

CSE351 FALL 2017 PRELAB2

Write a grammar error checker using lex for the given input C++ file (input.cpp and input2.cpp) using C lex/yacc.

The grammar of the syntactic rules of any one of input file for the parser will be union of some syntactic rules of C++ syntax grammar. (Not every syntactic characteristics of C++ !!)

You should write a lex file for the operation of the yacc (i.e, you should write return statements at the lex file).

If any one of syntax error is detected, the error checker should write the line number where the error is detected to the screen and terminate.

Before submitting your work, zip lex and yacc files. Submit your zip file to the Prelab2 part in Coadsys.

Important Notice: In your works, you should include a detailed comment in your code for usage when you will use any other statements, expressions and concepts which are not presented in labs. You can write your comments in Turkish. We will control your comments and if we do not think it is adequate to explain what is going to be done or you do not write any comment, your work will be graded out of 0 and behaved as a copied assignment.

You are responsible to send files which can be compiled clearly with lex or yacc or gcc compiler. If any files you send cannot be compiled nor the generated parser cannot be run clearly (i.e., generation of segmentation fault or run time errors), your work will be graded out of 30.

Every assignments will be controlled for **plagiarism** with **MOSS** system. We will add some selective yacc and lex files from last semester's assignments and internet sites. If any suspicious work is detected, it will be behaved as a copied assignment and punished strictly.

Deadline of works is given in Lab Plan. The times will not be changed.

Important Notice for this Work: Please do not include any other syntax characteristics of C++ which is not in example file. If we fetch such a case, your work will be graded out of 0 and behaved as a copied assignment.

In this prelab, you should not include %type, %term, %nonassoc, %start, variables started with \$, functions started with YACC_ and LEX_, yylval, attribute grammar semantic blocks which are not covered in labs for building your syntactic rules. In your yacc files, you should not include terminal symbols in static way such as 'X','[',']', etc ... If we fetch such a case, your work will be graded out of 0 and behaved as a copied assignment.

Please note that, the codes generated with lex and yacc will be compiled only with gcc compiler. If you add any one of C++ statement which are not C statement in lex or yacc file, your work will be graded out of 0 and behaved as a copied assignment.

Please note that, we can use your grammar error checker with different C++ input files with same syntax characteristics of example C++ file.

GOOD LUCK