Drew Lukens

Dr. Turner

Csc 121

October 14th 2019

Final Project Write up

**Overall Structure**

When you first start the program you will be greeted by a main menu screen. Here there will be several options. You select an option by typing the number of the desired option. You will be able to Create multiple accounts, deposit withdraw and transfer funds. On top of this you are able to total the amount contained in the bank. Each option has a custom list prompt to tell the user what type of information they need to type.

**Design and Testing**

When designing this I first started with a large output Mainscreen function. I then began to add on features. I decided to create a new account class to tie all the metadata for the account together. I had trouble designing a good findAccount function that used the string that was inputted to locate the account. In order to create this program I first wrote it in a text editor called “Atom”. I then copied the code into Visual Studios 2017. This allowed me to easily Compile and test my code.

**How this design satisfies the course requirements.**

-I used several variables throughout the project.

-In order to store the accounts I used a vector of accounts.

-An important design choice was to create the custom “account” class.

-I used several output streams to present information to the user while also using input streams to gather choices from the user.

-I used conditionals several places in the project to determine what the next step should be. Most notable when deciding which part of the project the user wanted to run.

-I used a couple while loops when using iterators to run through the given function. These were useful whenever I needed to check the vector for a specific account.

-Originally I had a different function for deposit transfer and withdraw, but towards the end I removed them and hardcoded the info into the main function. I kept the findAccount function and the overloaded function.

-I used a couple sloppy object creations notably in the first option when the user wants to open a new account.

**Known errors and things that need to be fixed.**

There were a lot of errors that arised when the user inputs information.

-If you type in a value that doesn’t match the initial data type of int the program will run the mainscreen() function repeatively.

-There is no way to ensure that an account can be searched and found. You need to type the exact name of the account when you try to manipulate it. Find function is faulty

-Lots of sloppy account object creations.

- It would always crash for me on the first time mid selection. But when it would re open it would work correctly. Probably reaches the max memory allocation and crashes?

**What did I learn and find difficult?**

I learn how difficult creating custom classes can be and the challenges that arise with them when you need to print or find them in a vector. That was the biggest challenge that I faced. I struggled to find an easy way to search the vector when the user inputs a string name. I probably made it a lot more complex than it needed to be. I also learned how useful iterators can be to cycle through a container.

**Potential Further Work-**

-Upgrade the finding account in the vector list.

-Rewrite the deposit withdraw and transfer functions.

I had to move those to the option selection due to time constraints.

**Conclusion**

This was a simple solution to a potentially complex software problem. It is far from being able to be implemented by any major banking company. Although with nearly 200 lines of code it satisfies all the requirements that I was trying to complete.