I learned what JUnit testing is. It is an open-source testing framework for java programmers. The java programmer can create test cases and test their own code or projects they are working on. I have never used JUnit testing before, but it seemed pretty strait forward. Up until now, the way I have tested my code was to constantly run the program over and over trying to meet every scenario. JUnit testing helps out, especially with large programs. I could not imaging testing hundreds of files of code without JUnit testing.

My big take away from this task was the assert command. The command was helpful and easy to learn. It didn’t require a lot of lines code either. The VendingMachine file wasn’t large, but it also wasn’t simple code. Testing with all those methods would have taken a long time. With JUnit cases I was able to test them individually and quicker. All of my testing was laid out and was more precise.

I liked using JUnit testing. This would have made my larger projects go faster. You create a test case file and are able test every method in the code. I like how when you create a test case from the main, you are given every method with their type (string, double, int). I also noticed each test did not require a lot of code. I was able think more about how I wanted to test. This also documents your tests so others can see how you tested or better understand what the code is supposed to do.

There are not many things I did not like about it. I would think JUnit testing would be a waste of time with smaller projects, like the Fibonacci and the Rectangle file. I feel trial and error would be better suited for those. At that level, test cases seem to be more work than the code itself. If the test fails, it can be sometimes difficult to determine if there was something wrong with the test case or the actual code. You need to make sure the test case covers every scenario, but sometimes user inputs can be a better method of testing*.*