**Task 2:** The error in task 2 involved the switch statements. The fib sequence is fib(0) = 0 not 1 so even though you are testing the correct thing, the actual code was giving the error.

**Task 3:** The error in task 3 involved the constructor of the point class putting ‘Y’ inside both x and y.

/\*\*

\* Gets the area.

\*

\* **@return** the area

\*/

**public** Double getArea() {

**double** xDiff = p2.getX() - p1.getX() ;

**double** yDiff = p2.getY() - p1.getY();

**return** Math.*abs*(xDiff \* yDiff );

}

/\*\*

\* Gets the diagonal.

\*

\* **@return** the diagonal

\*/

**public** Double getDiagonal() {

**double** xDiff = p2.getX() - p1.getX() ;

**double** yDiff = p2.getY() - p1.getY();

**return** Math.*sqrt*(Math.*pow*(xDiff, 2) + Math.*pow*(yDiff, 2));

}

**Task 5:**

In general, I learned how to write a unit test. From creating, to running the test. A unit test executes a specific functionality in the code to be tested and asserts a certain behavior or state. I also learned that Software unit tests help the developer to verify that the logic of a piece of the program is correct. Running tests automatically helps to identify software regressions introduced by changes in the source code.

The testing framework used for this exercise is JUnit and it uses annotations to identify methods that specify a test. In unit testing you write one test class for each of the classes that you want to test. The test class includes a test method for every method implemented by the class being tested. These tests should be written to make it likely that when all the tests pass, the code essentially is functioning as defined by the specification or as it is supposed to. I also learned the different description for each assertion methods which made it easier to understand JUnit testing.

JUnit support for unit testing is extremely detailed. What I thought was the most interesting was the exception catching. I also like that there is a lot of documentation helping to understand and easy support by Eclipse. Generating certain thing automatically also helps with the development of the unit tests.

**BUGS**

The only bugs was comparing only less than 0 rather than less than or equal 0.