Christopher Morcom (A20385764)

Jon Hanrath

CS115 Section 2

05 December 2016

**CS 115 Final Project Analysis**

**Summary**

In summary. I think this project was a little challenging, but was possible to complete with all extra credit if worked on with excellent time-management and efficiency. This project took almost 20 hours to complete for me (in which about 90% of the extra credit was completed). To me, this project seemed much harder than the standard labs we took did class, but it gave very good insight into how a real solo programming project could be: tedious, frustrating, and very time-consuming. However, I did complete the project in the end, only making errors where I could not do what seemed to be basic operations to a person – and not a computer.

**Problems**

Over the course of the project I became stumped due to a multitude of errors. Some of which include: not being able to create a simple alphabetical sort using the String.compareTo() method, having to create a temp .txt file which removed comments from the original cipcs115.txt file and then reading from that (which was not specified in the project guidelines as I did use the cipcs115.txt file), and trying to enable the use of ‘all’ as a valid entry in the voteInfo() and stateInfo() methods. Additionally, due to Professor Hanrath’s email to me, I took the challenge to not use the “continue;” or “break;” statements anywhere in my program (unless it was to **completely confirm** that there would be no errors in the major project-required methods.

To get around most of these issues, I just found another way of doing what I wanted. For instance, in the sortList() method, I just simply ‘gave up’ due to time constraints. For the commenting extra credit, I wrote a new file and scanned an ArrayList from that. Finally, for the voteInfo() and stateInfo() methods, I solved the stateInfo() method first and copied and pasted my code to the voteInfo() method, tweaking it a little bit so that the outputs for each method was correct.

A completely separate JAVA problem I seem to have encountered was that whenever I passed a Boolean value through a ‘for’ loop without reassigning it, the Boolean variable always seemed to be rewritten to a ‘true’ state. To get around this, I used integer values assigned to ‘1’ or ‘0’ values.

**Concepts Learned**

As I completed the project and succeeded my errors, I learned a few things. The first is: “Insurance is good.” Early in the project, I coded about 400+ lines of code which all got deleted due to a Windows Update. Since then, I have created multiple backups of my program files and saved them consistently. Also, I applied this idea to some of the methods in my project by using multiple “insurance variables” to make sure loops aren’t broken early and errors are carried through. In one specific case, I had to use “throws IOException” in my finalStats() method to account for the user putting two ‘q’ commands consecutively. This was because I was not allowed to use try-catch cases throughout my program. Also, I found out that the scanner class has methods that do not consume newline characters. For instance, when I tried to use scan.nextInt() and scan.nextLine() in succession, scan.nextLine() would return what appeared to be nothing, whereas it in fact returned the ‘\n’ or newline character.

**Would Do Differently**

In conclusion, I do not think I would have done very much differently if I could do this project again. This is because of my current programming level and my understanding of computer logic. If anything, I would use different libraries like JTextPane and hashCode to make it easier to program. I could also have shortened my program by writing and calling to more methods and passing information along methods more efficiently. However, in my belief, I have done the best I could with my current knowledge and creativity levels regarding programming in JAVA.