**Summing it all up**

For Task2 and Task3, after running both the Fibonacci and the Rectangle programs it was a matter of looking at the code and knowing where the bug was and changing it in code. Then the program ran right. In the Fibonacci program, the issue was that the initial values were not set right. The value for n=0 was set not set right, and it was set to the right value. From the Rectangle program, the issue was that the x-value of the Point class was not transferred properly to Rectangle class. After fixing the issue, the program passed JUnit easily. Running Junit at this point hasn’t been that intensive.

Much of the usage of JUnit was made from Task4. Upon inspection of the codes, it seems that the VendingMachineItem class is structurally simple containing only two methods. It most likely doesn’t contain any bugs. The VendingMachine class seems to be more complex than the VendingMachineItem class. From both classes, when the programs are supplied bad inputs the exceptions will generally flag it. If bugs are found, they would potentially be in the VendingMachine class. The right test would bring these bugs out.

Much of the understanding of JUnit came from YouTube videos as well as junit.org. I believe the knowledge that I gained from these sources was only a scratch form the surface. If I can get more familiarity with it, I can probably make a better judgement on how useful this tool can be. The much broader issue here I believe is how useful unit testing can be, and JUnit is just a tool that we are using. And as far I know, when all the code is combine there will potentially be unexpected outcomes that we might not foresee. Doing unit testing will help minimize these risks.

I may know now why I am not able to push from Eclipse, and this might be because of my computer processor which processes information for 64-bits. I will into this further. In the meantime, I will upload my deliverables as zipfiles.