

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.



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 (assertEquals method) that sends it to the framework.
- The 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: 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

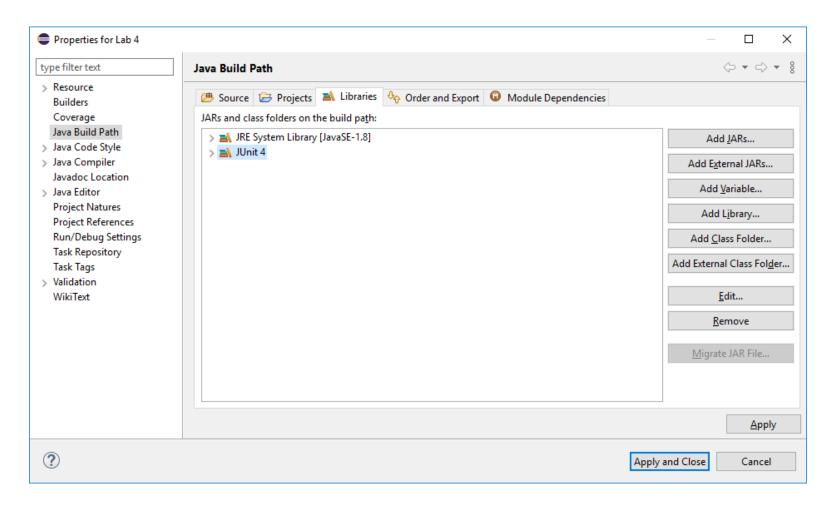


You are Required to... Ensure Correct set up of JUnit 4

- 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



You are Required to... Ensure Correct set up of JUnit 4 (2)



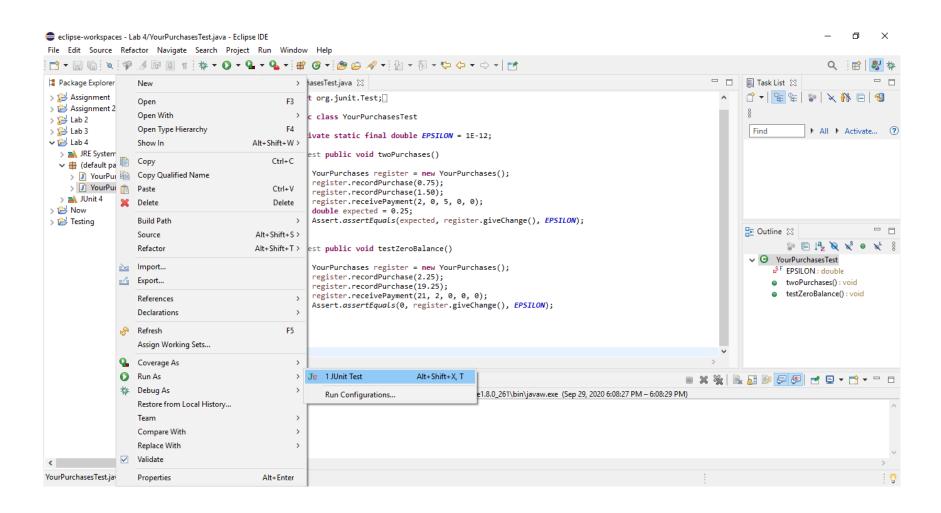


Execute your Code as Junit Test

- 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)

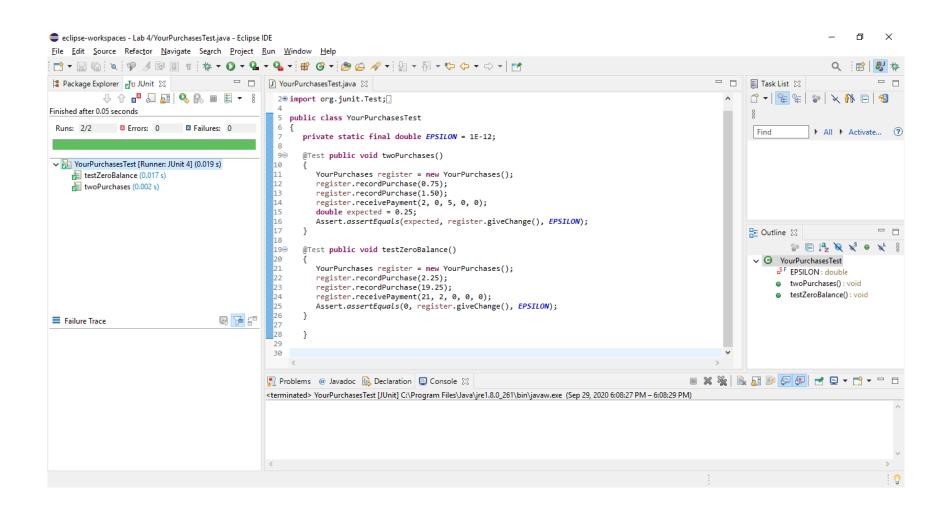


Execute as JUnit Test





When your run is successful...





Error is encountered: unsuccessful run

```
U YourPurchasesTest2.java × $ □ □ D YourPurchasesTest2.java ×
nished after 0.017 seconds
                                                                    > $\mathcal{B}$ Lab4 > $\mathcal{B}$ src > $\mathcal{B}$ fall22lab4 > $\mathcal{O}$ YourPurchasesTest2 >
                                                                     1 package fall22lab4;
                                                                      2⊕import static org.junit.Assert.*;[
/ Big fall22lab4.YourPurchasesTest2 [Runner: JUnit 4] (0.000 s)
                                                                        public class YourPurchasesTest2 {
   testGiveChange2 (0.000 s)
                                                                     9
                                                                             private static final double EPSILON = 1E-12;
                                                                    10
                                                                    11
                                                                             //THIS TEST WOULD FAIL.
                                                                   @12
                                                                             // TODO: WRITE ONE OR MORE TESTS TO SYSTEMATICALLY FIND THE SOURCE OF THE TEST FAILURE
                                                                    13⊝
                                                                             @Test
                                                                    14
                                                                             public void testGiveChange2() {
                                                                    15
                                                                                   YourPurchases aPurchase = new YourPurchases();
                                                                    16
                                                                                   aPurchase.recordPurchase(1.5);
                                                                    17
                                                                                   aPurchase.receivePayment(5, 0, 0, 0, 0);
                                                                                   double changeResult = aPurchase.CalculateChange();
                                                                    18
                                                                    19
                                                                                   double expected = 0.25;
                                                                                   Assert.assertEquals(expected, changeResult, EPSILON);
                                                                    20
                                                           💂 🍞 🚌 📗 21
                                                                                   assertTrue(true);
Failure Trace
                                                                    22
| java.lang.AssertionError: expected:<0.25> but was:<6.5>
                                                                    23
at fall22lab4.YourPurchasesTest2.testGiveChange2(YourPurchasesTest2.java:20)
                                                                    24
                                                                   📳 Problems @ Javadoc 🚇 Declaration 🔗 Search 📮 Console 🗴 🔧 Breakpoints 🍰 Call Hierarchy 🚳 Palette
                                                                   <terminated> YourPurchasesTest2 (1) [JUnit] C:\Users\saadawh\.p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86 64 11.0.13.v20211116-1829\jre\bin\javaw.exe (Sep. 27, 20)
```



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 NEW TEST CASES



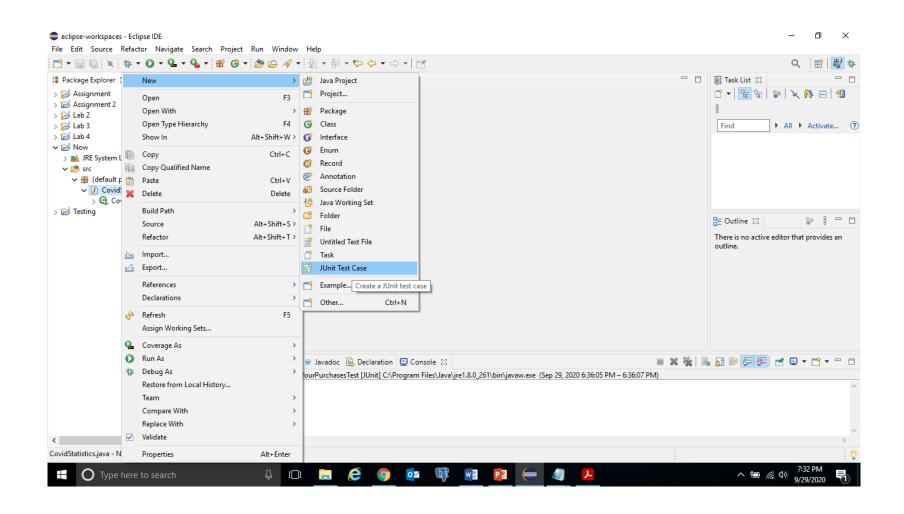


Create a New Test Case

- 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

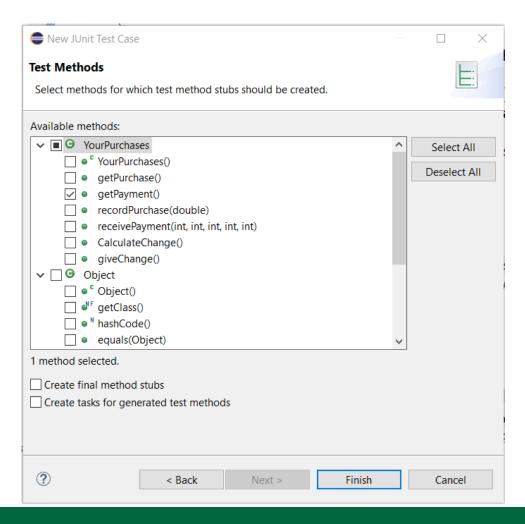


Creating a New Test Case (4) - Example

	New JUnit Te	st Case	\Box \times
		e of the new JUnit test case. Specify the class under test to to be tested on the next page.	
	O New JUnit 3 to	est New JUnit 4 test New JUnit Jupiter test	
	Source folder:	Lab4/src	Browse
	Package:	w23lab4	Browse
	Name:	YourPurchasesTest3	
	Superclass:	java.lang.Object	Browse
	Which method stubs would you like to create? @BeforeClass setUpBeforeClass()		
าร		w23lab4.YourPurchases	Browse
, ,	?	< Back Next > Finish	Cancel



Creating a New Test Case (5) - Example





Part 2 – To Be Announced in Lab Session



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 tests as needed in part 2 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.

