Homework 3: Shape Instantiation

Due date: September 12, 2016 (Mon)

For this homework, you will modify the Shape class and subclasses shown in lecture, and you will make some additional subclasses of Shape.

Specification

Shape class - You must provide the following methods in your Shape class:

```
public Shape(double newArea, double newPerimeter)
public double getArea()
public double getPerimeter()
public String toString() - Prints out “Shape <area> <perimeter>”
```

Rectangle class - This must be a subclass of Shape. You must provide the following methods in your Rectangle class:

```
public Rectangle(double newWidth, double newHeight)
public double getWidth()
public double getHeight()
public String toString() - Prints out “Rectangle <area> <perimeter>”
```

Square class - This must be a subclass of Rectangle. You must provide the following methods in your Square class:

```
public Square(double newSide)
public double getSide()
public String toString() - Prints out “Square <area> <perimeter>”
```

Circle class - This must be a subclass of Shape. You must provide the following methods in your Circle class:

```
public Circle(double newRadius)
public double getRadius()
public String toString() - Prints out “Circle <area> <perimeter>”
```

Octagon class - This must be a subclass of Shape and it represents a regular octagon. You must provide the following methods in your Octagon class:

```
public Octagon(double newSide)
public double getSide()
public String toString() - Prints out “Octagon <area> <perimeter>”
```

Hexagon class - This must be a subclass of Shape and it represents a regular hexagon. You must provide the following methods in your Hexagon class:

```
public Hexagon(double newSide)
public double getSide()
public String toString() - Prints out “Hexagon <area> <perimeter>”
```

Each toString() method must return a String with the following format: <shape> <area> <perimeter>. Areas and perimeters must be printed with at least one digit to the left of the decimal point and exactly three digits to the right of the decimal point.

Turning-in Your Work

Place all your .java files into a ZIP file called homework3.zip. The following JAR command must be used to do this:

```
jar cvf homework3.zip *.java
```

Submit your homework3.zip file using AutoLab (https://autolab.andrew.cmu.edu).

Grading

AutoLab will grade your assignment as follows:

- Java files exist and compile: 5 points
- Shape, Square, Rectangle, and Circle classes: 10 points each
- Octagon and Hexagon classes: 20 points each
- Follows coding conventions: 10 points
    - We’ll deduct one point for each coding convention issue detected.
- Author JavaDoc comment at beginning of file: 5 points

AutoLab will show you the results of its grading within approximately one or two minutes of your submission. You may submit multiple times so as to correct any problems with your assignment. Autolab uses the last submission as your grade.