Programming Assignment 3-4

Create Java classes for `Triangle`, `Rectangle`, and `Circle`. Provide each class with a method

```
public double computeArea()
```

Make all of these classes _immutable._ (Follow the guidelines in the slides for creating this type of class.) Provide one constructor for each class; the constructor should accept the data necessary to specify the figure, and to compute its area. The values accepted by the constructor should be stored in (private) instance fields of the class. For example, `Rectangle` should have instance fields `width` and `length`, and the constructor should look like this

```
public Rectangle(double width, double length)
```

For `Triangle`, you may use arguments `base` and `height`. And for `Circle`, use `radius` as the constructor arugment.

Whenever you create instance fields for one of these classes, provide public accessors for them (but do not provide mutators since the class is supposed to be immutable – for instance, the dimensions of a `Rectangle` should be read-only). For example, you will have in the `Rectangle` class:

```
private double width;

public double getWidth() {
      return width;
}
```

Create a fourth class `Main` that will, in its `main` method, test these three figure classes as follows: It will create one instance of each (you can make your own choice for the dimensions of your figures) and then print to the console the area of each. Typical output would be:

```
Area of Triangle is 34.75
Area of Rectangle is 36.0
Area of Circle is 58.57
```

Here are some area formulas, in case you do not remember them:

```
Area of a rectangle  =  width * height
Area of a triangle = 1/2 * base * height
Area of a circle = PI * radius * radius
```
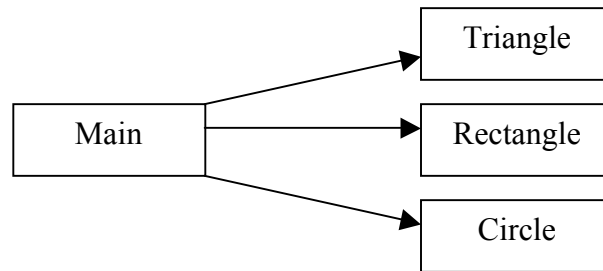
Extra Credit! (Can get 15 points instead of 10). Create a second constructor to your `Triangle` class that accepts three sides (instead of `base` and `height`). The constructor should set the values of the *sides* as instance variables.

Design Tip. It is best to compute area of a triangle is just one way – (1/2) base * height. This means that your new constructor, which accepts three sides, should compute the base and the height (assume that the base is a longest side) using the input values for the three sides. Then, when computeArea is called, it always returns (1/2)base*height.

To be consistent, your original constructor `Triangle(double base, double height)` also needs to be modified so that it computes values for the three sides. Since base and height do not give enough information to tell you precisely the lengths of the three sides, make this assumption: One of the sides is the base, another is the height (and you can compute the third).