# Assessment #2

In this assessment we will be testing your ability on understanding and implementing the syntax analysis portion of the compilation process. This is done with the use either top-down or bottom-up parsers. This will be partially practical with an optional portion take will allow you to choose between practical and conceptual problems.

### PRACTICAL- 50 points

In this problem you will be designing code that should be able to take in a file that is analyzed for lexical and syntactical correctness. This programming language should allow for the following type of statements:

- switch
- foreach
- for
- while
- do-while
- block
- if
- assignment
- return

Imagine this programming language also includes the following rules:

<program> → VOID MAIN ’(‘ ’)’ <block>
<block> → ‘{‘ { <statement> ; } ‘}’

*** Code should be able to detect syntax and lexical errors. You may choose the syntactical structure of each statement, but they must be that of another programming language that you specify in the comments and in a word doc where you list the grammar rules of your language.

YOUR CHOICE – Choose remaining 50 points

1. 25 points
   Create an LR Parsing table for the following grammar (10 Points) and show the steps to solve the following problems
   S -> a C | A C
   A -> a B b | A a
   B -> b B | c C
   C -> c C | d

   a. abbbccd
   b. accd
   c. acdbaacd
   d. acdbd
   e. abcdbad

2. (50 points) Given the grammar from the previous problem, programmatically show where each string fails the grammar or create a parse tree showing they are in the language
   a. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.218.1084&rep=rep1&type=pdf

3. (25 points) Given the grammar from the previous problem if the string is in the language show the parse tree, right most derivation, handle, phrases and simple phrases for the following strings, or prove they are not in the language

   a. abBbcC
   b. accCd
   c. aCbaacd
   d. acdabd
   e. abCbad