Lab 5 – Methods (14 pts)

Lab Objectives

- Be able to write methods
- Be able to call methods
- Be able to write `javadoc` comments
- Be able to create HTML documentation using the `javadoc` utility

~~Deliverables~~

~~This lab has four tasks. When you have all tasks done, run the report in Blackboard. The report is a Blackboard test with short-answer, file-response, multiple-answer, and other types of questions.~~

~~In the report, you may be asked to provide code segments, Java source code files (must have extension *.java*), screenshot of program execution, files in PDF format, and your analysis of the results.~~

~~If a short answer question requests a code segment, please ensure that your input is readable: all **new lines and indents** are in place.~~

~~Screenshots in your report **must show a full screen**, so your computer can be identified. Please resize your IDE panels the way that the required dialog or output is visible along with the source code. Show as much source code as possible.~~

~~NOTE:~~

- ~~Use **Blackboard only** to submit your work; **no email** submission unless your instructor directs it.~~
- ~~If Blackboard gives you multiple submission attempts (usually three), the **last one** will be evaluated and graded.~~
- ~~**No late submissions**, **no changes** in your submission after the due date.~~

Introduction

Methods are commonly used to break a problem down into small manageable pieces. A large task can be broken down into smaller tasks (methods) that contain the details of how to complete that small task. The larger problem is then solved by implementing the smaller tasks (calling the methods) in the correct order.

This also allows for efficiencies since the method can be called as many times as needed without rewriting the code each time.

Finally, we will use documentation comments for each method, and generate HTML documents similar to the Java APIs that we have seen.

During your lab class, try to complete Tasks #2-#4 for at least one of the six geometric figures. Later

you can just repeat the same procedures for the rest.

See textbook Ch. 5 and Lectures 4a, 4b for help.

Task #1 `void` Methods (4 pts)

1. Copy the file *Geometry.java* as directed by your instructor. This program will compile, but, when you run it, it doesn't appear to do anything except wait. That is because it is waiting for user input, but the user doesn't have the menu to choose from yet. We will need to create this.
2. Below the `main` method, but in the `Geometry` class, create a `static` method called `printMenu` that has no parameter list and does not return a value. It will simply print out instructions for the user with a menu of options for the user to choose from. The menu should appear to the user as:

```
This is a geometry calculator.
Choose what you would like to calculate.
1.  Find the area of a circle
2.  Find the area of a rectangle
3.  Find the area of a triangle
4.  Find the circumference of a circle
5.  Find the perimeter of a rectangle
6.  Find the perimeter of a triangle
Enter the number of your choice:
```

3. Add a line in the `main` method that calls the `printMenu` method as indicated by the comments.
4. Compile, debug, and run. You should be able to choose any option, but you will always get 0 for the answer. We will fix this in the next task.

Task #2 Value-Returning Methods (4 pts)

1. Write a `static` method called **`circleArea`** that takes in the radius of the circle and returns the area using the formula $A = \pi r^2$.
2. Write a `static` method called **`rectangleArea`** that takes in the length and width of the rectangle and returns the area using the formula $A = lw$.
3. Write a `static` method called **`triangleArea`** that takes in the base and height of the triangle and returns the area using the formula $A = \frac{1}{2}bh$.
4. Write a `static` method called **`circleCircumference`** that takes in the radius of the circle and returns the circumference using the formula $C = 2\pi r$.
5. Write a `static` method called **`rectanglePerimeter`** that takes in the length and the width of the rectangle and returns the perimeter of the rectangle using the formula $P = 2l + 2w$.

6. Write a `static` method called **`trianglePerimeter`** that takes in the lengths of the three sides of the triangle and returns the perimeter of the triangle which is calculated by adding up the three sides.

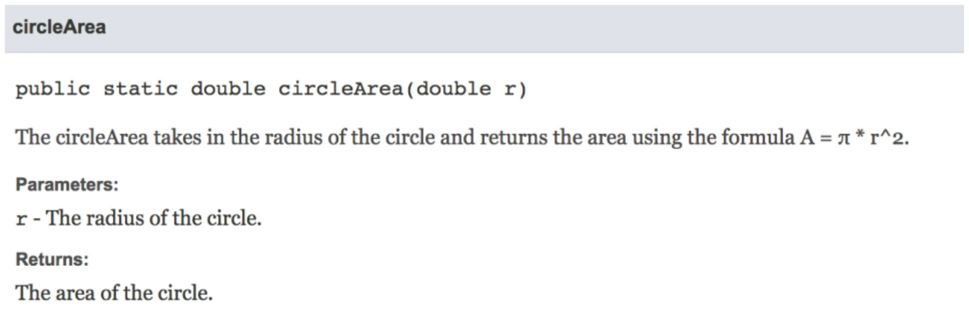
Task #3 Calling Methods (4 pts)

1. Add lines in the `main` method in the `Geometry` class which will call these methods. The comments indicate where to place the method calls.
2. Create a table with sample data and hand calculated results for all 6 menu items.
3. Compile, debug, and run.
4. Test out the program using your sample data and write computer results next to hand calculated results.
5. Put the table with sample data, hand calculated results and computer results in your report.

Task #4 Java Documentation (2 pts)

1. Write `javadoc` comments for each of the **seven** `static` methods you wrote. They should include[1]:
   a. A short summary of what the method does.
   b. A description of what the program requires to operate and what the result of that operation is.
   c. `@param` listing and describing each of the parameters in the parameter list (if any).
   d. `@return` describing the information that is returned to the calling statement (if any).
2. Generate documentation for your project. (At the end of Lecture4a-Methods slides, you can find the ways to run Javadoc command depending on the development tool you use. In Eclipse, it is *Generate Javadoc* in the *Project* menu.) By default, the documentation will be saved in a new folder *doc*.
   2.1. Use your file manager to find file *index*.html in the folder *doc.*
   2.2. Open *index.html* in your browser. Find out how text and tags in comments are shown in the documentation.
   2.3. Check the methods summaries and details to ensure your comments were put into the documentation correctly. (The picture below shows a fragment of possible description generated for one of the methods.)

[1] The example can be found in *SalesReport.java* posted on Blackboard with Lecture 4b.



circleArea

```
public static double circleArea(double r)
```

The circleArea takes in the radius of the circle and returns the area using the formula A = π * r^2.

Parameters:

`r` - The radius of the circle.

Returns:

The area of the circle.

2.4. Print your screen as a PDF file using the browser's menu (or shortcut key Ctrl-P / Command-P).