

Program → void main () { *Declarations Statements* }
Declarations → ε | *Declarations Declaration*
Declaration → *Type Identifiers* ;
Type → int | boolean
Identifiers → *Identifier* | *Identifiers, Identifier*
Statements → ε | *Statements Statement*
Statement → ; | *Block* | *Assignment* | *IfStatement* | *WhileStatement*
Block → { *Statements* }
Assignment → *Identifier = Expression* ;
IfStatement → if (*Expression*) *Statement* |
if (*Expression*) *Statement* else *Statement*
WhileStatement → while (*Expression*) *Statement*
Expression → *Conjunction* | *Expression* || *Conjunction*
Conjunction → *Relation* |
Conjunction && *Relation*
Relation → *Addition* |
Relation < *Addition* |
Relation <= *Addition* |
Relation > *Addition* |
Relation >= *Addition* |
Relation == *Addition* |
Relation != *Addition*
Addition → *Term* |
Addition + *Term* |
Addition - *Term*
Term → *Negation* |
Term * *Negation* |
Term / *Negation*
Negation → *Factor* | ! *Factor*
Factor → *Identifier* | *Literal* | (*Expression*)

Program → void main () '{ Declarations Statements }'
Declarations → { Declaration }*
Declaration → Type Identifier;
Type → int | boolean
Identifiers → Identifier { , Identifier }*
Statements → { Statement }*
Statement → ; | Block | Assignment | IfStatement | WhileStatement
Block → '{ Statements }'
Assignment → Identifier = Expression ;
IfStatement → if (Expression) Statement { else Statement }_{opt}
WhileStatement → while (Expression) Statement
Expression → Conjunction { || Conjunction }*
Conjunction → Relation { && Relation }*
Relation → Addition [| < | <= | > | >= | == | !=] Addition
Addition → Term { [+ | -] Term }*
Term → Negation [* | /] Negation
Negation → { ! }_{opt} Factor
Factor → Identifier | Literal | (Expression)