OBJECT ORIENTED PROGRAMMING (JAVA) (CST8284)

LAB 4

UNIT TEST FRAMEWORK (Java Unit Test – JUnit 4)

Marks: 5%

Important Information

To be successful in this lab:

❖ Ensure that you **review the hybrid tasks** for Week #4 **first before** starting off with this lab exercise.

❖ Review the **helpful video** included in the **Hybrid for Week #4** in Brightspace page to enrich your learning experience.

ALGONQUIN
COLLEGE

Some Introduction

The essence of this lab is to learn about, and use a **unit testing** framework.

❖ Unit testing frameworks are used to **execute** and **evaluate** test suites, as well as make it easy to add test cases.

❖ In this lab we will use the **JUnit 4**

❖ Available at: https://junit.org/junit4/

You do not need to install JUnit if using Eclipse

Some Introduction (2)

❖ In **JUnit** testing, you design a **test case** for **each class** you develop.

❖ A **method** is provided for each of the test cases you wish to run.

❖ Test methods are annotated. **Annotations** are used to **mark** the test methods (advanced Java features that place markers in the code, which is interpreted by another tool).

❖ In JUnit, **@Test** annotation is used to mark methods. Other methods may exist in the text class without @Test (e.g. methods performing steps to be shared among test methods).

Some Introductions (3)

❖ For each test case, conditions can be computed to check if **true**.

❖ Result is passed to a method ($\mathrm{assertEquals}$ method) that sends it to the framework.

❖ The $\mathrm{assertEquals}$ method takes the **expected** and **actual** values as **arguments** (for floating point numbers it takes a **tolerance** value). Method fails if the two given values are unequal.

❖ It is **good practice** to end the name of the test class with "**Test**", for example: $\mathrm{CovidTest}$

Some Introductions (4) - Assertions and their definitions

From your Hybrid tasks for this week, what **assertions** can you **identify** that are used in JUnit testing methods? What are their descriptions?

Hint: assertNull(…), assertNotNull(…), assertTrue(test), assertFalse(test), assertSame(…), etc.

PART 1

YOUR TASKS USING THE CODE PROVIDED

You are Required to…

❖ **Inspect** the code files that you received and **load them** into eclipse (Lab 4)

❖ Check to ensure that you have **JUnit 4** appropriately **set up** in your eclipse.

❖ Do the following if you need to **correctly set up JUnit 4** in eclipse:

➢ Go to **Package Explorer** at the top bar of your eclipse editor

ALGONQUIN COLLEGE
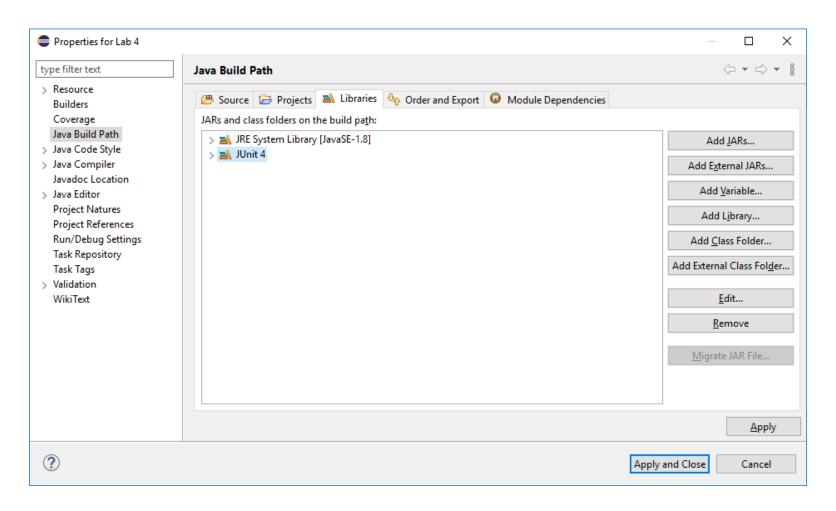
You are Required to… Ensure Correct set up of JUnit 4

- ➢ Click on **Project**

- ➢ Scroll down to **Properties** on the menu

- ➢ Scroll down to **Java Build Path** on the menu and click on it.

- ➢ From **Libraries,** Select **Add Library (**from the list on the right-hand side of that page)

- ➢ Select **Junit (and then JUnit 4** from menu**)**

- ➢ Click **Apply and Close**

ALGONQUIN
COLLEGE

You are Required to… Ensure Correct set up of JUnit 4 (2)
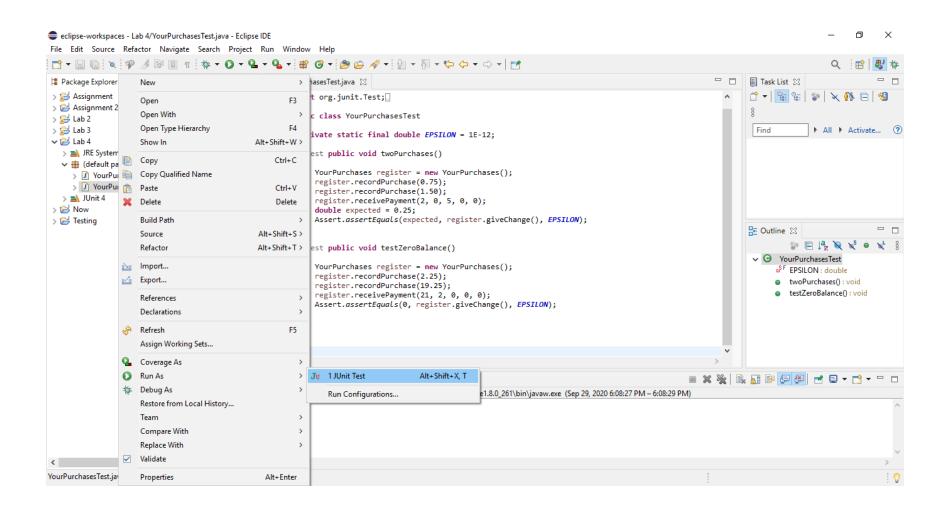
Execute your Code as Junit Test

❖ To execute, **right click** on the **YourPurchasesTest** and select **Run As,** and then **JUnit Test** (as shown in slide 13)

❖ When your execution is **successful**, a **green bar** shows in the panel in eclipse, cataloging **Runs, Errors and Failures** (see slide 14)

❖ When **unsuccessful**, a **red bar** shows in the panel and the catalog of Runs, Errors and Failures (see slide 15)

Execute as JUnit Test

When your run is successful…

Error is encountered: unsuccessful run

Demo: Part 1

❖ **Run** the code provided to you (**YourPurchases.java** and **YourPurchasesTest.java**) to show **GREEN BAR**

➢ **Explain** the output on your eclipse screen to your professor

❖ **Run** the code provided to you (**YourPurchases.java** and **YourPurchasesTest2.java**) to show **RED BAR**

PART 2

CREATING A NEW TEST CASE
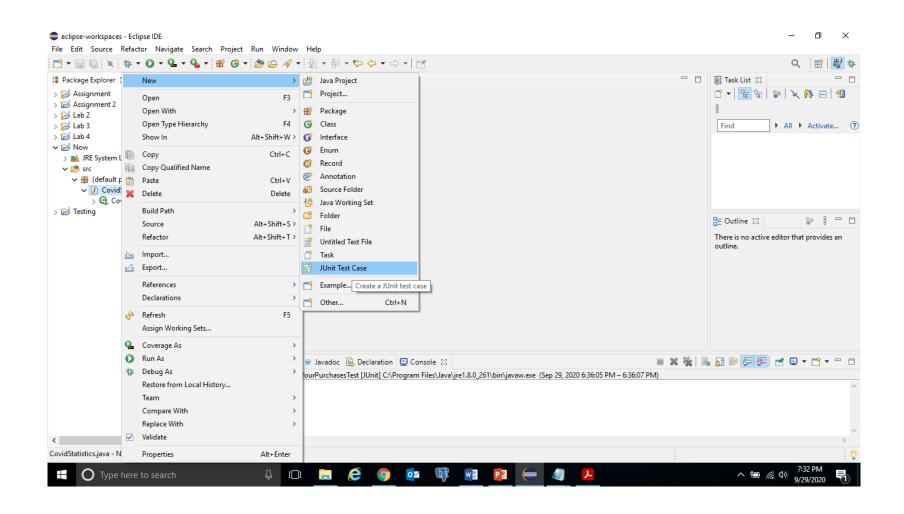
Create a New Test Case

❖ In your eclipse editor, select the source code for which the Test case is required (use **YourPurchases.java**)

❖ Make a **right click** on the source code file to display a long menu window (as shown in next slide).

❖ At the **top** of the menu, select **New,** and then scroll down to select **Junit Test Case**

Creating a New Test Case (2)

Creating a New Test Case (3)

❖ In the New JUnit test case window, ensure to select the **New Junit 4 Test**

❖ Select the **Package** (default packages are discouraged – always name your package

❖ Enter the **Name** (ending with "Test")

❖ Click **Next** to select the **methods** for which your test cases are required (see picture slide below) and then click **Finish**

ALGONQUIN COLLEGE

Creating a New Test Case (4) - Example

Creating a New Test Case (5) - Example

Demo for Part 2 – Create Example Test Case

Using what you learned in slides 18 – 22:

❖ **Demonstrate** to your Professor **how you will create** a **new** Test Case still using (**YourPurchases.java**)

❖ Use JUnit test cases to find the source of the test failure. You may do this systematically by creating a test testing one method, then add another test for two methods (the previous successful one and another method). Do this until you get a test failure. Then, last added method is the reason for the failure.

Grading Rubrics

❖ Successfully configuring your Junit 4 test environment and running the provided code to show **GREEN BAR** – 0.5%

❖ Successfully introducing an error/failure into the code provided to show **RED BAR** – 0.5%

❖ Successfully **writing your own tests** into the **YourPurchasesTest** code file (see instructions in the code) – 1.5%

❖ Successfully **creating a new test suite** and **answering** relevant questions – 2%

❖ Correctly **including Javadoc comments** into the test file provided to you – 0.5%

References

❖ Java How to Program, Early Objects Plus MyProgrammingLab with Pearson eText -- Access Card Package, 11/E. Author: Deitel ISBN: 9780134800271

❖ Big Java Early Objects, 7/E. Author: Horstmann, C. Wiley. ISBN: eText: 978-1-119-49909-1 or loose-leaf paper: 978-1-119-74020-9.