John LeClaire

Independent Project Narrative

The idea to program a plagiarism checker was given to me by Mr. Kiang. I didn't want to do it at first because of how difficult it sounded. The difficulty of the project actually made me choose it, because I soon began to ponder how such a difficult task could be done in the first place. This put me into a real world scenario since most computer programmers don't get to decide the easiest project to do. Most are assigned tasks that are necessary to finish for their contractor.

As soon as I started, I was faced with problems that I had never even seen before in Java programming. I needed to quickly figure out how to get the program to accept and read files. Having challenges like these allowed me to have a broader understanding of Java. I now had to interact with Java through an environment other than the command line. Little did I know, all I had to do was pass the file path of the document as a parameter when instantiating a Scanner object. It was developing a more open mindset that helped me with this project.

The next big problem I encountered was finding a method to create an array of every 6-word phrase of a document. It wasn't as simple as I thought it was. I couldn't find a Scanner method that could read the document one word at a time: I could only read every line. I managed to overcome this problem from reading every line in the document, then using the split() method in the String class, which breaks up a string at each specified character. In my case, it was a space. This way, I could easily get an array that had one word per index. From there, it was a breeze writing the rest of the mechanics.

The biggest and final problem I ran into was after I had a working product. My program was working as I had intended. It would compare one document with every other document in a folder. After debugging my program and thinking about the overall functionality, I came to the conclusion that my program by itself was not ergonomic. I had to think from a real world perspective. A teacher would want a plagiarism tool that would analyze the whole class, rather than a specific person. There would be no reason to use my program unless a document was already suspected. I intended my program to be a plagiarism checker, not a confirmer. I concluded that in order to make my program more useful, I would need to make it check every document with every other document. This sounded like a daunting task, since I had already completed the framework. Modifying the framework would've been difficult because it was built to specifically work with one file. There was no simple switch I could flip that would read every file. After spending the majority of my time thinking about this problem, I wondered if I could make a driver for my driver. It sounded frivolous at first, but coding can do anything. I thought *why not*, and soon found out it was a simple process. I created a new driver, turned my original driver into a class while making its entire main method a callable method, instantiated my original driver x amount of times for each document in the folder through the new driver, and used a for loop to call the main method of each "driver" class. I didn't think it would work, but it worked.

The outcome was great. I had a program that implemented many designs that I had never thought of before, such as reading files, hashing strings, and creating drivers from drivers. I'm very proud of it, and I learned how to code all of those just from doing this project. The thing I'm most glad that I learned is how the I/O system and FileReader works.

I can see how useful this will be in more advanced projects, and now I have the knowledge needed to use them.

I collaborated with Nicky Kyono a lot. Our projects both started off by reading .txt files, so we both worked together on figuring out how to return strings from a .txt file. We also talked more about the content that we already had learned, such as how to iterate through array lists. It was a fun experience as we both showed each other how we hurdled over the difficulties in our projects.

Overall, this was a very fun project that really challenged me in some areas. I learned a lot more about Java just from having to implement different packages, and I got to experience a little bit more about what it truly means to be a coder. I think I had a good process with this project. I met challenges, and successfully worked my way around them. Not only was I aiming for a working final product, I also had to add a bit more than I planned to make my product more user friendly. At the end of the day, I had a project that I really worked hard at and actually succeeded my expectations.