**Task 1 – Preparation Reading.**

Done.

**Task 2 – Getting Started - Fibonacci.**

First I ran the JUnit Test and it failed so I checked the Fibonacci.java code and found out that the first switch statement should return 0 instead of 1 so I fixed that and again ran the JUnit Test and it was successful this time. So I committed the Fibonacci project to my ait624-amirihormozaki-project3 repository.

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

The error was in Point.java code where the this.x = x was this.x = y.

Here is the source code in which I tried to minimize the duplicated code:

/\*\*

\* The Class Rectangle.

\*/

**public** **class** Rectangle {

/\*\* The p2. \*/

**private** Point p1, p2;

**private** **double** xLength;

**private** **double** yLength;

/\*\*

\* Instantiates a new rectangle.

\*

\* **@param** p1 the p1

\* **@param** p2 the p2

\*/

Rectangle(Point p1, Point p2) {

**this**.p1 = p1;

**this**.p2 = p2;

/\*\* Calculates length of x \*/

xLength = (p2.x - p1.x);

/\*\* Calculates length of y \*/

yLength = (p2.y - p1.y);

}

/\*\*

\* Gets the area.

\*

\* **@return** the area

\*/

**public** Double getArea() {

**return** Math.*abs*(yLength \* xLength);

}

/\*\*

\* Gets the diagonal.

\*

\* **@return** the diagonal

\*/

**public** Double getDiagonal() {

**return** Math.*sqrt*(Math.*pow*(xLength, 2) + Math.*pow*(yLength, 2));

}

}

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

I didn’t find any bug for VendingMachineTest.java but I found one for VendingMachineItemTest.java based on which if the item had no name it wouldn’t throw an exception. So I added couple of lines of code to VendingMachineItem.java to correct it:

private final static String NAME\_IS\_EMPTY = "Name can not be empty.";

and

if(name == null) {

throw new VendingMachineException(NAME\_IS\_EMPTY);

}

**Task 5 – Summing it All Up.**

A description (2-3 paragraphs) of what you learned from this project (particularly Task 4)

Well this project was a bit challenging for me specially that I forgot about its due date and didn’t have enough time to spend on it. The first three tasks were easy but the forth one was tricky. Still it was very helpful and gave me a clearer picture of what a test case does and how a test case should be written.

When I started this project I didn’t know what I should test but later on I learned there are more things than just normal functions that can be tested like boundary or exceptions.

A description (2-3 paragraphs) of what you liked and didn’t like about JUnit’s support for unit testing

Unit testing is a very important software development process. I was expecting that frameworks like JUnit make the unit testing automated somehow and I don’t know why. But it doesn’t and I had to write all the tests myself. It gets more important when we need to create test cases for larger projects with much more line of codes and classes.

But still I liked JUnit. It makes the unit testing easier I guess and less complicated and obviously faster. (this is a separate paragraph :D)