Reflection Paper

When I first started writing the j-unit tests I thought they seemed tedious and even pointless, but then I remembered reading the first page of our book, which was giving an example of the difference between programming for a school class and doing major programing projects for companies. The story talked about how the programs were simple and straight forward enough for one person to do, how once a program is completed it is never looked at again, and each program was only a small percentage of the overall course grade.

If programming in the real world involves programs that are too complex for a single programmer to work on, then easily readable, easily testable code would be very important to be able to maintain and add to. Doing the j-unit tests for this simple program reminds me of when I was able to easily do simple math in my head, so I never wrote out all of the work, but eventually my math classes got tough enough that I had to learn how to write out my work when I already should have known how to do this. It is clear that being in the habit of writing tests for programs is tedious now but will pay off big time in the real world.

Every program I have written up to this point, I have worked hard to complete it, but then I have barely looked at it after words. I know that in the real world programs will be constantly maintained, so writing test codes to make this easier will obviously make by time at work a lot easier and more efficient.

While any program I will write at this point is only worth a small percentage of my overall course grade, it is vitally important that every program written in the real world meet all of the necessary specifications, as if the program were not important it would not be written. Testing as the code is being written will help to make sure that the program is functioning as it needs to be.