

Homework 2

Before attempting this project, be sure you have completed all of the reading assignments, hands-on labs, discussions, and assignments to date.

(25 points) Design a Java class named `Polygon` that contains:

- A private int data field named `numSides` that defines the number of sides of the polygon. The default value should be 4.
- A private double data field named `sideLength` that defines the length of each side. The default value should be 10.0.
- A private double data field named `xCoord` that defines the x-coordinate of the center of the polygon. The default value should be 0.0.
- A private double data field named `yCoord` that defines the y-coordinate of the center of the polygon. The default value should be 0.0.
- A private double data field named `apothem` that defines the apothem of the polygon. The default value should 5.0.
- A private double data field named `perimeter` that defines the perimeter of the polygon. The default value should 20.0.
- A no argument constructor that creates a `Polygon` using the default number of sides, default side length, default x- and y-coordinates, and the default apothem.
- A constructor that creates a `Polygon` using a specified number of sides, side length, x- and y-coordinates and the apothem
- A `getArea()` method that returns a double value representing the area of the polygon. (Note the area of a regular Polgon can be calculated from $\frac{1}{2} * \text{Apothem} * \text{Perimeter}$)
- Getter and setter methods for all data fields
- A `toString()` method that displays the number of sides, side length, x-coordinate, y-coordinate and apothem values in String format

Be sure your code compiles.

Next, write a Java test program, named `TestPolygon`, to create five different polygons. When creating the five polygons, create one using the no argument constructor. For the remaining four, feel free to use any number of sides, side length and x-, and y-coordinates and apothem that are not equal to the default values and not equal to each other. For each of the five polygons, call all of the methods and display the results. For example for a `Polygon` with 3 sides, side length of 2.0 and x-coordinate and y-coordinates of 1.0, and apothem of 1.0 the following output may result:

```
toString() results: (numSides=3, sideLength=2.0,  
xcoord=1.0, ycoord=1.0, apothem=1.0)  
getNumSides() results: 3  
getSideLength() results: 2.0  
getXCoord() results: 1.0  
getYCoord() results: 1.0  
getApothem() results: 1.0  
getPerimeter() results: 6.0  
getArea() results: 3.0  
setNumSides(4) results: 4
```

```

setSideLength(3) results: 3.0
setXCoord(2.0) results: 2.0
setYCoord(2.0) results: 2.0
setApothem(2.0) results:2.0

```

Keep in mind, for five Polygons, you will have five different output results. Also, note there is no requirement to actually draw a Polygon.

Grading Rubric:

In all programming assignments the following grading rubric will be used to determine your grade:

Attribute	Exceeds	Meets	Does not meet
Design (5 points)	<p>(5 points)</p> <p>Exhibits proper use of parameters, and selection of data types all of the time.</p> <p>Employs correct and appropriate use of programming structures (loops, conditionals, classes etc.) all of the time.</p> <p>Efficient algorithms used all of the time.</p>	<p>(3-4 points)</p> <p>Exhibits proper use of parameters, and selection of data types most of the time.</p> <p>Employs correct and appropriate use of programming structures (loops, conditionals, classes etc.) most of the time.</p> <p>Efficient algorithms used most of the time.</p>	<p>(0-2 points)</p> <p>Rarely exhibits proper use of parameters, and selection of data types.</p> <p>Rarely employs correct and appropriate use of programming structures (loops, conditionals, classes etc.)</p> <p>Poorly structured and inefficient algorithms.</p>
Functionality (10 points)	<p>(9-10 points)</p> <p>Extra effort was apparent through the addition of significant and additional functionality beyond the scope of the assignment.</p>	<p>(7-8 points)</p> <p>Program fulfills most functionality.</p> <p>Most requirements were fulfilled.</p> <p>Screen captures provided demonstrating the successful compiling and running of the program.</p>	<p>(0-6 points)</p> <p>Program does not fulfill functionality.</p> <p>Few requirements were fulfilled.</p>
Test cases (5 points)	<p>(5 points)</p> <p>Test cases provide comprehensive</p>	<p>(3-4 points)</p>	<p>(0-2 points)</p>

	<p>coverage of all code paths.</p> <p>Discussion of run-time errors included.</p>	<p>Test cases provide coverage of most code paths.</p> <p>Test cases results well documented providing pass/fail results for each test case.</p>	<p>No or insufficient test cases</p> <p>Minimal supporting evidence provided to verify testing actually took place.</p>
Java Style Guide (5 points)	<p>(5 points)</p> <p>Code impeccably neat and well-organized.</p> <p>Extensive In-line comments providing additional insight into code design and functionality</p>	<p>(3-4 points)</p> <p>Header comments include filename, author, date and brief purpose of the program.</p> <p>In-line comments used to describe major functionality of the code.</p> <p>Meaningful variable names and prompts applied.</p> <p>Class names are written in UpperCamelCase.</p> <p>Variable names are written in lowerCamelCase.</p> <p>Constant names are written in All Capitals.</p> <p>Braces use K&R style.</p>	<p>(0-2 points)</p> <p>Code rarely follows recommended Java style guide</p>

Submission requirements

Your deliverables include all Java files (.java) and a single word (or PDF) document. The Java files should be named appropriately for your applications. Your word document should include screen shots showing the successful compiling and running of each application, and a detailed description of the test plan for your application. The test plan should include the input, expected output, actual output and if the test case passed or failed. Submit your files to the Homework 2 assignment area no later than the due date listed in the calendar. You should include your name and HW2 in your word (or PDF) file submitted (e.g. firstnamelastnamehw2.docx or firstnamelastnamehw2.pdf)