

# CST8132 Object Oriented Programming

## Exercise 3 – Using JUnit Testing

**Due in the lab period of week 4**

### Steps:

- For this lab exercise you are given a starter code for a class that implements a modified approach to solving the Calculator problem from lab exercise #1. For a quick-start guide, refer to the uploaded document “JUnit Testing.pdf”.
- The Eclipse IDE already has JUnit functionality (no need to install additional .jar files and add them to a build path). You can verify the availability of JUnit by right clicking on your project (i.e. in the project directory display) and choose “Properties”. From the menu of the left side of the “Properties” window, choose “Java Build Path”. Then choose the “Libraries” tab. You should see “JUnit 4” in the “JARS and class folder on the build path:” window.
- Import the starter code into a new Java project.
- Organize your JUnit test cases in separate package: right click on your project and choose “New -> package”. Name this package “calculatortest”.
- Right click on the “calculatortest” package and choose “New -> Junit Test Case”. If there are multiple classes in the project, you may want to specify which class is being tested in the “Class under test:” text field.
- Add the JUnit test cases as specified below:

```
i.    public void testAdd() //invoke the add() method
ii.   public void testDivideByZero() //use Double.POSITIVE_INFINITY in the assert()
iii.  public void testDivideBySelf() //tests any number divided by itself – should be equal to 1
iv.   public void testMultiply() //choose meaningful numbers to test
v.    public void testMultiplyForInfinity()//use Double.POSITIVE_INFINITY in the assert()
                                           // and Double.MAX_VALUE in the constructor and
                                           //multiply() method

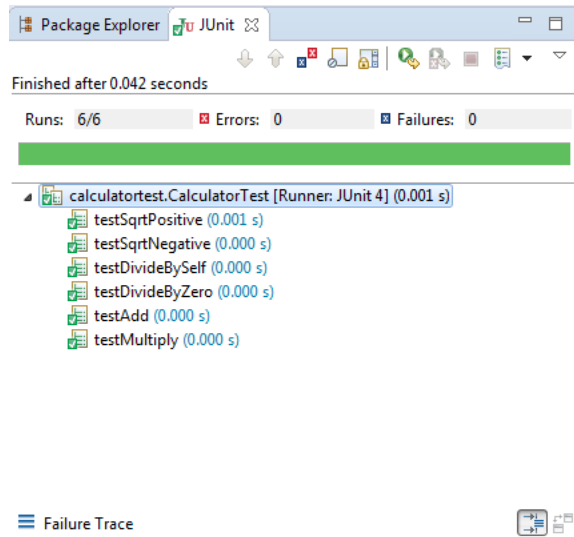
vi.   public void testSqrtPositive()
vii.  public void testSqrtNegative()//use Double.NaN in the assert()
```

- Each JUnit test case should have:
  - i. A Calculator reference instantiated with an appropriate constructor.
  - ii. An appropriate “assert” statement.
  - iii. Invocations of methods from the Calculator class as needed.
  - iv. Annotate each test case with @Test

### Grading Guide (Total Score 15):

Name class and package correct	1
All test cases implemented with local reference instantiated	7
All test cases contain correctly implemented asserts	7

## Sample Program Execution Screen Capture: Test Success



## Sample Program Execution Screen Capture: Test Failure

