Antonio Scalfaro

CMSC 330

Professor Yuhua Jiang

Project 2

Due: 05/06/2024

**Approach:**

The approach I took to this project was similar to project 1. It took me some time to gather my plan of how I wanted to accomplish the goals of the project but once I got started, I followed a similar pattern to implement all of the new components. After each component was added, the tests were run to ensure the component worked correctly. This was difficult because I have not used C++ before but once I was comfortable with the syntax, it was smooth sailing. I built each class and was able to make each class contain a single in line function and thus reduced the need for additional cpp files. The hardest addition to the project was the error catching. I accomplished it but the error catching works not quite to what I would prefer. For the duplicate instantiation of a variable, it will catch it and not include it in the testing output. For variables not instantiated, it will catch it and throw an error message to the output. The latter is the preferred behavior but overall I am still pleased with the performance of the program.

**Test Plan:**

The testing of this project is all done through one file, the input.txt file. It has all the test cases that were provided by the project folder. The program passes all of these tests with the correct output being output to the consol. The only undesirable test is the duplicate variable instantiation, which does not throw an error but does catch the test and allows the program to continue to run.

**Test Output:**

A screenshot of a computer

Description automatically generated

**Lessons Learned:**

This was a lot of fun and a very challenging project to tackle. As I stated previously, I have never used C++ and so a lot of my time spent on this project was to familiarize myself with the syntax of C++. I really enjoyed making each class and got a bit creative to avoid adding additional cpp files and keep the new classes to a single in line function. I need to explore more as to why one of my exception handling classes works appropriately (throws exception to console) and the other only catches and skips over that line of input. It was also a challenge with compiling and running this program while implementing new classes and behavior. For some reason, the program would produce errors in compilation that forced me to create a brand new directory and copy the files into the new directory and recompile it there. It would work error free after moving the files to a new directory. Other than that, the logic itself was fairly simple and the code runs as desired and I am happy with my work and the results.