**Learnings from Assignment One**

This assignment posed two unique challenges: relearning coding conventions after taking a year away from computer science, and learning a brand new language. While the ability to write and organize code came back naturally, learning C++ proved to be a daunting task.

Having never coded in the language and using only Java for object oriented programming, coping with the differences in memory allocation, initialization, and pointers was both an enlightening and frustrating experience.

My key take aways from the assignment are as follows:

1. Never assume two languages are similar.
   1. While some features are similar, many conventions between languages can differ drastically
2. Pointers are very useful
   1. While confusing, the ability to manipulate memory directly and refer to a single instance of an object is a powerful thing that can create a lot of flexibility and efficiency in your design if properly used
3. Having a clear design helps immensely
   1. Trying to simply visualize the program and how it should work was not an option when creating the program from scratch. Traditionally most programs are handed to students half finished, so starting with an empty project and trying to create a full system was difficult without being able to refer and compare to something. Having a basic UML of the system proved very helpful when deciding what to code and how
4. Setting an object to “=” something doesn’t necessarily make it the same
   1. Contrary to Java, setting an object = to something just copies the object into another instance, rather than having them as the same object with two different references. This lead to a lot of difficulty in persistence and properly modifying data

Overall I learnt that C++, while powerful and having lots of functions, is not a very user friendly program. It is very technical, but with that comes a lot of different ways to approach a problem and solve problems in efficient and simple ways.

Notes on the Program

As it will be seen, persistence across user sessions (logging out) and system sessions (closing program) is not working 100%. This was due to using a for- each loop in the get methods for users and accounts when retrieving them from the master lists (account\_list etc). I realized too late what the source of the issue was, and was unfortunately unable to change those functions to return pointers instead. This would have required rewriting much of my code to handle pointers and to properly allocate memory throughout the program. When push came to shove, I decided having other features work properly was better.