

## GRAMMAR

1.  $\langle \text{program} \rangle \rightarrow \langle \text{moduleDeclarations} \rangle \langle \text{otherModules} \rangle \langle \text{driverModule} \rangle \langle \text{otherModules} \rangle$
2.  $\langle \text{moduleDeclarations} \rangle \rightarrow \langle \text{moduleDeclaration} \rangle \langle \text{moduleDeclarations} \rangle \mid \epsilon$
3.  $\langle \text{moduleDeclaration} \rangle \rightarrow \text{DECLARE MODULE ID SEMICOL}$
4.  $\langle \text{otherModules} \rangle \rightarrow \langle \text{module} \rangle \langle \text{otherModules} \rangle \mid \epsilon$
5.  $\langle \text{module} \rangle \rightarrow \text{DEF MODULE ID ENDDF TAKES INPUT SQBO } \langle \text{input\_plist} \rangle \text{ SQBC SEMICOL } \langle \text{ret} \rangle \langle \text{moduleDef} \rangle$
6.  $\langle \text{ret} \rangle \rightarrow \text{RETURNS SQBO } \langle \text{output\_plist} \rangle \text{ SQBC SEMICOL } \mid \epsilon$
7.  $\langle \text{input\_plist} \rangle \rightarrow \text{ID COLON } \langle \text{dataType} \rangle \langle \text{input\_plist\_dash} \rangle$
8.  $\langle \text{input\_plist\_dash} \rangle \rightarrow \text{COMMA ID COLON } \langle \text{dataType} \rangle \langle \text{input\_plist\_dash} \rangle \mid \epsilon$
9.  $\langle \text{output\_plist} \rangle \rightarrow \text{ID COLON } \langle \text{type} \rangle \langle \text{output\_plist\_dash} \rangle$
10.  $\langle \text{output\_plist\_dash} \rangle \rightarrow \text{COMMA ID COLON } \langle \text{type} \rangle \langle \text{output\_plist\_dash} \rangle \mid \epsilon$
11.  $\langle \text{dataType} \rangle \rightarrow \text{INTEGER} \mid \text{BOOLEAN} \mid \text{REAL} \mid \text{ARRAY SQBO } \langle \text{range\_array} \rangle \text{ SQBC OF } \langle \text{type} \rangle$
12.  $\langle \text{range\_array} \rangle \rightarrow \langle \text{index} \rangle \text{ RANGEOP } \langle \text{index} \rangle$
13.  $\langle \text{type} \rangle \rightarrow \text{INTEGER} \mid \text{REAL} \mid \text{BOOLEAN}$
14.  $\langle \text{driverModule} \rangle \rightarrow \text{DRIVERDEF DRIVER PROGRAM DRIVERENDDF } \langle \text{moduleDef} \rangle$
15.  $\langle \text{moduleDef} \rangle \rightarrow \text{START } \langle \text{statements} \rangle \text{ END}$
16.  $\langle \text{statements} \rangle \rightarrow \langle \text{statement} \rangle \langle \text{statements} \rangle \mid \epsilon$
17.  $\langle \text{statement} \rangle \rightarrow \langle \text{ioStmt} \rangle \mid \langle \text{simpleStmt} \rangle \mid \langle \text{declareStmt} \rangle \mid \langle \text{conditionalStmt} \rangle \mid \langle \text{iterativeStmt} \rangle$
18.  $\langle \text{declareStmt} \rangle \rightarrow \text{DECLARE } \langle \text{idList} \rangle \text{ COLON } \langle \text{dataType} \rangle \text{ SEMICOL}$
19.  $\langle \text{idList} \rangle \rightarrow \text{ID } \langle \text{idList\_dash} \rangle$
20.  $\langle \text{idList\_dash} \rangle \rightarrow \text{COMMA ID } \langle \text{idList\_dash} \rangle \mid \epsilon$
21.  $\langle \text{ioStmt} \rangle \rightarrow \text{GET\_VALUE BO ID BC SEMICOL} \mid \text{PRINT BO } \langle \text{var} \rangle \text{ BC SEMICOL}$

22.  $\langle \text{var} \rangle \rightarrow \text{ID} \langle \text{whichId} \rangle \mid \text{NUM} \mid \text{RNUM} \mid \text{TRUE} \mid \text{FALSE}$
23.  $\langle \text{whichId} \rangle \rightarrow \text{SQBO} \langle \text{index} \rangle \text{SQBC} \mid \epsilon$
24.  $\langle \text{simpleStmt} \rangle \rightarrow \langle \text{assignmentStmt} \rangle \mid \langle \text{moduleReuseStmt} \rangle$
25.  $\langle \text{assignmentStmt} \rangle \rightarrow \text{ID} \langle \text{whichStmt} \rangle$
26.  $\langle \text{whichStmt} \rangle \rightarrow \langle \text{lvalueIDStmt} \rangle \mid \langle \text{lvalueARRStmt} \rangle$
27.  $\langle \text{lvalueIDStmt} \rangle \rightarrow \text{ASSIGNOP} \langle \text{expression} \rangle \text{SEMICOL}$
28.  $\langle \text{lvalueARRStmt} \rangle \rightarrow \text{SQBO} \langle \text{index} \rangle \text{SQBC} \text{ASSIGNOP} \langle \text{expression} \rangle \text{SEMICOL}$
29.  $\langle \text{index} \rangle \rightarrow \text{NUM} \mid \text{ID}$
30.  $\langle \text{moduleReuseStmt} \rangle \rightarrow \langle \text{optional} \rangle \text{USE MODULE ID WITH PARAMETERS} \langle \text{idList} \rangle \text{SEMICOL}$
31.  $\langle \text{optional} \rangle \rightarrow \text{SQBO} \langle \text{idList} \rangle \text{SQBC} \text{ASSIGNOP} \mid \epsilon$
32.  $\langle \text{expression} \rangle \rightarrow \langle \text{arithmeticOrBooleanExpression} \rangle \mid \langle \text{op\_plus\_minus} \rangle \langle \text{unaryOrExpr} \rangle$
33.  $\langle \text{unaryOrExpr} \rangle \rightarrow \text{BO} \langle \text{arithmeticExpr} \rangle \text{BC} \mid \text{ID} \mid \text{NUM} \mid \text{RNUM}$
34.  $\langle \text{arithmeticOrBooleanExpression} \rangle \rightarrow \langle \text{arithmetic\_bool} \rangle \langle \text{bool} \rangle$
35.  $\langle \text{arithmetic\_bool} \rangle \rightarrow \langle \text{arithmeticExpr} \rangle \langle \text{arithmeticExpre\_dash} \rangle$
36.  $\langle \text{bool} \rangle \rightarrow \langle \text{logicalOp} \rangle \langle \text{arithmetic\_bool} \rangle \langle \text{bool} \rangle \mid \epsilon$
37.  $\langle \text{arithmeticExpre\_dash} \rangle \rightarrow \langle \text{relationalOp} \rangle \langle \text{arithmeticExpr} \rangle \langle \text{arithmeticExpre\_dash} \rangle \mid \epsilon$
38.  $\langle \text{arithmeticExpr} \rangle \rightarrow \langle \text{term} \rangle \langle \text{arithmeticExpr\_recur} \rangle$
39.  $\langle \text{arithmeticExpr\_recur} \rangle \rightarrow \langle \text{op\_plus\_minus} \rangle \langle \text{term} \rangle \langle \text{arithmeticExpr\_recur} \rangle \mid \epsilon$
40.  $\langle \text{term} \rangle \rightarrow \langle \text{factor} \rangle \langle \text{term\_dash} \rangle$
41.  $\langle \text{term\_dash} \rangle \rightarrow \langle \text{op\_mul\_div} \rangle \langle \text{factor} \rangle \langle \text{term\_dash} \rangle \mid \epsilon$
42.  $\langle \text{factor} \rangle \rightarrow \text{BO} \langle \text{arithmeticExpr} \rangle \text{BC} \mid \langle \text{var} \rangle$
43.  $\langle \text{op\_plus\_minus} \rangle \rightarrow \text{PLUS} \mid \text{MINUS}$
44.  $\langle \text{op\_mul\_div} \rangle \rightarrow \text{MUL} \mid \text{DIV}$
45.  $\langle \text{logicalOp} \rangle \rightarrow \text{AND} \mid \text{OR}$
46.  $\langle \text{relationalOp} \rangle \rightarrow \text{GT} \mid \text{LT} \mid \text{GE} \mid \text{LE} \mid \text{EQ} \mid \text{NE}$

47.  $\langle \text{conditionalStmt} \rangle \rightarrow \text{SWITCH BO ID BC START } \langle \text{caseStmts} \rangle \langle \text{default} \rangle \text{ END}$
48.  $\langle \text{caseStmts} \rangle \rightarrow \text{CASE } \langle \text{value} \rangle \text{ COLON } \langle \text{statements} \rangle \text{ BREAK SEMICOL } \langle \text{caseStmt} \rangle$
49.  $\langle \text{caseStmt} \rangle \rightarrow \text{CASE } \langle \text{value} \rangle \text{ COLON } \langle \text{statements} \rangle \text{ BREAK SEMICOL } \langle \text{caseStmt} \rangle \mid \epsilon$
50.  $\langle \text{value} \rangle \rightarrow \text{NUM} \mid \text{TRUE} \mid \text{FALSE}$
51.  $\langle \text{default} \rangle \rightarrow \text{DEFAULT COLON } \langle \text{statements} \rangle \text{ BREAK SEMICOL} \mid \epsilon$
52.  $\langle \text{iterativeStmt} \rangle \rightarrow \text{FOR BO ID IN } \langle \text{range} \rangle \text{ BC START } \langle \text{statements} \rangle \text{ END} \mid \text{WHILE BO } \langle \text{arithmeticOrBooleanExpression} \rangle \text{ BC START } \langle \text{statements} \rangle \text{ END}$
53.  $\langle \text{range} \rangle \rightarrow \text{NUM RANGEOP NUM}$

## ASSUMPTIONS

1. The print statement can print ID, constant, true, false and any indices of an array.
2. We do not allow empty statements, i.e. statements having only a semicolon.
3. Negated expressions such as  $\text{-(a+b)}$  etc. are allowed, where a,b are identifiers.
4. We have used  $\langle \text{range\_array} \rangle$  for RANGEOP operation of array where the limits can be integers or identifiers and  $\langle \text{range} \rangle$  for inside the **for loop** where the limits can be only integers.