CSE 5343, Project 1: simpleC Scanner

You need to build a scanner for a subset of C, using JFlex. An initial version is available on the course web page under “Projects”. You need to extend this scanner to add all functionality described in TODO comments in simpleC.flex. Test your implementation by performing lexical analysis on two C programs (fft1.c and fft2.c), also available on the course web page.

You need to submit new versions of simpleC.flex and simpleC.cup. The project is due by September 3 (Monday), 11:59 pm.

Some Details

- Do your work on stdlinux. Before starting, check “Resources” on the course web page for one-time setup instructions.
- Do not change MyLexer.java (the driver program) or MySymbol.java (a helper class, extension of Symbol). The output from MyLexer will be used for grading.
- The input will always be ASCII – you do not need to worry about Unicode characters (that is, you can ignore universal-character-name defined in Section 6.4.3 of the ANSI C document).
- Read simpleC.flex very carefully: there are many important details in this file.
- You do not need to handle ‘long long’ integer literals, ‘long double’ floating point literals, and the following eight punctuators: # ## <= := < > % % : % : . You can assume that these will never appear in the input.
- You can assume that each integer literal and floating point literal has a value small enough to fit in the corresponding Java type. See simpleC.flex for more details.
- One simplification to make life easier: consider the following input 32 456Last chance. The implementation should generate the following sequence of tokens: INTEGER_LITERAL with int value 32, INTEGER_LITERAL with long value 456, IDENTIFIER with value 'ast', IDENTIFIER with value 'chance', EOF. Of course, this is not what a real-world C language implementation would do, but for this assignment it makes the scanner a little simpler.

Project Submission

On or before 11:59 pm on the due date, you should submit two files: simpleC.flex and simpleC.cup. Log in to carmen.osu.edu to upload the project. You can submit up to 24 hours after the deadline; if you do so, your project score will be reduced by 10%. If you submit more than 24 hours after the deadline, the submission will not be accepted and you will receive zero points for this project.

Academic Integrity

The project you submit must be entirely your own work. Minor consultations with others are OK, but they should be at a very high level, without any specific details. The work on the project should be entirely your own: all the design, programming, testing, and debugging should be done only by you, independently and from scratch. Sharing your code with others is not acceptable. Submissions that show excessive similarities will be taken as evidence of cheating and dealt with accordingly; this includes any similarities with projects submitted in previous instances of this course. Academic misconduct is an extremely serious offense with severe consequences. Additional details on academic integrity are available from the Committee on Academic Misconduct (see http://oaa.osu.edu/coamresources.html). I strongly recommend that you check this URL. If you have any questions about university policies or what constitutes academic misconduct in this course, please contact me immediately.