Lab TDD 3: Adding a Test Case

Objectives

In this lab you will add run a test case

The lab continues from where TDD 2 lab leftoff

Instructions Project Setup

Step 1: Add a multiply method

1. You are going to add a test method to implement the test case subtracting a number from itself. In this case 2 – 2 == 0
2. Rename one of the test methods as testSub1() and delete any other test methods
3. Implement the test method as shown below

```
@Test
public void testSub1() {
    int result = TestClass.c.sub(2, 2);
    assertEquals("Error message",0, result);
}
```

4. Run the tests and notice that this test passes. The reason is that the default dummy return value just happens to be the same as the test expected value. The test is broken
5. Fix the method stub to return -999 which we know is not going to be an expected value for any of our tests

```
@Override
public int sub(int a, int b) {
    return -999;
}
```

6. Rerun the test and see it fail

Step 3: Add the production code

1. Add the code shown which is wrong.
2. Rerun the tests and see them pass

```
@Override
public int sub(int a, int b) {
    return b-a;
}
```

3. Notice that this is the wrong code but the test just happens to pass. This is a false negative and due to coincidental correctness where the wrong code produces a pass for one test case. However, this can be identified by adding a second test case that tests the same code.

Step 4: Add a second test case

1. Add a second test case for “2 -3 == -1”

```
@Test
public void testSub2() {
    int result = TestClass.c.sub(2, 3);
    assertEquals("Error message",-1, result);
}
```

2. Run the tests and now the first test fails but the second one passes.
3. Correct the incorrect code and run all the tests again and ensure they pass

Save your code since you will use it in the next lab