Polygon Lab

You have been given an interface file (Polygon.java). You are to write 2 classes, a Rectangle class (Rectangle.java) and a Triangle class (Triangle.java), both of which implement the Polygon interface.

Specs for the Rectangle Class

In addition to the public methods from the interface that you will need to implement, you should also implement the following methods:

public String **toString**(); // all well-written classes should override this inherited method from Object
// sample return strings:
```
[(0,0) (0,5) (4,5) (4,0)] [Rectangle] [area = 20] [perimeter = 18]
[(0,0) (0,5) (5,5) (5,0)] [Square] [area = 25] [perimeter = 20]
```

public boolean **isRectangle**();
public boolean **isSquare**();
public double **diagonalLength**();// coordinates of vertices will have been listed in consecutive order
                                // working around the polygon – therefore, you'll be able to easily find
                                // diagonal vertices

Specs for the Triangle Class

In addition to the public methods from the interface that you will need to implement, you should also implement the following methods:

public String **toString**(); // all well-written classes should override this inherited method from Object
// sample return strings:
```
[(1,0) (1,4) (4,0)] [Isosceles Right] [area = 8] [perimeter = 14]
[(1,0) (1,4) (1,9)] [Not a Triangle] [area = none] [perimeter = none]
```

```
/**
  * Determine the type of triangle.
  * @returns returns one of the following: Equilateral, Isosceles, Isosceles Right,
  *                                         Scalene, Right, Not a Triangle
  */
```
public String **triangleType**();

public boolean **isTriangle**();
public boolean **isEquilateral**();
public boolean **isIsosceles**();
public boolean **isRight**();
public boolean **isScalene**();

We'll use BlueJ to test our methods – there won't be a Driver! This is very neat. It's been started for you.

Note: All math for area/perimeter should be done with doubles for an accurate result – however, you should round your answer before putting it into the String (you may NOT use Math.round() – you must use the way that Ch4 rounds). The format of the String you return for the toString() and triangleType() methods must match the examples above **EXACTLY** (we'll discuss this in class).