Today’s topic:

- Completing Unit testing of classes and methods.
- Using JUnit for unit testing in Java.
Unit Testing

Most of our classes will *not* have a main method included in them.

Therefore to test the methods in a class it is beneficial to complete *unit test* to validate that the class works as it should.

Unit Testing

Unit testing is designed to test a *unit* of work, rather than the *entire system*.

If each class is *independently* tested, we can be assured that it will *work* in as described by all programs that use the class.

We have been using classes that have been *provided* to us, we would *expect* they have passed unit testing so we are *confident* to use them.
JUnit

- JUnit is a **specialized** group of **classes** that help facilitate **unit testing**.
- Most IDEs (including Eclipse) install the JUnit classes automatically.
- If you want to download and run JUnit from the command line, you can find information on how to do so on the resources page of the class website.

Using JUnit in Eclipse

- Turn on the JUnit window.
  - **Window** → **Show View** → **Other...**
Using JUnit in Eclipse

- Turn on the JUnit window.
  - Select: Java → Junit
  - Click OK

The window is now visible by the project explorer.
Using JUnit in Eclipse

Right click on the class to test.
- New → JUnit Test Case

Select the method stubs to create and click Next.
Using JUnit in Eclipse

4. Select the class methods stubs to create and click Finish.

Add the JUnit library to the build path, if you haven’t already.
Using JUnit in Eclipse

- Finish writing your test cases.

- Run the test case.

- View the results.
Using JUnit in Eclipse

If you want to run multiple test cases, you can do so with a Test Suite.

Right click on the package
- New → Other...

Using JUnit in Eclipse

- Right click on the package
  - Select Java → JUnit → JUnit Test Suite
  - Click Next
Using JUnit in Eclipse

- Select the Test classes you want to include.
  - Click Finish

Using JUnit in Eclipse

- Run again using AllTests
JUnit Testing Class

- **setUp() Method**
  - *Runs prior to test methods*
  - Uses `@Before`

- **tearDown() Method**
  - *Runs after all test methods*
  - Uses `@After`

- **test Methods**
  - *Complete any tests you write*
  - *Can have multiple tests*
  - Uses `@Test`

---

JUnit Testing Class

- Utilize delivered assert methods:
  - *Consult the API for the Assert class in JUnit*

  [http://junit.sourceforge.net/javadoc/org/junit/Assert.html](http://junit.sourceforge.net/javadoc/org/junit/Assert.html)