CSE341 – Programming Languages (Fall 2020)

Homework #3

Handed out: December 28, 2020.

Due: 11:55pm January 13, 2020.

Hand-in Policy: Source code should be handed in via Moodle. No late submission will be accepted.

Collaboration Policy: No collaboration is permitted. Any cheating (copying someone else's work in any form) will result in a grade of -100 for the first offense and -200 for the subsequent ones.

Grading: Each project will be graded on the scale 100.

G++ Language Syntax Analyzer: Given the description of the G++ language (G++Syntax.pdf) you are asked to implement a syntax analyzer that accepts or rejects a given program in G++. Syntax analysis rules of G++ is explained in the document in G++Syntax.pdf.

You are asked to implement the syntax analysis in two different ways:

1. There are tools to implement syntax analysis given the rules in a meta-grammar such as CFGs. One such tool is "Yacc" that lets you generate C code to do the syntax analysis.
2. For this course, you will be implementing a syntax analysis algorithm for G++ in Lisp as well. For this you are not expected to use a meta-grammar to define the lexical syntax of your language.

Both analyses tool should start the interpreter. It will read one line at a time from the user and check if the input syntax is correct while generating the parse tree for later processing. You are also expected to implement code for expressions as well as if statements.

G++ Language Interpreter using Flex and Yacc (35 points): Implement your interpreter using Yacc. For this you will also need your lexer from the previous homework. Hand in your "gpp_interpreter.y" file for this part of the homework. You are also expected to submit the corresponding C file generated by your lexer and syntax analyzer along with any other .h or .c file that is needed to compile "gpp_interpreter.c" using GCC on Ubuntu (16 or later).

Full score would require the lexer code to implement the CFG for expressions and if statements. This also means that you need to have the proper things for some basic data types. 20 points will be taken away for those not implementing a proper CFG.

G++ Language Interpreter in Lisp (65 points): Implement your interpreter in Common Lisp. Hand in your "gpp_interpreter.lisp" file for this part of the homework. This file should have a function called "gppinterpreter" that will start your interpreter. "gppinterpreter" can have zero or one input. The input can be a file name which will be loaded by the interpreter and interpreted right away.

There will be 20 point reduction if proper CFG parsing algorithm is not implemented.

Examples: The following table provides the correct tokenization for a few instances of G++ language. Note that lexical syntax error messages should give some information about the error encountered.

;; helloworld.g++ (+ 10 10)	Syntax OK. Result: 20
(list 1 2 123)	Syntax OK. Result: (1 2 123)
(++ 1 2 3)	SYNTAX_ERROR Expression not recognized

Hand in your code in a single tar for zip file named yourstudentnumber_lastname_firstname_hw3.zip. For example, if your name us John Wayne, student number is 123456789 and zipped everything in a RAR file, then you should submit your file as “123456789_Wayne_John_hw3.rar”.