21 Aug 2016

Flex – Clite

Compilers 2016 Assignment 1 | **Deadline: Saturday, 27-08-2016, 11:59 p.m.**

For the first assignment (as well as phase 1 of your project), you have to build a lexical analyzer that converts a text string into a series of tokens. There are lots of tools for handling this. For this assignment, you'll make use of the **flex**, an open source lexical analysis tool, and build a parser for **C-lite**.

The way flex works is that you first create a file that defines symbols and production rules that describe how to parse a text file. Then you run flex to generate a C file. This C file can then be compiled and run on the required input file, in this case, the Clite program.

Important links:

- Documentation for flex.
- C-lite Grammar (For future reference)

The objective of this assignment is to implement a parser for a subset of the C-lite grammar. Your program should successfully parse all valid programs, and throw an error if it encounters an invalid program. Ideally, your program should also ignore comments.
Output to stdout:

"Success" -> Successful parse
"Syntax error" -> Invalid program

Create a file flex_output.txt which on encountering a token, prints 2 lines. 1 with the token type and the 2nd line with the token value as per the table given above. Only the following tokens need to be processed.

Output	**Token Found**
Integer **<value>**	A sequence of one or more digits.
Float **<value>**	An integer, a dot, and another integer.
Keyword **<word>**	if | else | while | for | int | float
Identifier **<name>**	Legal names/identifiers*.
Assignment **=**	=
Comparison **<symbol>**	== | < | > | <= | >=
Operator **<symbol>**	+ | - | * | /
Open-bracket **{**	{
Close-bracket **}**	}
Open-paren **(**	(
Close-paren **)**	)
Semicolon **;**	;

*An identifier is a sequence of letters, digits and underscore (_), starting with a letter, that is not one of the keywords.

Please stick to the output format as there will be no manual evaluation of codes submitted.

Submission Format:

Compress a) the flex code (named Assignment1.l), b) a sample input test case (named test_input), and c) a readme file and upload the zip file. The output file generated must be as specified above.

The zip file should be named rollno_Assignment1.zip.