CD

LAB PROJECT

ANUBHAV TIKKU RN:59

DIVYANSH VERMA RN:49

SAURAV PANDA RN:37

**Language:-Decaf**

**Grammar**

Program ::= Decl+

Decl ::= VariableDecl | FunctionDecl | ClassDecl | InterfaceDecl

VariableDecl ::= Variable ;

Variable ::= Type ident

Type ::= int | double | bool | string | ident | Type [ ]

FunctionDecl ::= Type ident ( Formals ) StmtBlock |

void ident ( Formals ) StmtBlock

Formals ::= Variable+, | ǫ

ClassDecl::= class ident < extends ident> <implements ident+,> { Field∗ }

Field ::= VariableDecl | FunctionDecl

InterfaceDecl ::= interface ident { Prototype∗ }

Prototype ::= Type ident ( Formals ) ; | void ident ( Formals ) ;

StmtBlock ::= { VariableDecl∗ Stmt∗ }

Stmt ::= <Expr>; | IfStmt | WhileStmt | ForStmt | BreakStmt | ReturnStmt | PrintStmt | StmtBlock

IfStmt ::= if ( Expr ) Stmt <else Stmt>

WhileStmt ::= while ( Expr ) Stmt

ForStmt ::= for ( <Expr>; Expr ; <Expr>) Stmt

ReturnStmt ::= return < Expr > ;

BreakStmt ::= break ;

PrintStmt ::= Print ( Expr+, ) ;

Expr ::= LValue = Expr | Constant | LValue | this | Call | (Expr) |

Expr + Expr | Expr - Expr | Expr \* Expr | Expr / Expr |

Expr % Expr | - Expr | Expr < Expr | Expr <= Expr |

Expr > Expr | Expr >= Expr | Expr == Expr | Expr ! = Expr |

Expr && Expr | Expr || Expr | ! Expr | ReadInteger ( ) |

ReadLine ( ) | New ( ident ) | NewArray ( Expr , Type )

LValue ::= ident | Expr . ident | Expr [ Expr ]

Call ::= ident ( Actuals ) | Expr . ident ( Actuals )

Actuals ::= Expr+, | ǫ

Constant ::= intConstant | doubleConstant | boolConstant |

stringConstant | null