Compilers Design

CSCI415 Fall 21

**Project phase 2**

After implementing the scanner in phase 1 , now we need to implement the parser for the JSON language. The required deliverables are:

1. **Write the grammar rules for the JSON language ,** the rules should include

Program rule : this rule should be the starting symbol of your grammar and from it we can move to all the rules for example

**Program -> begin statements\_block end**

Then statements block should be the oring of all statements of your language:

* **Declare statement**
* **Assign statement**
* **Expression**
* **Condition**
* **While\_loop**
* **Function**
* **Read statement**
* **Write statement**

**For each statement you will identify its rule.**

1. **After identifying the grammar rules separately ( in a normal text format ), you should implement the parser from scratch using any language. The input of the parser should be the tokens from the scanner implemented in phase 1.** Sample input output is shown in the table:

|  |  |
| --- | --- |
| tokens | Parser output |
| Integer id semicolon | Declare statement: Integer id semicolon |
| While id ! num\_const Do | While statement: while expression Do  Expression: id ! num\_const |
| Write id semiclon | Write statement: write id semicolon |

The actual parser output should be in the form of parse tree like those in ANTLR but for simplicity you can write the output in the form of text like shown in the table. The grammar rule should be identified then it should also be clear how the grammar rule is derived to match the input. For example, in the while statement there is an expression, so the while is derived then the expression is derived to reach the input.

**Bonus:** you can use graphical library to represent the parser output in the form of a tree. This work is extra and it will be considered bonus. ( 2 to 3 marks)