For this refactoring assignment, I have decided to stick with my event simulator project! After looking through my code more thoroughly there are definitely spots where refactoring would make sense.

Duplicated code:

I have a few functions, for both the grocery store as well as the bank. There is nothing different about the methods besides the names. My thought, to make a parent class that the bank and grocery classes will inherit from. I wasn't aware c++ had something similar to java in that you can basically do the equivalent of what extending an abstract class does but in c++. After seeing an example how to use inheritance I think it will be a good exercise to use that as one of my refactoring techniques.

The motivation:

My motivation behind this is to become a better developer and uphold best practices. As I will be entering the real software engineering world soon I would like to be a programmer that others want to work with based on not only my attitude but skill. That starts with making sure I do my best to follow best practices; making code easily readable and reusable. Inheritance lends itself well to those three principles. It will be a good practice to use c++ in a more object oriented manner also. I have mainly used c++ in a static aspect and I would like to get away from that.

Making sure it works:

To ensure my program still works, I will refactor in small segments. Do one section, run some tests to make sure the results are still the same, then refactoring another section until all code that is scheduled for recatoring is completed. Doing this incrementally will ensure data integrity. If there is time, I will also add some unit specific tests. This would have helped speed up the refactoring process as I could the necessary changes and then run tests that were already built had I done this step earlier on.