Jlaja Pandora Podier

Robert Bathmann

COSC 603: Software Testing and Maintenance

Spring 2016

Project 3 Unit Testing with JUnit

**Task 2 Getting Started – Fibonacci**

The Fibonacci JUnit test case failed on the equality test case for the zero (0) test case in which a zero was expected but the value returned was one (1). The JUnit Test code is as follows: *assertEquals*("0", 0, fibonacci.fibonacci(0)); but the switch statement code for calculating and returning the nth Fibonacci number code for the zero case had a return of one (1) as oppose to returning zero in which that threw an error because it was expecting a return value of zero but the code had it returning one (1). In order to correct the error, the switch statement had to be corrected for the zero (0) case to return a value of one (1).

**public** **int** fibonacci(**int** n) {

**switch** (n) {

//case 0: return 1;

**case** 0: **return** 0;

**case** 1: **return** 1;

**default**: **return** (fibonacci(n - 1) + fibonacci(n - 2));

}

}

}

**Task 3 – A Little More Advanced – Rectangle**

When the RectangleTest JUnit test case was ran, there was an error that occurred in the software with the GetArea and Get Diagonal in which the value returned was not the expected value for the calculation for the Area or the diagonal. After investigating the source code, we determined that the error was in the Point class in which the instantiation of the new point set the values for both x and y coordinates to the ‘y’ coordinate. The statement that had to be changed was the **this**.x = y to **this**.x = x because the ‘x’ coordinate had to be set for ‘x’ and the y coordinate for y.

Point(Double x, Double y) {

//this.x = y; //This is the error in the code because it sets the x value to the y coordinate

**this**.x = x;

**this**.y = y;

}

}

**Task 4 – On Your Own – A Vending Machine**