Zack Nicholson, Leo Pangco, Thomas Widjaja, Alex Gomez Chavez

Professor Mak

CMPE 152—Section 1

12 September 2018

Lab 2 Writeup

Some of the assumptions made for the assignment came from the guidelines at hand. The assignment only required us to recreate a C++ frontend, which would mean some of the pascal-types that were accounted for in the vanilla assignment files needed to be switched to C++ types and logic. For example, one of the first edits done to the project was to completely rename all pascal types as C++ types, ranging from files, to variable names. Another example of assumptions made by the team came from the fact that the frontend file types were the only data paths that needed editing. At first glance, it may seem like a team must completely edit the entire project. Deeper into the assignment, where we were implementing logic, we also assumed that it would be necessary to create a new class to identify characters in addition to strings. The reason for this logic is because of all of the possible literal cases of C++ that were not accounted for in the vanila assignment files. One of our final assumptions for our assignment was that even though our frontend successfully parses and scans the input file given, there are still many other C++ data types and symbols that may not be accounted for that can possibly create unorthodox edge cases. However, what the input file provides results in a fairly robust program that can read very complex literal terms and handles such tokens with grace.

We used a few clever tricks to lessen our workload. In the character class we just checked for the “\” to determine escape sequences since all the sequences given were valid. More error checking will have to be added later. This was just a shortcut. We also used peek logic to check next character to ensure multi character sequences where valid before the current characters were destroyed. Specific ordering of functions was required when consuming characters, adding the proper text to the token, and adding the proper value to get the proper output without certain characters being eaten or added unnecessarily.