## Ericka Najera

## 8 November 2019

## Dr. Mejia

## Programming Assignment #3

I confirm that the work of this assignment is completely my own. By turning in this assignment, I declare that I did not receive unauthorized assistance. Moreover, all deliverables including, but not limited to the source code, lab report and output files were written and produced by me alone.

# **Program Explanation**

The overview of this assignment is to augment the code from the previous lab assignment in order to be able to add a new user, print bank statements for a user and read a transactions log and do each one accordingly. First, I did the new user since it was the easiest to implement. Then it was the transactions log and finally the bank statements. I did both the transactions and the bank statements at the same time because if I was coding one, I would get an idea for the other one and it I would pause and continue to the next one.

# **What did I learn?**

From this assignment I learned that you should refactor your code to work for any additional things which prevents you from basically making two methods that have the same functionality. At some point I had to add a few parameters to the methods I already had and made them more universal for the whole program. Luckily my log class documented everything a transaction happened, so when I read the transactions actions file, it documented everything by is self. I also learned how to create my own csv file. I imagined my bank statement file to be like a text file but making it a csv file makes it look more organized and neater. I feel like my solution for checking for input and asking the user again if the input is incorrect is a bit repetitive, but I honestly don’t know a way to fix it. Since it was one if the busiest weeks of the whole semester because I had work, arch lab, and two midterms. I had to prioritize other things and I started the lab on Thursday, and it took me about 8 hours to complete it.

# **Solution Design**

In this program I changed some of the data structures on little things which were like making an array list for the balances. On the new user I didn’t know how to design it other than ask the user for input and create accounts with balances of zero. Since the user can deposit into their account after they are created an account. For the transaction actions I read the file and based to the key word I would proceed with the transactions. After you run the file, it prints all possible all the transactions on the terminal as well as in the log file.

# **Testing**

I tested out my program by using it in the way of a user. I tested different scenarios while testing different one I found some errors that I managed to fix. I am pretty sure I didn’t test all possible exceptions but one of the big ones were find the file. Somehow reading the file gave me errors which I didn’t know why but later on it worked. When I tested my partner’s, Karla, code, I tried to be mean and act like a dumb user that input ints instead of strings or things like that. On some cases she didn’t pass the test and so I assume she was able to fix it. But other than that, she did a good job. Me testing her code is white box because I couldn’t see her code and I did black box testing because I could follow my code and see where the error could be.

# **Test results**

My test results turned out good, also there is room for improvement but practice makes perfect.





