# CPSC 254 Lab 6 Group 2-3 Test Results

**Design Style # 1** Error checking for opening and reading in a text file**:**

**Design Style # 2** Improved testing, such as using one vector in main instead of making a new one each function. Also improved error responses, to identify which part of the procedure function is not working

**Design Style # 3** Improve/Optimize/Make errors or error messages more specific for test functions:

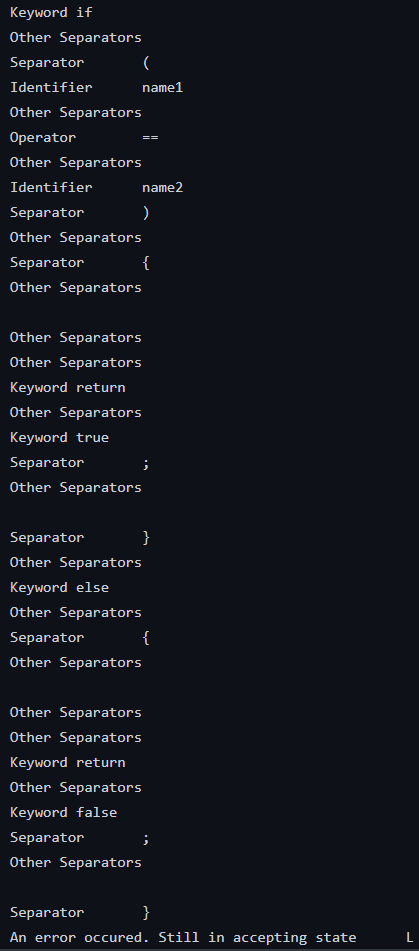
Implementing more specific error messages to describe which function was displaying an error. Messages now displaying the input that was received and if the output in the result was correct which each function. If the desired output does not match the output that was presented, then the function provides an error statement describing what rule did not work.

**Design Style # 4** Improve token class to optimize testing. I.E. adding more constructors

**Design Style # 5** Simplify redundant or repetitive tasks that use multiple lines of code

**Coding Style # 1** Output file for testing**:**

During the testing of this function, I tried it on a handful of test cases and it works quite well at outputting the token and lexme variables. I could probably modify ofstream a little bit more to change the way the file is formatted and written too.

****

**Coding Style # 2** Struct to hold file and token values

**Coding Style # 3** (Google) C++ styling (matching bracket spacing style, same comment style)

**Coding Style # 4** More descriptive comments

**Coding Style # 5** Renaming of necessary variables

**Results for Test Cases:** within the main function, the “temp.txt” file containing the code to be used as tokens, is checked whether it can be opened and if so then checked if it is not empty. The vector of tokens are then generated. When making a function call for a test case, the vector of tokens is passed. The results are printed, the contents of the vector are shown, and the “outputfile.txt” is created with a copy of the results as well.

