Boolean filled):void getDateCreated():java.util.Date
		toString():String
		compareTo(GeometricObject o):int
		max(GeometricObject o1, GeometricObject o2): GeometricObject ➔ ***see the below***

```
//method max

public static GeometricObject max(GeometricObject o1, GeometricObject o2) {

   if (o1.compareTo(o2) > 0)

     return o1;

   else

     return o2;

 }
```

3. **Rectangle** class (25 pts)

class	Variable	Constructor/method
Rectangle	-width: double	Rectangle () Rectangle(double width, double height)
	-height: double	getwidtht(): double and setwidtht(double):void
		getHeight(): double and setHeight(double):void
		getArea():double getPerimeter(): double getName():String compareTo(GeometricObject o):int

Rectangle:

Area = width * height

Perimeter = width * 2 + height *2

4. **Circle** class (25 pts)

class	Variable	Constructor/method
Circle	-radius: double	Circle () Circle(double radius) **Circle(double radius, String color, boolean filled)**
		getRadius(): double and setRadius(double):void
		getDiameter(): double and setDiameter(double):void
		getArea():double getPerimeter(): double getName():String compareTo(GeometricObject o):int

Circle:

Diameter = radius * 2

Area = radius * radius * π

Perimeter = diameter * π

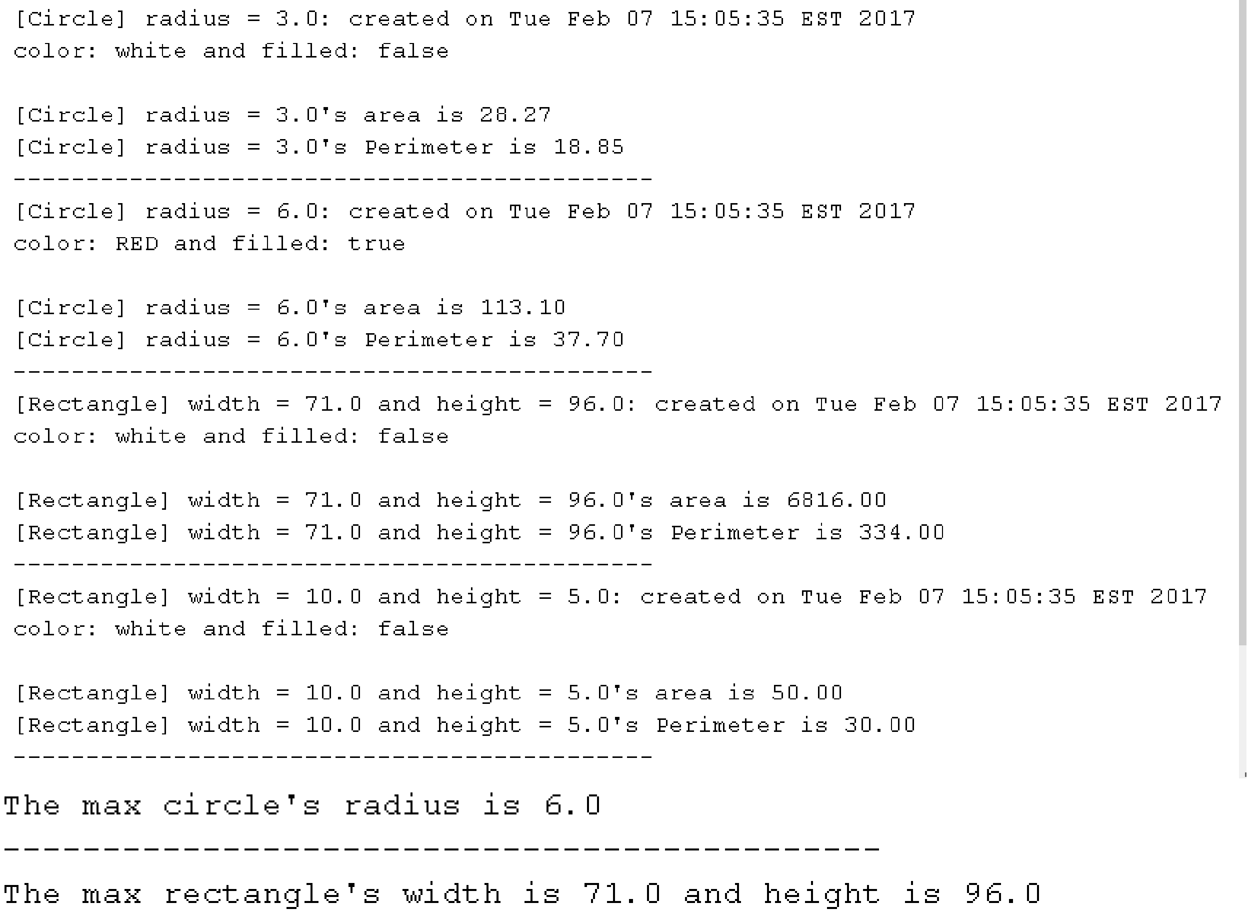
Do not change my **ShapeTest.java** file.

See the below:

```

import java.util.ArrayList;
// FileName: ShapeTest.java
// Program tests the Shape hierarchy.
//created by: Sylvia Yeung

public class ShapeTest
{
  // create Shape objects and display their information
  public static void main( String args[] )
  {
   //add four objects
ArrayList<Shape> shapes = new ArrayList<>();
 Circle circle1 = new Circle(3.0);
 Circle circle2 = new Circle( 6.0, "RED", true);
 Rectangle rect1 = new Rectangle( 71, 96 );
 Rectangle rect2 = new Rectangle(  );
shapes.add(circle1);
shapes.add(circle2);
shapes.add(rect1);
shapes.add(rect2);

   // call method print on all shapes
    for ( Shape currentShape : shapes )
    {
      System.out.printf( "%s: %s",
        currentShape.getName(), currentShape );

      if ( currentShape instanceof GeometricObject)
      {
     //  GeometricObject circleShape =
      //   ( Circle) currentShape;
        System.out.printf( "%s's area is %.2f\n",
          currentShape.getName(), currentShape.getArea() );
     //print the perimeter
       System.out.printf( "%s's Perimeter is %.2f\n",
          currentShape.getName(), currentShape.getPerimeter() );
        //print the line
      System.out.println("-----------------------------------------");
         } // end if
        } // end for
     // Display the max circle
   Circle circle = (Circle) GeometricObject.max(circle1, circle2);
   System.out.println("The max circle's radius is " + circle.getRadius());
```

```
  System.out.println("--------------------------------------------");
    // Display the max rectangle
   Rectangle rect = (Rectangle) GeometricObject.max(rect1, rect2);
   System.out.println("The max rectangle's width is " + rect.getWidth() + " and height is " +
rect.getHeight());
    } // end main
} // end class ShapeTest
```

The result:



```
[Circle] radius = 3.0: created on Tue Feb 07 15:05:35 EST 2017
color: white and filled: false

[Circle] radius = 3.0's area is 28.27
[Circle] radius = 3.0's Perimeter is 18.85
--------------------------------------------
[Circle] radius = 6.0: created on Tue Feb 07 15:05:35 EST 2017
color: RED and filled: true

[Circle] radius = 6.0's area is 113.10
[Circle] radius = 6.0's Perimeter is 37.70
--------------------------------------------
[Rectangle] width = 71.0 and height = 96.0: created on Tue Feb 07 15:05:35 EST 2017
color: white and filled: false

[Rectangle] width = 71.0 and height = 96.0's area is 6816.00
[Rectangle] width = 71.0 and height = 96.0's Perimeter is 334.00
--------------------------------------------
[Rectangle] width = 10.0 and height = 5.0: created on Tue Feb 07 15:05:35 EST 2017
color: white and filled: false

[Rectangle] width = 10.0 and height = 5.0's area is 50.00
[Rectangle] width = 10.0 and height = 5.0's Perimeter is 30.00
--------------------------------------------
The max circle's radius is 6.0
--------------------------------------------
The max rectangle's width is 71.0 and height is 96.0
```