CPSC 1181 - Assignment 1 [50 marks

This assignment is to be completed individually.

Instructions:

- When you download a zip file with java files, extract all of the java files before copying them into Eclipse project.
- Create a new project in Eclipse: File -> New -> Java Project. Name the project A1 and select finish.
- Import the java files *by clicking and dragging* ***them directly onto the SRC directory*** *in your A1 Eclipse Project*. You will be asked if you wish to copy or link these files, *select copy.*
- Complete the exercises described below.

Submission

- Submit a **zip file** containing **only** Calculator.java and TestCalculator.java to Brightspace prior to the **due date set in Brightspace**. Do not include class files
- Submissions that are less than 24 hours late receive a 1% per hour late penalty. Submission that are more than 24 hours late will not be accepted.
- You can use one of your 2 day extensions.
    - If you submit more than 24 hours late, one will be used automatically
    - **Add a comment to your submission in Brightspace IF**
        - You want to use an extension for a submission less than 24 hours late
        - You are submitting more than 24 hours late, but **don't** want to use an extension…you just want feedback.
- Submissions that are unzipped or that contain .class or other unneeded files will be penalized.

Exercise 1 [12 marks]

On D2L download the file A1.zip. Extract the file and import Calculator.java into Eclipse. Notice that the provided class has almost no documentation. Write the appropriate JavaDoc style documentation describing the purpose, input, and output of each method. **All files must contain the @author information for the class.** The TestCalculator and other test classes do not need method documentation, but still should have the class-level documentation.

Exercise 2 [23 marks]

In this exercise, you will write the unit tests to ensure that all of the methods of the class work correctly. Right click on your project and go down to Build Path -> Add Libraries. Select JUnit and hit Next. Make sure Junit 5 is selected and hit Finish. If you don't see Add Libraries, select Configure Build Path, the Libraries Tab, and the Add Library button to select Junit.

Open up the file TestCalculator.java. You will see a series of empty test cases. Each test case should be used to test one method from Calculator.java. You will see that testAdd has been started for you. Right click on TestCalculator -> Run As -> Junit Test. You will see that currently these tests all pass.

Fill in the remaining test cases to ensure that all of the provided methods work as expected for different test cases. If your testing is done correctly, it should help you identify any methods that are incorrectly implemented.

Notes

- The remainder function should only accept and return int. Don't change the type.
- The remainder function should throw an ArithmeticException if the second value is 0. Use a test to make sure it does.

- The divide method uses doubles. There is no exception for divide by 0. The result should be Double.POSITIVE_INFINITY or Double.NEGATIVE_INFINITY

Exercise 3 [10 marks]

- Fix the method(s) in the Calculator class. This includes checking for null when necessary. [10 marks]

Style Guidelines [5 marks]

Ensure Calculator.java and TestCalculator.java have followed good coding style and conventions (see the style guidelines document in Brightspace and Appendix E at the back of the textbook). [5 marks]