Our Project Grammar

```
Program
                     ::== Block $
                                                          Curly braces
Block
                     ::== { StatementList
                                                          denote new scope.
StatementList
                     ::== Statement StatementList
                     ::== E
Statement
                     ::== PrintStatement
                     ::== AssignmentStatement
                     ::== VarDecl
                     ::== WhileStatement
                     ::== IfStatement
                     ::== Block
PrintStatement
                     ::== print ( Expr )
                                                            = is assignment.
AssignmentStatement ::== Id = Expr
VarDecl
                     ::== type Id
WhileStatement
                     ::== while BooleanExpr Block
                     ::== if BooleanExpr Block
IfStatement
Expr
                     ::== IntExpr
                     ::== StringExpr
                     ::== BooleanExpr
                     ::== Id
IntExpr
                     ::== digit intop Expr
                     ::== digit
StringExpr
                     ::== " CharList "
                     ::== ( Expr boolop Expr )
BooleanExpr
                     ::== boolval
Id
                     ::== char
CharList
                     ::== char CharList
                     ::== space CharList
                     ::== E
                     ::== int | string | boolean
type
char
                     ::== a |
                              b | c ... z
                     ::== the space character
space
                     ::== 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
digit
boolop
                     ::== == | !=
                                                       == is test for equality.
boolval
                     ::== false | true
intop
                     ::== +
```

Comments are bounded by /* and */ and ignored by the lexer.