**GROUP 39**

**AKANKSHYA MISHRA 2016A7PS0026P**

**NARAPAREDDY BHAVANA 2016A7PS0034P**

**KARABEE BATTA 2016A7PS0052P**

**AASTHA KATARIA 2016A7PS0062P**

Grammar:

*The nonterminal <program> is the start symbol of the given grammar.*

1. <program> ===> <otherFunctions> <mainFunction>

2. <mainFunction>===> TK\_MAIN <stmts> TK\_END

3. <otherFunctions>===> <function><otherFunctions> | eps

4. <function>===>TK\_FUNID <input\_par> <output\_par> TK\_SEM <stmts> TK\_END

5. <input\_par>===>TK\_INPUT TK\_PARAMETER TK\_LIST TK\_SQL <parameter\_list> TK\_SQR

6. <output\_par>===>TK\_OUTPUT TK\_PARAMETER TK\_LIST TK\_SQL <parameter\_list> TK\_SQR | eps

7. <parameter\_list>===><dataType> TK\_ID <remaining\_list>

8. <dataType>===> <primitiveDatatype> |<constructedDatatype>

9. <primitiveDatatype>===> TK\_INT | TK\_REAL

10. <constructedDatatype>===>TK\_RECORD TK\_RECORDID

11. <remaining\_list>===>TK\_COMMA <parameter\_list> | eps

12. <stmts>===><typeDefinitions> <declarations> <otherStmts><returnStmt>

13. <typeDefinitions>===><typeDefinition><typeDefinitions> |eps

14. <typeDefinition>===>TK\_RECORD TK\_RECORDID <fieldDefinitions> TK\_ENDRECORD TK\_SEM

15. <fieldDefinitions>===> <fieldDefinition><fieldDefinition><moreFields>

16. <fieldDefinition>===> TK\_TYPE <primitiveDatatype> TK\_COLON TK\_FIELDID TK\_SEM

17. <moreFields>===><fieldDefinition><moreFields> | eps

18. <declarations> ===> <declaration><declarations>|eps

**19 a.<declaration>===> TK\_TYPE <dataType> TK\_COLON TK\_ID <declaration’> TK\_SEM**

**19 b. <declaration’> === TK\_COLON <global\_or\_not> | eps**

20. <global\_or\_not>===>TK\_GLOBAL| eps

21. <otherStmts>===> <stmt><otherStmts> | eps

22. <stmt>===> <assignmentStmt> | <iterativeStmt>|<conditionalStmt>|<ioStmt>| <funCallStmt>

23. <assignmentStmt>===><SingleOrRecId> TK\_ASSIGNOP <arithmeticExpression> TK\_SEM

**24 a. <singleOrRecId>===> TK\_ID <singleOrRecId’>**

**24 b. <singleOrRecId’> ===> TK\_DOT TK\_FIELDID | eps**

25. <funCallStmt>===><outputParameters> TK\_CALL TK\_FUNID TK\_WITH TK\_PARAMETERS <inputParameters>

26. <outputParameters> ==> TK\_SQL <idList> TK\_SQR TK\_ASSIGNOP | eps

27. <inputParameters>===> TK\_SQL <idList> TK\_SQR

28. <iterativeStmt>===> TK\_WHILE TK\_OP <booleanExpression> TK\_CL <stmt><otherStmts> TK\_ENDWHILE

**29. <conditionalStmt>===> TK\_IF <booleanExpression> TK\_THEN <stmt><otherStmts> <conditionalStmt’>**

**30. <conditionalStmt’>===> TK\_ELSE <otherStmts> TK\_ENDIF | TK\_ENDIF**

31. <ioStmt>===>TK\_READ TK\_OP **TK\_ID** TK\_CL TK\_SEM | TK\_WRITE TK\_OP **<var>** TK\_CL TK\_SEM

**32. \*Deleted\***

**33 a. <arithmeticExpression> ===> <term><arithmeticExpression’>**

**33 b. <arithmeticExpression’> ===> <add\_sub\_operator> <term> <arithmeticExpression’> | eps**

**33 c. <term> ===> <factor><term’>**

**33 d. <term’> ===> <mul\_div\_operator> <factor> <term’> | eps**

**34. <factor> ===> TK\_OP <arithmeticExpression> TK\_CL | <var>**

**35 a. <add\_sub\_operator> ===> TK\_PLUS | TK\_MINUS**

**35 b. <mul\_div\_operator> ===> TK\_MUL | TK\_DIV**

36. <booleanExpression>===>TK\_OP <booleanExpression> TK\_CL <logicalOp> TK\_OP <booleanExpression> TK\_CL

37. <booleanExpression>===> <var> <relationalOp> <var>

38. <booleanExpression>===> TK\_NOT <booleanExpression>

39. <var>===> TK\_ID | TK\_NUM | TK\_RNUM

40. <logicalOp>===>TK\_AND | TK\_OR

41. <relationalOp>===> TK\_LT | TK\_LE | TK\_EQ |TK\_GT | TK\_GE | TK\_NE

42. <returnStmt>===>TK\_RETURN <optionalReturn> TK\_SEM

43. <optionalReturn>===>TK\_SQL <idList> TK\_SQR | eps

44. <idList>===> TK\_ID <more\_ids>

45. <more\_ids>===> TK\_COMMA <idList> | eps

FIRST Sets:

1. first(<program>) = { TK\_FUNID, TK\_MAIN }
2. first(<mainFunction>) = { TK\_MAIN }
3. first(<otherFunctions>) = { TK\_FUNID, eps }
4. first(<function>) = { TK\_FUNID }
5. first(<input\_par>) = { TK\_INPUT }
6. first(<output\_par>) = { TK\_OUTPUT, eps }
7. first(<parameter\_list>) = { TK\_INT, TK\_REAL, TK\_RECORD }
8. first(<dataType>) = { TK\_INT, TK\_REAL, TK\_RECORD }
9. first(<primitiveDatatype>) = { TK\_INT, TK\_REAL }
10. first(<constructedDatatype>) = { TK\_RECORD }
11. first(<remaining\_list>) = { TK\_COMMA, eps }
12. first(<stmts>) = { TK\_RECORD, TK\_TYPE, TK\_ID, TK\_RECORDID, TK\_IF, TK\_READ, TK\_WRITE, TK\_WHILE, TK\_SQL, TK\_CALL, TK\_RETURN }
13. first(<typeDefinitions>) = { TK\_RECORD, eps }
14. first(<typeDefinition>) = { TK\_RECORD }
15. first(<fieldDefinitions>) = { TK\_TYPE }
16. first(<fieldDefinition>) = { TK\_TYPE }
17. first(<moreFields>) = { TK\_TYPE, eps }
18. first(<declarations>) = { TK\_TYPE, eps }
19. a. first(<declaration>) = { TK\_TYPE }

b. first(<declaration’>) = {TK\_COLON, eps}

1. first(<global\_or\_not>) = { TK\_GLOBAL, eps }
2. first(<otherStmts>) = { TK\_ID, TK\_RECORDID, TK\_IF, TK\_READ, TK\_WRITE, TK\_WHILE, TK\_SQL, TK\_CALL, eps }
3. first(<stmt>) = { TK\_ID, TK\_RECORDID, TK\_IF, TK\_READ, TK\_WRITE, TK\_WHILE, TK\_SQL, TK\_CALL }
4. first(<assignmentStmt>) ={ TK\_ID, TK\_RECORDID }
5. a. first(<singleOrRecId>) = {TK\_ID}

b. first(<singleOrRecId’>) = {TK\_DOT, eps}

1. first(<funCallStmt>) = { TK\_SQL, TK\_CALL }
2. first(<outputParameters>) = { TK\_SQL, eps }
3. first(<inputParameters>) = { TK\_SQL }
4. first(<iterativeStmt>) = { TK\_WHILE }
5. first(<conditionalStmt>) = { TK\_IF }
6. first(<conditionalStmt’>) = { TK\_ELSE, TK\_ENDIF }
7. first(<ioStmt>) = { TK\_READ, TK\_WRITE }
8. **\*Deleted\***
9. a. first(<arithmeticExpression>)= {TK\_OP, TK\_ID, TK\_NUM, TK\_RNUM}

b. first(<arithmeticExpression’>)= {TK\_PLUS, TK\_MINUS, eps}

c. first(<term>)= {TK\_OP, TK\_ID, TK\_NUM, TK\_RNUM}

d. first(<term’>)= {TK\_MUL, TK\_DIV, eps}

1. first(<factor>)= {TK\_OP, TK\_ID, TK\_NUM, TK\_RNUM}
2. a. first(<add\_sub\_operator>)= { TK\_PLUS , TK\_MINUS }

b. first(<mul\_div\_operator>)= { TK\_MUL, TK\_DIV }

1. first(<booleanExpression>) = { TK\_OP, TK\_ID, TK\_NUM, TK\_RNUM, TK\_NOT }
2. first(<var>) = { TK\_ID, TK\_NUM, TK\_RNUM }
3. first(<logicalOp>) = { TK\_AND, TK\_OR }
4. first(<relationalOp>) = { TK\_LT, TK\_LE, TK\_EQ, TK\_GT, TK\_GE, TK\_NE }
5. first(<returnStmt>) = { TK\_RETURN }
6. first(<optionalReturn>) = { TK\_SQL, eps }
7. first(<idList>) = { TK\_ID }
8. first(<more\_ids>) = { TK\_COMMA, eps }

FOLLOW Sets:

1. follow(<program>)= {$}
2. follow(<mainFunction>)= {$}
3. follow(<otherFunctions>)= { TK\_MAIN }
4. follow(<function>)= { TK\_FUNID, TK\_MAIN }
5. follow(<input\_par>)= { TK\_OUTPUT, TK\_SEM }
6. follow(<output\_par>)= { TK\_SEM }
7. follow(<parameter\_list>)= { TK\_SQR }
8. follow(<dataType>)= { TK\_COLON, TK\_ID }
9. follow(<primitiveDatatype>)= { TK\_COLON, TK\_ID }
10. follow(<constructedDatatype>)= { TK\_COLON, TK\_ID }
11. follow(<remaining\_list>)= { TK\_SQR }
12. follow(<stmts>)= { TK\_END }
13. follow(<typeDefinitions>)= { TK\_ID, TK\_TYPE, TK\_RECORD\_ID, TK\_WHILE, TK\_IF, TK\_END, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN }
14. follow( <typeDefinition>)= { TK\_RECORD, TK\_TYPE, TK\_ID, TK\_WHILE, TK\_IF, TK\_END, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN }
15. follow(<fieldDefinitions>)= { TK-ENDRECORD }
16. follow(<fieldDefinition>)= { TK\_TYPE, TK\_ENDRECORD }
17. follow(<moreFields>)= { TK\_ENDRECORD }
18. follow(<declarations>) = {TK\_ID, TK\_RECORDID, TK\_IF, TK\_READ, TK\_WRITE, TK\_WHILE, TK\_SQL, TK\_CALL, TK\_RETURN}
19. a. follow(<declaration>) = {TK\_ID, TK\_RECORDID, TK\_IF, TK\_READ, TK\_WRITE, TK\_WHILE, TK\_SQL, TK\_CALL, TK\_TYPE}

b. follow(<declaration’>) = {TK\_SEM}

1. follow(<global\_or\_not>)= { TK\_SEM }
2. follow(<otherStmts>) = { TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
3. follow(<stmt>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
4. follow(<assignmentStmt>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
5. a. follow(<singleOrRecId>) = { TK\_ASSIGNOP }

b. follow(<singleOrRecId’>) = { TK\_ASSIGNOP }

1. follow(<funCallStmt>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
2. follow(<outputParameters>) = { TK\_CALL }
3. follow(<inputParameters>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
4. follow(<iterativeStmt>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
5. follow(<conditionalStmt>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
6. follow(<conditionalStmt’>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
7. follow(<ioStmt>) = { TK\_ID, TK\_RECORDID, TK\_WHILE, TK\_IF, TK\_READ, TK\_WRITE, TK\_SQL, TK\_CALL, TK\_RETURN, TK\_ENDWHILE, TK\_ELSE, TK\_ENDIF }
8. **\*Deleted\***
9. a. follow(<arithmeticExpression>)= {TK\_CL, TK\_SEM}

b. follow(<arithemeticExpression’>)= {TK\_CL, TK\_SEM}

c. follow(<term>)= {TK\_PLUS , TK\_MINUS, TK\_CL, TK\_SEM}

d. follow(<term’>)= {TK\_PLUS , TK\_MINUS, TK\_CL, TK\_SEM}

1. follow(<factor>)= {TK\_MUL, TK\_DIV, TK\_PLUS , TK\_MINUS, TK\_CL, TK\_SEM}
2. a. follow(<add\_sub\_operator>)= {TK\_OP, TK\_ID, TK\_NUM, TK\_RNUM}

b. follow(<mul\_div\_operator>)= {TK\_OP, TK\_ID, TK\_NUM, TK\_RNUM}

1. follow(<booleanExpression>) = { TK\_CL, TK\_THEN }
2. follow(<var>) = {TK\_CL, TK\_MUL, TK\_DIV, TK\_PLUS , TK\_MINUS, TK\_SEM, TK\_LT, TK\_LE, TK\_EQ, TK\_GT, TK\_GE, TK\_NE, TK\_THEN }
3. follow(<logicalOp>) = {TK\_OP}
4. follow(<relationalOp>) = {TK\_ID, TK\_NUM, TK\_RNUM}
5. follow(<returnStmt>) = {TK\_END}
6. follow(<optionalReturn>) = {TK\_SEM}
7. follow(<idList>) = {TK\_SQR}
8. follow(<more\_ids>) = {TK\_SQR}