**Name:**

**CST 183 Lab Assignment #5 (Chapter 5)**

**Methods**

**Lab Objectives**

Be able to write methods

Be able to call methods

Be able to write javadoc comments

Be able to create HTML documentation for our Java class using javadoc

**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.

**Task #1 void Methods**

1. Copy the file Geometry.java (code listing 5.1) from the Student CD or 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. Above 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:
7. Add a line in the main method that calls the printMenu method as indicated by the comments.
8. 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**

1. Write a static method called **circleArea** that takes in the radius of the circle and returns the area using the formula A = π*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 = ½*bh*.
4. Write a static method called **circleCircumference** that takes in the radius of the circle and returns the circumference using the formula C = 2π*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 = 2*l +*2*w*.
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**

1. Add lines in the main method in the GeometryDemo class which will call these methods. The comments indicate where to place the method calls.
2. Compile, debug, and run. Test out the program using your sample data.

**Paste in the source code of your Geometry.java program here:**

**Paste in a screensot of your Geometry.java program execution here:**

**Task #4 Java Documentation**

1. Write javadoc comments for each of the 7 static methods that you just wrote. They should include
   1. A one line summary of what the method does.
   2. A description of what the program requires to operate and what the result of that operation is.
   3. @param listing and describing each of the parameters in the parameter list (if any).
   4. @return describing the information that is returned to the calling statement (if any).
2. Generate the documentation. Check the method summary and the method details to ensure your comments were put into the Java Documentation correctly.

**Paste in a screenshot of your Geometry.java program documentation here:**