Project 1 Summary Report

Jade Pearl

CMSC 430: Compiler Theory and Design

Professor Gregory Williams

Due Date: March 26, 2024

Project Approach

My approach to this project was to first go through the course learning resources for the week one more time so that I would ensure that I was not missing anything important. After downloading all the required materials for the project, I opened all the codes in VSCode (which is the editor I use) and I copied all the given test case files into the skeleton code folder to set everything up. The Project 1 Approach was extremely helpful as an outline to walk me through the tasks I needed to do. I worked through the instructions while reading the approach document to make sure I was not doing anything out of order that would potentially mess up the compiler and testing. I ran into a few issues along the way, especially with regular expressions. When I had issues, had classmates look at what I had and we were able to work out my issues from there by telling me where I needed to change my scanner.l file or the listing.cc code. Most of my errors simply came from my lack of experience with regular expressions but once I got feedback from classmates as to why my regular expressions were not working, I was able to reason with myself through correcting the incorrect expressions.

Test Cases

For this project, I used the 8 provided test cases while progressing through the assignment to make sure the compiler was working as intended. In addition, I created three more test cases which followed the criteria for the assignment: one test case with no errors, another with multiple errors on one line, and a third containing all lexemes from the flex file.

|  |  |  |  |
| --- | --- | --- | --- |
| Input | Expected Results | Actual Results | Pass/Fail |
| example1.txt | Compiles successfully | Compiles successfully | Pass |
| example2.txt | Returns that there are 2 lexical errors | Returns 2 lexical errors: # and $ | Pass |
| example3.txt | Contains all lexemes and compile successfully | Compiles successfully with all lexemes | Pass |

Here are the screenshots for each test case:

A screenshot of a computer program

Description automatically generatedA screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

Lessons Learned and/or Possible Improvements

This project has enhanced my understanding of lexical analysis and regular expressions. Expanding on a partially implemented lexical analyzer for a scanner taught me how to parse source code and identify tokens based on predefined rules. I also improved on my understanding of regular expressions because I was able to reason with myself through my mistakes so that they could be corrected effectively. Long regular expressions were harder for me to understand so I had struggled writing the implementations out that were requested in the project requirements. Looking ahead, I hope to improve my understanding of lexical analysis and even more so of regular expressions so that those parts of future projects are not as difficult for me to establish. I also hope to expand on the syntactical and semantic analyses so that I am better acquainted with other compiler design concepts.