Project #1
CS 3510 – Spring 2014
Seth Yost, Mitch Birti

I. Requirements:
Develop a scanner which can tokenize a program written in the C- language described in Appendix A of the text.

II. Design:
We built a state machine off of the description of the C- language so we could parse a character stream into various tokens.

III. Implementation:
A CharReader provides characters to the CminusScanner one-at-a-time. The scanner feeds them into a state machine (constructed using a switch statement on a State enum). The state machine interprets them and constructs Token objects with a TokenType enum and any implied token data (e.g. for NUM, the actual number).

IV. Testing:
We wrote a Tester class that outputs the tokens generated by the scanner when it is given a test file. The test file contains a diverse style of whitespace so that the most edge cases could be covered, in addition to many of the types of tokens that must be interpreted.

V. Summary/Conclusion:
Our project is working properly. We didn't encounter any major problems with our Scanner implementation, which was extremely surprising, but we both wrote half of the state machine and then read the other programmer's half; this eliminated all but one bug before we ran the program. The one bug in the scanner involved detecting the end of the stream. We had been checking for the wrong character, so it was a quick fix. The only other problem we had was with our Tester class. We forgot to include a statement to fetch the next token, so we ended up just printing out the same token over and over. Again, it was an easy fix.