CIS 1068

February 27, 2018

Administrative Stuff

- Assignment 6
- Thursday, office hours
  - rescheduled: 12:30-1:50
  - or appointment, or drop by
- Midterm 1

Testing

```java
public static int sum(int a, int b) {
    return a+b;
}
```

- really simple, but pretend it’s a more complicated method
- how would you test it?
Testing

```java
int x=2, y=3, expectedResult=5, actualResult;

actualResult=sum(x, y);
if (expectedResult!=actualResult) {
    System.out.println("error. added " + x + " and " + y + ".
    Expected " + expectedResult + ", but found " + actualResult);
} else {
    System.out.println("error. added " + x + " and " + y + ": OK");
}
```

```java
x=5; y=0; expectedResult=5;
actualResult=sum(x, y);
if (expectedResult!=actualResult) {
    System.out.println("error. added " + x + " and " + y + ".
    Expected " + expectedResult + ", but found " + actualResult);
} else {
    ...
```

Functions We'd Like To Test

```java
public class Stuff {
    public static int sum(int a, int b) {
        return a+b;
    }

    public static boolean isEven(int x) {
        return x%2==0;
    }
}
```

in Netbeans, right click (or CTRL-click on Mac)

Find Generated Test-File Under Test Packages
Test Code

```java
/**
 * Test of sum method, of class Stuff.
 */
@Test
public void testSum() {
    // ...
}

/**
 * Test of isEven method, of class Stuff.
 */
@Test
public void testIsEven() {
    // ...
}
```

We Fill in Our Implementation

```java
/**
 * Test of sum method, of class Stuff.
 */
@Test
public void testSum() {
    assertEquals(5, Stuff.sum(2, 3));
    assertEquals(5, Stuff.sum(5, 0));
    assertEquals(-5, Stuff.sum(-2, -3));
}

/**
 * Test of isEven method, of class Stuff.
 */
@Test
public void testIsEven() {
    assertTrue(Stuff.isEven(0));
    assertTrue(Stuff.isEven(28));
    assertTrue(Stuff.isEven(8));
    assertFalse(Stuff.isEven(7));
}
```

We Fill in Our Implementation

```java
/**
 * Test of sum method, of class Stuff.
 */
@Test
public void testSum() {
    assertEquals(5, Stuff.sum(2, 3));
    assertEquals(5, Stuff.sum(5, 0));
    assertEquals(-5, Stuff.sum(-2, -3));
}
```

Testing for Even

```java
/**
 * Test of isEven method, of class Stuff.
 */
@Test
public void testIsEven() {
    assertTrue(Stuff.isEven(0));
    assertTrue(Stuff.isEven(28));
    assertTrue(Stuff.isEven(8));
    assertFalse(Stuff.isEven(7));
}
```

notice that we are:

- calling the method `sum` which lives in the class `Stuff`
Handy Test Functions We’ve Used So Far

**assert Equals**

```java
assertEquals(expectedValue, FunctionToTestWithArgs);
assertEquals(5, TheClassName.sum(2,3));
```

**assert True**

```java
assertTrue(FunctionToTestWithArgs);
assertTrue(TheClassName.isEven(16));
```

**assert False**

```java
assertFalse(FunctionToTestWithArgs);
assertFalse(TheClassName.isEven(3));
```