Lab 2

This week you will be demoing the work you did on the scapbook you created last week.

Demo Points

1. Declare an integer variable called n. (1 Point)
2. Set the n to 5; (1 Point)
3. Declare an integer array called a. (1 Point)
4. Write a loop that initializes a to the squares of the first five integers. (2 Point)
5. Write a loop that calculates the average of the values of a. (2 Point)
6. Write a loop that will move the items in array a into another array called b so that the order of b is the reverse of the order of a. (3 Point)
7. Write a loop that will calculate the first 10 Fibonacci numbers. (5 Point)

Next Week

Next week you will be implementing the Point class that was described in lecture. You will also create a Circle class that uses a Point as the center of the circle and contains a new floating point member variable to represent the radius of the circle.

Here are the JUnit Tests for the Point Class:

```
package oop;

import static org.junit.Assert.*;

public class TestPoint {

    @Test
    public void testConstructor() {
        // Point has an X and a Y value
        Point p = new Point(2, 3);
        assertEquals(2, p.getX());
        assertEquals(3, p.getY());
    }

    @Test
    public void testMove() {
        Point p = new Point(2, 3);
        assertEquals(2, p.getX());
        assertEquals(3, p.getY());

        p.move(4,5);
        assertEquals(4, p.getX());
        assertEquals(5, p.getY());
    }

    @Test
    public void testToString() {
        Point p = new Point(2, 3);
        assertEquals("p(2, 3)", p.toString());
    }

}
```

Here are the JUnit tests for the Circle Class:

```
package oop;

import static org.junit.Assert.*;

public class TestCircle {

    @Test
    public void testConstructor() {
        Point p = new Point(1, 2);
        Circle c = new Circle(p, 1.5);
        assertEquals(c.getCenter(), p);
        assertEquals(c.getRadius(), 1.5, .001);
    }

    @Test
    public void testToString() {
        Point p = new Point(1, 2);
        Circle c = new Circle(p, 1.5);
        assertEquals(c.toString(), "Circle [center=p(1, 2), radius=1.5]");
    }

    @Test
    public void testMove() {
        Circle c = new Circle(new Point(1, 2), 1.5);
        assertEquals(c.move(new Point(3, 4)).toString(),
                "Circle [center=p(3, 4), radius=1.5]");
    }

    @Test
    public void testEnlarge() {
        Circle c = new Circle(new Point(1, 2), 1.5);
        assertEquals(c.enlarge(2.5).toString(),
                "Circle [center=p(1, 2), radius=2.5]");
    }
}
```

Complete stories 2 – 7 using Test Driven Development. To help you out this week, we have included the JUnit tests you will need to run. Write one of the test and get it to work before adding the next test. *Do not type in all of the tests at once, then start debugging.* If you write all of the tests first, you will waste hours debugging your tests before you can even begin writing your program. If, despite this warning, you type in all of the test, your best solution is to delete everything you have written and start again. You may, of course, try to write all of the code at once, but life is short and wasting it on frustrating, counterproductive activities is foolish.

Getting the JUnit tests to work earns the two points for the unit tests. To get the points for the system tests. Open a Scrapbook page, create a Point (or Circle) and call the method.

Next Weeks Demo Points

1. Story 2: Point
    a. Passes JUnit tests (2 marks)

b. Execute System test (3 Marks)

2. Story 3-7: Circle
    a. Passes JUnit tests (2 marks)
    b. Execute System test (3 Marks)