CS122 Programming Project1: Inheritance

Due date: 02/09/2017 11:59pm on Blackboard (50 points)

How to submit your project: Create a folder named as Project1 and put all your XXX.java files into the folder. Then zip the folder and submit it on Blackboard. Be careful that I need your .java files, not .class files.

Part1:

Design and implement a set of classes that define a series of shapes: rectangle, square and circle. Use Shape as the abstract parent class, which requires two abstract methods called computeArea and computeCircumference. For each child class, define their own data and methods as needed. The toString method is necessary for each class. You might need to search on internet the formula for area and circumference for each shape.

Create a main driver class to instantiate several shapes of different types and exercise the behaviors (methods) you provided.

Part2:

For this part, you can either create a new tester class or continue working on the previous tester by adding new instructions.

1. Your Rectangle.java should contain two instance variables: length, and width. In the driver class, create two rectangles with same length and width. Use the `equals` method, which Shape inherits from the Object class, and Rectangle inherits from Shape, to determine whether two rectangles are the same. For example,

   ```
   if (rect1.equals(rect2))
         System.out.println("They are same rectangles.");
   else
         System.out.println("They are different rectangles.");
   ```

   You have not implemented the equals method, but you can still call it because this method is defined in the Object class. However, are the results returned what you expected?

2. The problem above is that as defined in the Object class, equals method does an address comparison. It says that two objects are the same if they live at the same memory location. The two Rectangle objects in this program are not aliases, so even if they contain exactly the same information they will be "not equal." To make equals() compare the actual information in the object, you can override it with a definition specific to the class. It makes sense to say that two rectangles are "equal" if they have the same length and width.

3. Override the equals method in Rectangle class. Your method should have a header like

   public boolean equals(Rectangle r) { … }

   It takes a Rectangle object as parameter and returns true if it is equal to the current object, false otherwise. Now when you run the driver class, it should give the correct results.

4. (Optional +5 points) Override the equals method in the other child classes. Modify the driver program to test each of the different versions of equals method.