Assignment 2: Write a Lexical Analyzer for a simple C language using JFlex

Write a lexical analyzer for a simple C language. Your lexer must be implemented using the JFLex tokenizer generator to produce a Scanner class. An initial version is available and you need to extend this project to meet some requirements described later.

Start Project:

You are provided a start project which is a really simple C lexer. The start lexer can recognize following lexemes:

- White space characters (include spaces, tabs and new line characters)
- Single identifier (have length of one letter)
- Integer literal (number value of integer type)
- Some basic keywords and punctuators
- The EOF (end of input) character

You can run the start project by following steps (make sure that you have java and javac comands configured that can run in your command line environment)

```
java -cp JFlex.jar JFlex.Main simpleC.flex

javac -cp ;jflex.jar MyLexer.java

java -cp ;jflex.jar MyLexer [input file]
```

A batch file is also provided to aid your test. You can compile and run by simply type:

```
run.bat [input file]
```

Otherwise, you are recommended to develop your project with Eclipse IDE and its CUP/LEX plugin. You may have a look at [2] for installation guide.

If the input content is lexial correct, a token stream will be printed via standard console. Otherwise, if there're tokens can't be recognized, the program will throw error information.

End Project

The end project is only provided with a released program which may helps you while you are doing your homework. You can have a look at its output for comparison.

You can run the released program by simply type:

```
java –jar MyLexer [input] [output]
```

If output parameter is not provided, the program will print result via standard console. Otherwise, result will be printed in that output file.

Project Details:

Your job is to complete the start project so that your lexer could recognize and produce an appropriate output for our two test files: test1.c and test2.c.

In another way, your lexer, at least, must have below features:

- Recognize identifiers that have multiple characters. At the moment, your lexer can only recognize one-character identifier.
- Recognize more keywords and punctuators, at least, those presented in test1.c and test2.c.
- Recognize floating point number literals (including literals with scientific notation)
- Recognize string literal.
- Recognize comments (both single-line and multi-line comments). You should able to get comment content of both types. Don't pay attention to nested comments.
- Write output to a specified file (instead of standard console output)

In case of raising errors, at the moment, your lexer can only produce "illegal character" error. Your must implement your program to handle warning or error in following situations:

- Warning: When recognizing a really big integer literal (more than 4-byte signed integer)
- Error: Unterminated string literal: A literal start with a quote (") but don't have a corresponding ending quote.
- Error: Missing end comment mark: There is no ending comment (*/) for a corresponding start comment (/*). (Don't pay attention to reverse situation.)

Note that you are strongly recommended to start with the provided project though it's not mandatory. You can develop your own project as long as it meets our submission requirements.

At first you may have a look at JFlex user's manual [1] to know about its specification and how to modify it.

Project Submission

Compress your project into a single file named <StudentID_Assignment2>.zip and submit to our course website. Make sure that both source code and a runable version of your program, named MyLexer.jar, are also included. Then we must able to test your program by just type:

```
java –jar MyLexer.jar [input] [output]
```

Reference

[1] JFlex User's Manual, http://jflex.de/manual.html

[2] CLE Documentation, http://cup-lex-eclipse.sourceforge.net/installation.html