

Grammar:

```
1 Program      -> ProgramItems
2 ProgramItems -> ProgramItem ProgramItems
3 ProgramItem  -> Statement | Expression | Declaration
```

Declaration

```
1 Declaration -> FunctionDeclaration
2             -> VariableDeclaration
3 FunctionDeclaration -> 'function' Identifier '(' Arguments ')' ':' Type
   BlockStatement
4 Arguments          -> Identifier [,Identifier]
5 Type               -> 'number' | 'void'
6 VariableDeclaration -> 'var' Identifier ('=' ConditionalExpression)?
```

Statement

```
1 Statement -> WhileStatement
2           -> IfStatement
3           -> BlockStatement
4           -> ReturnStatement
5 WhileStatement -> 'while' '(' Expression ')' BlockStatement
6 ReturnStatement -> 'return' Expression
7 BlockStatement -> '{' [ProgramItem] '}'
```

Expression:

```
1 Expression      -> AssignmentExpression
2                 -> AssignmentExpression [, AssignmentExpression]
3 AssignmentExpression -> ConditionalExpression
4                 -> ConditionalExpression '=' ConditionalExpression
5 ConditionalExpression -> BinaryExpression
6                 -> BinaryExpression '?' BinaryExpression ':' BinaryExpression
7 BinaryExpression    -> UnaryExpression
8                 -> UnaryExpression BinaryOps UnaryExpression
9 UnaryExpression     -> PrimaryExpression
10                  -> UnaryOps PrimaryExpression
11 PrimaryExpression   -> Identifier | Literal | CoverExpression | CallExpression
12 CoverExpression     -> '(' Expression ')'
13 CallExpression      -> Identifier '(' Parameters ')'
14 Parameters          -> ConditionalExpression [,ConditionalExpression]
```