Langara College

# Department of Computing Science & Information Systems

# CPSC1181 – Object-oriented Computing

###### **Lab2: JUnit Test**

**Objectives:**

* Set up JUnit test
* Implement JUnit test methods
* Handle null parameter and exceptions

**[45 marks]**

**Instructions:**

1. Download **Lab2Files.zip** from D2L and unzip it
2. Open Eclipse and make **LabProjects** as the workspace.
3. Inside Eclipse, create a project named **Lab2** to store all the files for this lab. Make sure uncheck **Create module-info.java file**.

**Part A: [10 marks] Add Javadoc Documentations**

1. Inside Eclips, right-click project Lab2->New->Source Folder and name it **src**
2. Right-click **src** to add a new package and name it **lab2**
3. Right-click the package name **lab2** to import the two files **Calculator.java** and **TestCalculator.java** inside the folder **Lab2Files**
4. Add **package lab2**; to the beginning of the two Java files
5. Right-click the project name Lab2->Build Path->Add Library…. Select JUnit and hit Next. Make sure JUnit 5 is selected and hit Finish.
6. Add @author to each of the two Java files
7. For the **Calculator** class, add purpose, @param, and @return if any to each method
8. For the **TestCalculator** class, no need to add any Javadoc comments to any method.

**Part B: [25 marks] Implement all the test methods**

1. In this part, you will write the unit tests to ensure that all of the methods of the class work correctly.
2. 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.
3. 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.
* **No other function should throw an exception.** In all functions, return a specific value (you choose) if there is a problem with the input.
* The divide method uses doubles. There is no exception for divide by 0. The result should be Double.POSITIVE\_INFINITY or Double.NEGATIVE\_INFINITY

**Part C: [10 marks] Fix the bugs in all the methods**

Fix the method(s) in the Calculator class. This includes checking for null where necessary.

**What to hand in**

1. Click Project->Generate Javadoc… to geneate the Javadoc documents for all the java files inside Lab2->src->lab2. Leave all the other selections unchanged.
2. Zip the folder **Lab2** and upload it to D2L.

**When to hand in**

By 11:59pm, Wednesday, January 18, 2023.