ERPLAG - Grammar

1. ⟨*program*⟩ → ⟨*moduleDeclarations*⟩ ⟨*otherModules*⟩ ⟨*driverModule*⟩ ⟨*otherModules*⟩
   (a) ⟨*moduleDeclarations*⟩ → ⟨*moduleDeclaration*⟩ ⟨*moduleDeclarations*⟩ | ϵ
   (b) ⟨*moduleDeclaration*⟩ → **DECLARE MODULE ID SEMICOL**
   (c) ⟨*otherModules*⟩ → ⟨*module*⟩ ⟨*otherModules*⟩ | ϵ
   (d) ⟨*driverModule*⟩ → **DRIVERDEF DRIVER PROGRAM DRIVERENDDEF** ⟨*moduleDef*⟩
   (e) ⟨*module*⟩ → **DEF MODULE ID ENDDEF TAKES INPUT SQBO** ⟨*input_plist*⟩ **SQBC SEMICOL** ⟨*ret*⟩ ⟨*moduleDef*⟩
   (f) ⟨*moduleDef*⟩ → **START** ⟨*statements*⟩ **END**
   (g) ⟨*ret*⟩ → **RETURNS SQBO** ⟨*output_plist*⟩ **SQBC SEMICOL** | ϵ
   (h) ⟨*input_plist*⟩ → **ID COLON** ⟨*dataType*⟩ ⟨*IPL*⟩
   (i) ⟨*IPL*⟩ → **COMMA ID COLON** ⟨*dataType*⟩ ⟨*IPL*⟩ | ϵ
   (j) ⟨*output_plist*⟩ → **ID COLON** ⟨*dataType*⟩ ⟨*OPL*⟩
   (k) ⟨*OPL*⟩ → **COMMA ID COLON** ⟨*dataType*⟩ ⟨*OPL*⟩ | ϵ
2. ⟨*statements*⟩ → ⟨*statement*⟩ ⟨*statements*⟩ | ϵ
3. ⟨*statement*⟩ → ⟨*ioStmt*⟩ | ⟨*simpleStmt*⟩ | ⟨*conditionalStmt*⟩ | ⟨*declareStmt*⟩ | ⟨*iterativeStmt*⟩
   (a) ⟨*ioStmt*⟩ → **GET_VALUE BO ID BC SEMICOL** | **PRINT BO** ⟨*print_var*⟩ **BC SEMICOL**
      i. ⟨*print_var*⟩ → ⟨*var*⟩ | **TRUE** | **FALSE**
   (b) ⟨*simpleStmt*⟩ → ⟨*assignmentStmt*⟩
      i. ⟨*assignmentStmt*⟩ → **ID** ⟨*whichStmt*⟩
      ii. ⟨*whichStmt*⟩ → ⟨*lvalueIDStmt*⟩ | ⟨*lvalueARRStmt*⟩
      iii. ⟨*lvalueIDStmt*⟩ → **ASSIGNOP** ⟨*expression_new*⟩ **SEMICOL**
      iv. ⟨*lvalueARRStmt*⟩ → **SQBO SQBC ASSIGNOP** ⟨*expression_new*⟩ **SEMICOL**
   (c) ⟨*simpleStmt*⟩ → ⟨*moduleReuseStmt*⟩
      i. ⟨*moduleReuseStmt*⟩ → ⟨*optional*⟩ **USE MODULE ID WITH PARAMETERS** ⟨*idList*⟩ **SEMICOL**
      ii. ⟨*optional*⟩ → **SQBO** ⟨*idList*⟩ **SQBC ASSIGNOP** | ϵ
   (d) ⟨*declareStmt*⟩ → **DECLARE** ⟨*idList*⟩ **COLON** ⟨*dataType*⟩ **SEMICOL**
   (e) ⟨*iterativeStmt*⟩ → **FOR BO IN** ⟨*range*⟩ **BC START** ⟨*statements*⟩ **END**
   (f) ⟨*iterativeStmt*⟩ → **WHILE BO** ⟨*expression*⟩ **BC START** ⟨*statements*⟩ **END**
   (g) ⟨*conditionalStmt*⟩ → **SWITCH BO ID BC START CASE** ⟨*value*⟩ **COLON** ⟨*statements*⟩ **BREAK SEMICOL** ⟨*caseStmt*⟩ ⟨*default*⟩ **END**
      i. ⟨*caseStmt*⟩ → **CASE** ⟨*value*⟩ **COLON** ⟨*statements*⟩ **BREAK SEMICOL** ⟨*caseStmt*⟩ | ϵ
      ii. ⟨*default*⟩ → **DEFAULT COLON** ⟨*statements*⟩ **BREAK SEMICOL** | ϵ

4. ⟨*expression_new*⟩ → ⟨*expression*⟩ | ⟨*U*⟩
5. ⟨*U*⟩ → **PLUS** ⟨*factor_new*⟩ | **MINUS** ⟨*factor_new*⟩
6. ⟨*expression*⟩ → ⟨*boolTerm*⟩ ⟨*bT*⟩
7. ⟨*bT*⟩ → ⟨*logicalOp*⟩ ⟨*boolTerm*⟩ ⟨*bT*⟩ | ϵ
    (a) ⟨*boolTerm*⟩ → ⟨*arithmeticExpr*⟩ ⟨*aE*⟩ | **TRUE** | **FALSE**
    (b) ⟨*aE*⟩ → ⟨*relationalOp*⟩ ⟨*arithmeticExpr*⟩ | ϵ
        i. ⟨*arithmeticExpr*⟩ → ⟨*term*⟩ ⟨*aT*⟩
        ii. ⟨*aT*⟩ → ⟨*pmop*⟩ ⟨*term*⟩ ⟨*aT*⟩ | ϵ
            A. ⟨*term*⟩ → ⟨*factor*⟩ ⟨*aF*⟩
            B. ⟨*aF*⟩ → ⟨*mdop*⟩ ⟨*factor*⟩ ⟨*aF*⟩ | ϵ
            C. ⟨*factor*⟩ → **BO** ⟨*expression*⟩ **BC** | ⟨*var*⟩
            D. ⟨*factor_new*⟩ → **BO** ⟨*arithmeticExpr*⟩ **BC** | ⟨*var*⟩
8. ⟨*logicalOp*⟩ → **AND** | **OR**
9. ⟨*relationalOp*⟩ → **LT** | **LE** | **GT** | **GE** | **EQ** | **NE**
10. ⟨*pmop*⟩ → **PLUS** | **MINUS**
11. ⟨*mdop*⟩ → **MUL** | **DIV**
12. ⟨*dataType*⟩ → **INTEGER** | **REAL** | **BOOLEAN** | **ARRAY SQBO** ⟨*range*⟩ **SQBC OF** ⟨*type*⟩
13. ⟨*type*⟩ → **INTEGER** | **REAL** | **BOOLEAN**
14. ⟨*var*⟩ → **ID** ⟨*whichID*⟩ | **NUM** | **RNUM**
15. ⟨*whichID*⟩ → **SQBO** ⟨*index_new*⟩ **SQBC** | ϵ
16. ⟨*index_new*⟩ → **NUM** | **ID**
17. ⟨*value*⟩ → **NUM** | **TRUE** | **FALSE**
18. ⟨*range*⟩ → ⟨*index_new*⟩ **RANGEOP** ⟨*index_new*⟩
19. ⟨*range_new*⟩ → **NUM RANGEOP NUM**
20. ⟨*idList*⟩ → **ID** ⟨*idL*⟩
21. ⟨*idL*⟩ → **COMMA ID** ⟨*idL*⟩ | ϵ

1 FIRST Set

1. ⟨*program*⟩ := { **DECLARE DEF DRIVERDEF** }
   (a) ⟨*moduleDeclarations*⟩ := { **DECLARE** ϵ }
   (b) ⟨*moduleDeclaration*⟩ := { **DECLARE** }
   (c) ⟨*otherModules*⟩ := { **DEF** ϵ }
   (d) ⟨*driverModule*⟩ := { **DRIVERDEF** }
   (e) ⟨*module*⟩ := { **DEF** }
   (f) ⟨*moduleDef*⟩ := { **START** }
   (g) ⟨*ret*⟩ := { **RETURNS** ϵ }
   (h) ⟨*input_plist*⟩ := { **ID** }
   (i) ⟨*IPL*⟩ := { **COMMA** ϵ }
   (j) ⟨*output_plist*⟩ := { **ID** }
   (k) ⟨*OPL*⟩ := { **COMMA** ϵ }
2. ⟨*statements*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE** ϵ }
3. ⟨*statements*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE** }
   (a) ⟨*ioStmt*⟩ := { **GET_VALUE PRINT** }
      i. ⟨*print_var*⟩ := { **TRUE FALSE ID NUM RNUM** }
   (b) ⟨*simpleStmt*⟩ := { **ID USE SQBO** }
      i. ⟨*assignmentStmt*⟩ := { **ID** }
      ii. ⟨*whichStmt*⟩ := { **ASSIGNOP SQBO** }
      iii. ⟨*lvalueIDStmt*⟩ := { **ASSIGNOP** }
      iv. ⟨*lvalueARRStmt*⟩ := { **SQBO** }
      v. ⟨*moduleReuseStmt*⟩ := { **USE SQBO** }
      vi. ⟨*optional*⟩ := { **SQBO** ϵ }
   (c) ⟨*declareStmt*⟩ := { **DECLARE** }
   (d) ⟨*iterativeStmt*⟩ := { **FOR WHILE** }
   (e) ⟨*conditionalStmt*⟩ := { **SWITCH** }
      i. ⟨*caseStmt*⟩ := { **CASE** ϵ }
      ii. ⟨*default*⟩ := { **DEFAULT** ϵ }
4. ⟨*expression*⟩ := { **PLUS MINUS BO ID NUM RNUM TRUE FALSE** }
5. ⟨*bT*⟩ := { **AND OR** ϵ }
   (a) ⟨*boolTerm*⟩ := { **PLUS MINUS BO ID NUM RNUM TRUE FALSE** }
   (b) ⟨*aE*⟩ := { **LT LE GT GE EQ NE** ϵ }
      i. ⟨*arithmeticExpr*⟩ := { **PLUS MINUS BO ID NUM RNUM** }
      ii. ⟨*aT*⟩ := { **PLUS MINUS** ϵ }
         A. ⟨*term*⟩ := { **PLUS MINUS BO ID NUM RNUM** }
         B. ⟨*aF*⟩ := { **MUL DIV** ϵ }

C. $\langle factor \rangle$:= { **PLUS MINUS BO ID NUM RNUM** }

6. $\langle logicalOp \rangle$:= { **AND OR** }
7. $\langle relationalOp \rangle$:= { **LT LE GT GE EQ NE** }
8. $\langle pmop \rangle$:= { **PLUS MINUS** }
9. $\langle mdop \rangle$:= { **MUL DIV** }
10. $\langle dataType \rangle$:= { **INTEGER REAL BOOLEAN ARRAY** }
11. $\langle type \rangle$:= { **INTEGER REAL BOOLEAN** }
12. $\langle var \rangle$:= { **ID NUM RNUM** }
13. $\langle whichID \rangle$:= { **SQBO** ϵ }
14. $\langle index\_new \rangle$:= { **NUM ID RANGEOP** }
15. $\langle value \rangle$:= { **NUM TRUE FALSE** }
16. $\langle range \rangle$:= { **NUM ID** }
17. $\langle idList \rangle$:= { **ID** }
18. $\langle idL \rangle$:= { **COMMA** ϵ }

2 FOLLOW Set

1. ⟨*program*⟩ := { **$** }
   (a) ⟨*moduleDeclarations*⟩ := { **DEF DRIVERDEF** }
   (b) ⟨*moduleDeclaration*⟩ := { **DECLARE DEF DRIVERDEF** }
   (c) ⟨*otherModules*⟩ := { **DRIVERDEF $** }
   (d) ⟨*driverModule*⟩ := { **DEF $** }
   (e) ⟨*module*⟩ := { **DEF DRIVERDEF $** }
   (f) ⟨*moduleDef*⟩ := { **DEF DRIVERDEF $** }
   (g) ⟨*ret*⟩ := { **START** }
   (h) ⟨*input_plist*⟩ := { **SQBC** }
   (i) ⟨*IPL*⟩ := { **SQBC** }
   (j) ⟨*output_plist*⟩ := { **SQBC** }
   (k) ⟨*OPL*⟩ := := { **SQBC** }
2. ⟨*statements*⟩ := { **END BREAK** }
3. ⟨*statement*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
   (a) ⟨*ioStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      i. ⟨*print_var*⟩ := { **BC** }
   (b) ⟨*simpleStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      i. ⟨*assignmentStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      ii. ⟨*whichStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      iii. ⟨*lvalueIDStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      iv. ⟨*lvalueARRStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      v. ⟨*moduleReuseStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      vi. ⟨*optional*⟩ := { **USE** }
   (c) ⟨*declareStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
   (d) ⟨*iterativeStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
   (e) ⟨*conditionalStmt*⟩ := { **GET_VALUE PRINT ID USE SQBO SWITCH DECLARE FOR WHILE END BREAK** }
      i. ⟨*caseStmt*⟩ := { **DEFAULT END** }
      ii. ⟨*default*⟩ := { **END** }

4. ⟨*expression*⟩ := { **SEMICOL BC** }
5. ⟨*bT*⟩ := { **SEMICOL BC** }
    (a) ⟨*boolTerm*⟩ := { **AND OR SEMICOL BC** }
    (b) ⟨*aE*⟩ := { **AND OR SEMICOL BC** }
        i. ⟨*arithmeticExpr*⟩ := { **LT LE GT GE EQ NE AND OR SEMICOL BC** }
        ii. ⟨*aT*⟩ := { **LT LE GT GE EQ NE AND OR SEMICOL BC** }
            A. ⟨*term*⟩ := { **PLUS MINUS LT LE GT GE EQ NE AND OR SEMICOL BC** }
            B. ⟨*aF*⟩ := { **PLUS MINUS LT LE GT GE EQ NE AND OR SEMICOL BC** }
            C. ⟨*factor*⟩ := { **MUL DIV PLUS MINUS LT LE GT GE EQ NE AND OR SEMICOL BC** }
6. ⟨*logicalOp*⟩ := { **PLUS MINUS TRUE FALSE BO ID NUM RNUM** }
7. ⟨*relationalOp*⟩ := { **PLUS MINUS BO ID NUM RNUM** }
8. ⟨*pmop*⟩ := { **PLUS MINUS BO ID NUM RNUM** }
9. ⟨*mdop*⟩ := { **PLUS MINUS BO ID NUM RNUM** }
10. ⟨*dataType*⟩ := { **COMMA SQBC SEMICOL** }
11. ⟨*type*⟩ := { **COMMA SQBC SEMICOL** }
12. ⟨*var*⟩ := { **MUL DIV PLUS MINUS LT LE GT GE EQ NE AND OR SEMICOL BC** }
13. ⟨*whichID*⟩ := { **MUL DIV PLUS MINUS LT LE GT GE EQ NE AND OR SEMICOL BC** }
14. ⟨*index_new*⟩ := { **SQBC** }
15. ⟨*value*⟩ := { **COLON** }
16. ⟨*range*⟩ := { **BC SQBC** }
17. ⟨*idList*⟩ := { **SEMICOL SQBC COLON** }
18. ⟨*idL*⟩ := { **SEMICOL SQBC COLON** }