**Formal Language and Compiler Design**

*Language grammar*

**Statement:**

Starting from syntax rules in BNF notation from Lab 1, construct the context free grammar corresponding to your minilanguage to be used in parsing.

**G = (N, E, P, S)**

N = {Identifier, Letter, Digit, Noconst, No, Character, Constchar, String, Program, Stmtlist, Stmt, Declaration, Type, Simpletype, Arraydeclaration, Simplestmt, Assignstmt, Expression, Operator, Iostmt, Structstmt, Ifstmt, Whilestmt, Condition, Relation}

E = {program, if, else, while, int, string, char, boolean, read, write, +, -, \*, /, <, >, =, ==, <=, >=, !=, ||, &&, (, ), [, ], ;}

P = {S->Program,

Program -> program { Stmtlist },

Stmtlist -> Stmt | Stmt ; Stmtlist,

Stmt -> Simplestmt | Structstmt | Declaration,

Declaration -> Type Identifier | Type Identifier , Declaration,

Type -> Simpletype | Arraydeclaration,

Simpletype -> int | char | string | boolean | char,

Arraydeclaration -> Simpletype[Noconst],

Simplestmt -> Assignstmt | Iostmt,

Assignstmt -> Identifier = Expression,

Expression -> Expression Operator Identifier | Identifier,

Operator -> + | - | \* | /,

Iostmt -> read Identifier | write Identifier,

Structstmt -> Ifstmt | Whilestmt,

Ifstmt -> if (Condition) {Stmtlist},

Whilestmt -> while (Condition) {Stmtlist},

Condition -> Expression Relation Expression,

Relation -> < | > | == | != | <= | >= | || | &&

}