Course: CSC 340.05 Toe

Student: Steven McHenry, SFSU ID: 916931878

Teammate: None

Assignment Number: 02

Assignment Due Date & Time: 10-04-2019 at 11:55 PM

**Program Analysis to Program Design**

*Your analysis of the provided information and the provided sample output. Compare to the*

*ASMT 01 Java version.*

The provided information and sample output were quite similar to that of assignment 1, except of course the added text file and the associated directions in regard to it. There is also some slight variance within the given output and the added clarification that we cannot search the text document for out words but must search our data structure.

*What problem you are solving. How it is different from that of ASMT 01.*

The problem we are solving is most different due to the extraction of information from the text file. We must separate the information in a specific way instead of being able to start with the information separated due to our own enum definitions. We are also solving the problem of deciphering the user’s inputs and then being able to handle all possible cases without allowing the user to break the program.

*How you load data from the data source. What the steps are. Why these steps.*

I am loading the data from the data source via the fstream package that can be included within a C++ program. I begin by created an fstream, string vector, and string variable in order to parse all of the data that is incoming and store the individual components. I then open the file and stream all the data until it reaches the end of the file. I store each line of the file using getline() within a token variable and then insert those tokens into my string vector. After the file reaches its end, I close the file and then send my string vector off to be added into my map.

*Which data structure(s) you use/create for your dictionary. And why.*

I chose to use a map with the key being a string and the value being a vector of strings. I chose this data structure because it would be relatively light while also allowing for the flexibility to add any amount of definitions necessary for each word.

**Program Implementation**

*Does your program work properly?*

As far as I can tell, my program works correctly based on the given criteria. Attempting to put in all input within the example I am returned with the same output. There is no hard coding as far as I can tell and this program could be easily expanded upon if necessary. There is also no real way to break the program if you give it too many inputs or completely random characters.

*How will you improve your program?*

I would improve my program by making it more object oriented with classes. Currently it is extremely procedural which could pose as a problem down the road of implementation. It could also surely be improved as far as memory resources go. I am still getting a handle on pointers and there are probably values being copied throughout that don’t necessarily have to be.

**Example Output**

**A close up of a newspaper

Description automatically generatedA close up of a newspaper

Description automatically generated**

**A close up of a newspaper

Description automatically generatedA screenshot of a cell phone

Description automatically generated**