John R McCravy III Chatbot Report

A summary of Design

Program 1 is very simple as it uses a file downloaded from the representative's website to get data from. The program searches through the file for the desired information (i.e. Contact Information and Personal Information) and with that it creates a file that is cleaner and only has the relevant information we want. **Program 2** builds on Program 1 by taking the information in the file output by Program 1 via the JSON reader class and it builds a representative object via the representative class. The Representative object with the name, region, home address, Columbia address, home phone, and Columbia phone number. The program uses a writer class I created for storing that representative's information. **Program 3** makes the chatbots information accessible via the command line. When an exact phrase is entered that matches the information desired, that information is then printed to the command line. Here I followed the directions for accessing the information from the slides (ex. "Contact Information" will yield all the contact information on the representative). **Program 4** uses an array of known queries to match the user input to. By splitting the user input into words, then examining each word of the user query to the possible queries we can determine the percent of words that match from the user query to each possible query, utilizing only the one with the highest percent match so long as it matches by at least 70%. **Program 5** is intended to log the sessions of each conversation. The SessionLogger class is intended to be used in place of "System.out.println" as its purpose is to first log the information and then print it out to the screen. The LogFileWriter does as its named and creates files with the chat log. The CSVFileWriter is used to create and append the CSV file which is used for statistical purposes regarding the chat bot.

Design Reflection

During the creation of the chatbot there were several oversights and realizations that lead to the result being something which I was not proud of. To put it simply, my project lacks a sensible design due to many mistakes which I have made in design and therefore the overall quality of the project. The purpose of the class was to learn and master advanced programming techniques such as memory management, pointers, iterators, polymorphism, inheritance, operator overloading, template functions and classes as per the syllabus and course title. I found these programming concepts and those of object-oriented languages to be both useful and understandable. However, when it came to my project, I believe what happened was that I attempted to apply certain principles either too broadly, or too often. For example, an acronym often taught is "DRY" meaning "Don't Repeat Yourself" and while this is absolutely an important concept, I believe I took it to an extreme. I found for example in the classes for reading and writing that I created an abstract base class which was implemented by the log file and the CSV file and while the concept was there, it would likely have been simpler and better design to just use the BufferedReader and BufferedWriter classes on their own as provided via the Java Standard. My code quickly became nonsensical and difficult to keep track of what was happening. I sat down and attempted to create UML diagrams and flow chats of how my program ran and it was clearly lacking in overall design however it was difficult to decide how to untangle the mess I had created. I believe another related issue that I faced was getting caught up in specific little details rather than focusing on the larger picture. I found myself rewriting my helper methods and reading the Java documentation for what the best practice would be to do this or that. While it felt good to have "fit in" a design principle or pattern (I also used a few singletons for my SessionLogger class) I soon realized when construction of the

final program came, the programs did not fit well together. Another issue I see is that I created each program as a separate thing to be passed along to the next one, and while that felt natural as we did each at a different time it came to me that may not have been the best plan. While I am not pleased with the outcome and performance of my project, I am pleased that I do understand the principles the class was intended to teach and am further pleased that I now have a direction to work towards. The goal in which I walk away from this class with is to learn how to better plan the implementation of these techniques, as I understand how to implement them individually and on their own but when it comes to which decisions will fit best into the overall design of another program or project, that is what I need to learn how to better discern.