

From this assignment I learned more about java syntax especially regarding objects and how classes can interact with one another in the case of the JUnit task. For instance I had my test class extend my class that was being tested so that it could have all the fields and methods that are needed for the test. I am not sure the best solution is having the test class live below the tested class in the hierarchy, especially when you have mountains of code to test. After looking through the liner algebra library I realized that it is best to have your source and test code reside in separate directories and then just have the test access the tested classes through java's inheritance and import functionality.

The other main thing I learned was the design of a hierarchical test suite where you can have a TestSuite class that runs multiple test objects as a package. From reading the test code I noticed that the input to the test and the solution were both hard coded. Which made me research how you do randomized testing. The problem with random testing is you have to find a means of computing the expected value given any random input. This method has to be proven in order for the test to be meaningful. So in order to make a random test case you would have to use a pseudo random number generator with either a static seed or a seed that changes every time you call the test; you would probably do this by reading the system time and taking a few of the least significant bits.