# CSE 351 Programming Languages
# FALL 2018
# PROJECT
# (Due: 28.12.2018 23:59)
# Shell Script to Perl Converter

In this project, you will write a converter which converts a Shell Script code to Perl code and makes semantic and syntax error checking for input shell script file conversion to output perl file. This converter should be written using **lex-yacc** with **C Language**.

**Notes for the project**

1) You should write a Shell Script to Perl converter which will be aware of some of the Shell Script Code structures. These code structure includes assignment operations consisting all arithmetic operations with nested parenthesis, echo statements and nested if/elseif/else blocks. There will be several combinations of those statements and blocks in an input shell script file. Nested if blocks of shell script code can consist any number of statements and blocks stated above.
2) For the project assignment, write one lex and one yacc file. Include all of additional C functions, C like data structures, C global variables in the yacc file. Do not write any additional C code and header file for the project.
3) Your converter should, at least, convert the example input shell script files to the corresponding output perl files. In this project, you should write an appropriate lexical, syntactic and semantic rules for the conversion operation with input shell script files. The output of the converter should be an appropriate perl file of the corresponding input shell script file. The converted output perl codes should be compiled with perl compiler of any one of the Linux Distributions (e.g., any one of the version of Ubuntu).
4) If there is syntax and semantic error in input shell script file, the converter should warn user and terminate the operation. Syntax and semantic error could be any syntax and semantic errors which can be seen in a shell script files consisting nested paranthesis, echo statements and nested if/elseif/else blocks. Your converter should detect all of any one of these errors correctly. Your converter should at least print the line number of detected syntax or semantic error and exit, if it detects a error. **Although we will give some erroneous code examples to you, your converter will not be limited to detect only these errors and it should correctly detect all of the syntax and semantic errors in the shell script code statements described above.**
5) You should include a detailed comment in your converter's lex and yacc files when you use any statements, expressions and concepts which are not explained in lab sections. Please, feel free to write your comments in Turkish. We will control your comments and if we think the comments are not adequate to explain what is going to be done or you do not write any comment, your project will be graded out of 0 and could be accounted with a copied project. In the project demo, you shall be asked any questions about comments you will have written in the codes.

6) In the project demo, we can give extra errorneous or error free shell script code files to the converter as input. **We would give extra errorneous codes consisting error which are not in the given errorneous shell script files examples to you.** So

please write a generic converter and do not forget that, at least, your converter should convert given example input shell script files to the corresponding perl output files. **Note that, your converter should do conversion operation without need of manual modification to any input shell script files. In addition not to get a major grade reduction, the output perl code files of error free shell script input codes should be compiled and ran clearly.**

7) Project submissions will be controlled for plagiarism and copied projects will be strictly punished !!
8) **This work should be done individually. Collaborative or group works will not be accepted !!**
9) **For the compilation process in the demos, we only want you to compile just only one lex and one yacc files and the C codes generated from lex and yacc command. If the files are not clearly compiled or converter will generate a run time error such as a segmentation fault, your project will be evaluated under 30.**
10) All warnings for all works and lab which has been announced during the semester are valid for the project. Any additional announcements covering project or the procedure of the project demo could be done during the semester.
11) You are responsible with the files you will send for your project assignment. You should send one lex file and one yacc file for the project code. Any additional C source and header files will not be accounted in the project assignment. If you think it will be helpful in the demo, you can include a MAKEFILE to the project assignment. In addition, you should prepare a README file which explains clearly the directives for building and running your converter using files you will send for your project assignment. Additionally, you should prepare a short report which explains the state of your development for the project. In this report, you should write the statements of shell script code which your converter is able to convert and not convert. You can mention about the example files which can be converted or not converted in this report. This report should be in PDF format. **In a such case that you do not send all necessary lex and yacc files, README file which explains clearly the directives to build and run the converter and the short report PDF file, we could consider your project as not working without asking.**
12) Please send your files corresponding to the project (i.e., at least, the files that we can build and run your converter clearly, the example input files you will have studied and the README file which consists directives how you build and run your converter, and the short report pdf file) in a **zip file** to the ***COADSYS*** link for the project until 28.12.2018 23:59. Name your file as <your_name>_<your_surname>_CSE351Project.zip ( example: Kemal_Serdaroglu_CSE351Project.zip). **Assignment files which is compressed with other compression types (tar.gz, rar etc.) will not be taken into consideration.**
13) **The late assignments and assignments via E-mail will not be accepted !!**

**BONUS:** You will get extra credit, if your converter convert the statements or/and detect any syntax and semantic error of while loop/nested while loop with nested if consisting all echo and assignment statements of shell script to the corresponding perl statements.

GOOD LUCK