Assignment 1

CSCI 2010U - Principles of Computer Science

Due: 4 October 2013 at Midnight

Q1. Java Interfaces and Abstract Classes [60 marks]

You are to develop a collection of classes for modelling circles, rectangles, and triangles. You must organize your project in the following four classes:

- The class GeometricObject models common features of geometric objects.
  - It contains the properties color and filled and their appropriate get() and set() methods.
  - It also contains abstract methods double getArea() and double getPerimeter()
  - It also contains an abstract method void makeArea(double area). This method is interesting as it is able to pick appropriate parameters of an object (e.g., radius for a circle) to get a desired area.
- The Rectangle is a subclass of GeometricObject.
  - It implements the abstract methods in its parent class. It also has members, width and height, needed to specify a rectangle.
  - Override makeArea() method. Assume that it only changes the height of the rectanble to get the desired value of the area.
- The Circle is a subclass of GeometricObject.
  - It implements the abstract methods in its parent class. It also has a member radius needed to specify a circle.
  - Override makeArea() method. It changes the radius of the circle to get the desired area.
- The Triangle is a subclass of Rectangle.
  - A triangle is defined using its width and height.
  - Override makeArea() method. Assume that it only changes the height of the triangle to get the desired value of the area.

We are also interested in finding the larger of the two geometric objects. E.g., given two circles we want to determine which of these has the larger area. We achieve that by implementing a int compareTo(Object o) method for these classes. This method is defined in the Comparable interface. You are asked to write the GeometricObject class that implements the Comparable interface.

Your program will create two circles c1 and c2, two rectangles r1 and r2, and two triangles t1 and t2, and compare their sizes and print out the names of the larger circle, the larger rectangle, and the larger triangle.

The program will read from a text file, whose name will be passed as a command line argument. The format of the input file is:

```
<c1 radius>
<c2 radius>
```

```
<r1 width height>
<r2 width height>
<t1 width height>
<t2 width height>
<c3 radius desired-area>
<r3 width height desired-area>
<t3 width height desired-area>
```

The first two lines specify the two circles by providing their radii and the last two lines specify the rectangles by providing their width and height.

Example:

Say you have text file test.txt containing

```
2
3
1 2
1 1
10 12
1 2
4 1000
1 7 14
3 4 56
```

And you type java A1q1.java test.txt you will get the following output

```
c1: radius = 2.0, area = 12.6
c2: radius = 3.0, area = 28.3
--
r1: width = 1.0, height = 2.0, area = 2.0
r2: width = 1.0, height = 1.0, area = 1.0
--
t1: width = 10.0, height = 12.0, area = 60.0
t2: width = 1.0, height = 2.0, area = 1
--
c2 is bigger than c1
r1 is bigger than r2
t1 is bigger than t2
--
c3 area adjusted to 1000
r3 area adjusted to 14
t3 area adjusted to 56
```

Note how decimal numbers are shown to the first decimal place only. It is intentional. Please format your decimal numbers accordingly.

Q2. Part 2 Java Exception Handling [40 marks]

Now extend the code that you wrote in the previous section to handle incorrect inputs. For our purposes negative (and zero) values for radius, width and height are considered illegal. When encountered with a negative value your program should display the following message and exit.

Example:

Say you have text file test.txt containing

```
2
3
1 2
1 1
-10 12
1 2
2 4000
3 4 303
3 4 532
```

And you type java A1q2.java test.txt you will get the following output

```
Illegal value encountered at line 5: [-10 12]
```

Submission

Please follow these instructions to the letter. Thank you.

1. Solution to Q1 will be in file A1q1.java and solution to Q2 will be in A1q2.java.
2. The first line of your A1q1.java and A1q2.java must adhere to the following format:
   // Your Name – Student Number
3. Please submit a1.zip file that contains A1q1.java and A2q2.java on the blackboard.
4. Ensure that the output of your programs follow exactly the format of the outputs shown in blue above.