Parser Grammar for Java while and if-else contructs

Team Members :

Mehul Garg 01FB16ECS206

Naman Bansal 01FB16ECS220

P Abhishikta Sai 01FB16ECS242

*T\_ABC : This indicates the symbol is non-terminal.*

*T\_ABC? : This indicates the non-terminal can be present zero or more number of times*

**Names**

T\_Name > -> T\_SimpleName | T\_QualifiedName

T\_SimpleName -> Id

T\_QualifiedName -> T\_Name . Id

T\_Literal -> IntLit | FloatLit | BoolLit | CharLit | StringLit | NullLit

**Packages**

S -> T\_PackageDecl? T\_ImportDecl? T\_TypeDecl

T\_PackageDecl -> package T\_Name ;

T\_ImportDecl -> import T\_Name; | import T\_name . \*

T\_TypeDecl -> ; | T\_Class

**Types**

T\_Type -> T\_PrimType

T\_PrimType -> T\_NumType > | boolean

T\_NumType > -> byte | short | int | long | char | float | double

**Modifiers**

T\_Modifiers -> T\_Modifier | T\_Modifiers T\_Modifier

T\_Modifier > -> public | private | protected | static | abstract | final | native

**Clases & Methods**

T\_ClassDecl -> T\_Modifiers class Id T\_ClassBody

T\_ClassBody -> {T\_Main T\_ ClassBodyDeclList?}

T\_Main -> T\_Modifiers? T\_Type main (T\_FormalParmList?) T\_Block | T\_Modifiers? void main (T\_FormalParmList?) T\_Block

T\_ ClassBodyDeclList -> T\_MethodDecl | static T\_Block

T\_MethodDecl -> T\_MethodHdr T\_Block

T\_MethodHdr -> T\_Modifiers? T\_Type Id (T\_FormalParmList?) | T\_Modifiers? void Id (T\_FormalParmList?) T\_Block

T\_FormalParmList > -> T\_FormalParam | T\_FormalParmList , T\_FormalParam

T\_FormalParam ->T\_Type T\_VarDeclId

T\_VarDeclId -> Id | T\_VarDeclId [ ]

**Blocks**

T\_Block -> {T\_BlockStmt | T\_BlockStmt T\_BlockStmt?}

T\_BlockStmt -> T\_LocalVar | T\_Stmt

T\_LocalVar -> T\_Type T\_VarDeclId ; | T\_Type T\_VarDeclId , T\_VarDeclId? ;

T\_Stmt -> T\_StmtOther | T\_If | T\_IfElse | T\_While

T\_StmtOther -> T\_Block | T\_ExprStmt | T\_BreakStmt | T\_RetStmt | ;

T\_ExprStmt -> T\_Assign | T\_DecExpr | T\_IncExpr

**T\_If > -> if ( T\_Expr ) T\_Stmt**

**T\_IfElse -> if ( T\_Expr ) T\_Stmt else T\_Stmt**

**T\_While -> while ( T\_Expr ) T\_Stmt**

T\_BreakStmt -> break Id? ;

T\_RetStmt > -> return T\_Expr? ;

**Expressions**

T\_Expr -> T\_Assign | T\_CondExpr | T\_DecExpr | T\_IncExpr | T\_Primary

T\_Assign -> T\_Name T\_AssignOp T\_CondExpr | T\_Name T\_AssignOp T\_Primary

T\_CondExpr -> T\_RelatExpr | T\_ArithExpr | T\_OtherExpr

T\_Relat -> T\_Expr T\_Relop T\_Expr

T\_ArithExpr -> T\_Expr T\_Arithop T\_Expr

T\_OtherExpr -> T\_Expr T\_Otherop T\_Expr

T\_DecExpr -> --T\_Expr | T\_Expr—

T\_IncExpr -> ++T\_Expr | T\_Expr++

T\_Primary -> T\_Literal | Id | (T\_Expr)

**Operators**

T\_AssignOp -> = | \*= | /= | %= | += | -=

T\_Relop -> == | < | > | != | <= | >=

T\_Arithop -> + | - | \* | / | %

T\_Otherop -> && | || | & | ^