CIS 1068

February 27, 2018
Administrative Stuff

- Assignment 6
- Thursday, office hours
  - rescheduled: 12:30-1:50
  - or appointment, or drop by
- Midterm 1
Histogram of 1068 Midterm 1

μ = 71.2, σ = 21.6, median = 76.0
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

... 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");
}

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 {
    ...
}
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

- Stuff
  - Source Packages
    - stuff
      - Stuff.java
  - Test Packages
    - stuff
      - StuffTest.java
  - Libraries
  - Test Libraries
/**
 * 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));
}
```
We Fill in Our Implementation

/**
 * 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));
}

notice that we are:

- calling the method `sum`
- which lives in the class `Stuff`
Testing for Even

/**
 * 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));
}
Handy Test Functions We’ve Used So Far

**assertEquals**

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

**assertTrue**

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

**assertFalse**

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