**Bold: Kind of Tested**

**Rules implemented: 1, 2,** 4, 7, **18,** **19, 20, 21, 23, 17, 16, 15**

**Rules left:** 3(partial)

1. Program ∷= Name List<ParamDec> Block

        class Name implements Runnable{

*variables declared in List<ParamDec> are instance variables of the class*

             public Name(String[] args){

                initialize instance variables with values from args.

                 }

             public static void main(String[] args){

                Name instance = new Name(args);

                        instance.run();

                 }

                 public void run(){

                declarations and statements from block

                 }

           }

1. ParamDec ∷= type ident

        instance variable in class, initialized with values from arg array

1. Block ∷= List<Dec>  List<Statement>

Decs are local variables in current scope of run method

Statements are executed in run method

Must label beginning and end of scope, and keep track of local variables, their slot in the local variable array, and their range of visibility.

1. Dec ∷= type ident

Assign a slot in the local variable array to this variable and save it in the new slot attribute in the  Dec class.

1. Statement ∷= ~~SleepStatement~~ | WhileStatement | IfStatement | ~~Chain~~ | AssignmentStatement
2. ~~SleepStatement ∷= Expression~~
3. AssignmentStatement ∷= IdentLValue Expression

        store value of Expression into location indicated by IdentLValue

IMPORTANT:

            insert the following statement into your code for an Assignment Statement

        after value of expression is put on top of stack and before it is written into the

            IdentLValue

**CodeGenUtils.*genPrintTOS*(GRADE, mv,assignStatement.getE().getType());**

1. ~~Chain ∷= ChainElem | BinaryChain~~
2. ~~ChainElem ::= IdentChain | FilterOpChain | FrameOpChain | ImageOpChain~~
3. ~~IdentChain ∷= ident~~
4. ~~FilterOpChain ∷= filterOp Tuple~~
5. ~~FrameOpChain ∷= frameOp Tuple~~
6. ~~ImageOpChain ∷= imageOp Tuple~~
7. ~~BinaryChain ∷= Chain (arrow | bararrow)  ChainElem~~
8. WhileStatement ∷= Expression Block

            goto GUARD

    BODY     Block

    GUARD  Expression

            IFNE  BODY

1. IfStatement ∷= Expression Block

             Expression

            IFEQ AFTER

        Block

        AFTER …

1. Expression ∷=   IdentExpression | IntLitExpression | BooleanLitExpression | ~~ConstantExpression~~ | BinaryExpression

**always generate code to leave value of expression on top of stack.**

1. IdentExpression ∷= ident

       load value of variable (**this could be a field or a local var**)

1. IdentLValue ∷= ident

              store value on top of stack to this variable (**which could be a field or local var**)

1. IntLitExpression ∷= intLit

       load constant

1. BooleanLitExpression ∷= booleanLiteral

       load constant

1. ~~ConstantExpression ∷= screenWidth | screenHeight~~
2. BinaryExpression ∷= Expression op Expression

      Visit children to generate code to leave values of arguments on stack

      perform operation, leaving result on top of the stack.  Expressions should

      be evaluated from left to write consistent with the structure of the AST.

1. ~~Tuple :≔ List<Expression>~~
2. op ∷= relOp | weakOp | strongOp
3. type ∷= integer | ~~image | frame | file~~ | boolean | ~~url~~