**Oscar So**

**CS 2024 – C++ Programming**

**Assignment #4: BankAccount**

Given that this assignment was a riff/continuation of assignment 3, I felt very prepared in further practicing the C++ concepts that I have learnt in class. This time, it was cool to use more than one header file and now having to have both menu and bank account files to work with. To be honest, I still have some trouble trying to understand the difference between the header declaration file and the actual definition of the functions in the corresponding .cpp files using “BankAccount::” as the tag. Overall, though, I still managed to finish this assignment rather quickly as my post on piazza was answered quickly by classmates and it was a great help.

For main.cpp, it was really similar to last assignment’s where I had to switch between cases of the user input to determine which function to run. Furthermore, given my previous knowledge in programming, the global functions that we were asked to write about, for example: the getBalance, deposit, and withdrawal functions, were all pretty standard and I instantly knew that I had to pass the BankAccount object within the parameters to make the program work.

One cool thing I did was that when doing withdrawals, I realized that the function written inside the BankAccount class was just a return of a Boolean versus the function of deposit where it actually changed the mBalance of the bank account. Knowing that deposit is just the opposite of withdrawals, when the withdrawal function is called inside the main.cpp file, I first check whether there is sufficient funds and if the value is positive. If so, I then called the deposit function on the withdrawal amount multiplied by (-1). This way, by depositing “negative money”, given there is no specifications to if the input for deposit is negative or not, I am technically withdrawing the amount and subtracting the value from the bank account this way as well.

Lastly, I learnt a lot about pass by reference versus pass by value in this assignment. Being a Java-heavy programmer, I always thought that my functions would always behave the same when I input the same parameters. However, for the global getBalance, I realized that when I did getBalance(BankAccount bank), the balance of the bank account was never updating and this was because I was actually passing this bank as a value/copy instead of passing by reference. This code would, I believe, be perfectly normal in Java or Python, but in C++, I learnt that I have to use the (&) symbol in order to pass by reference and refer to the actual bank that I am using by doing this: getBalance(BankAccount &Bank). I also learnt to use the this-> simplification method instead of (BankAccount::) when it came to accessing private variables inside the class.