Grammar:

```
// List of Colons

semi_colon_list
   :  ';'  semi_colon_list_tail

semi_colon_list_tail
   : semi_colon_list
   | ε

---

// Assignment Operators

assignment_operator //Code Gen = first. Add more later
   : '='
   | MUL_ASSIGN
   | DIV_ASSIGN
   | MOD_ASSIGN
   | ADD_ASSIGN
   | SUB_ASSIGN
   | LEFT_ASSIGN
   | RIGHT_ASSIGN
   | AND_ASSIGN
   | XOR_ASSIGN
   | OR_ASSIGN

---

//Data Types

typed_ID
   : type_specifier ID

type
   : CHAR
   | SHORT
   | INT
   | LONG
   | FLOAT
   | DOUBLE

type_specifier //TODO Const, Volatile later
   : type
   | VOID
   | SIGNED type
   | UNSIGNED type
```

```
47
48  ---
49
50  //Type List
51
52  type_specifier_list
53     :  type_specifier type_specifier_list_tail
54
55  type_specifier_list_tail
56     :  ε
57     | ',' type_specifier_list
58
59  ---
60
61  //Parameter List
62
63  parameter_specifier_list
64     :  type_specifier ID parameter_specifer_tail
65
66  parameter_specifer_list_tail
67     : ε
68     | ','  parameter_specifier_list
69
70  ---
71
72  //Program
73  program
74     :  body EOF
75
76  ---
77
78  //Body
79
80  body_statement
81     : statement body_statement_tail
82
83  body_statement_tail
84     : ε
85     : body
86
87  body_direct_declaration
88     : direct_declaration body_direct_declaration_tail
89
90  body_direct_declaration_tail
91     : ε
92     : body
93
```

```
94  body_function_declaration
95     : function_declaration body_function_declaration_tail
96
97  body_function_declaration_tail
98     : ε
99     : body
100
101 body_function_prototype
102    : function_declaration body_function_prototype_tail
103
104 body_function_prototype_tail
105    : ε
106    : body
107
108 body
109    : body_statement
110    | body_direct_declaration
111    | body_function_declaration
112    | body_function_prototype
113
114 ---
115
116 //Function Declaration
117
118 function_prefix
119    :  typed_ID '('
120
121 function_prototype
122    :  function_prefix type_specifier_list ')' semi_colon_list
123    |  function_declaration_prefix semi_colon_list
124
125 function_declaration
126    :  function_prefix parameter_specifier_list ')' '{' statement '}'
127
128 function_declaration_tail
129    |  semi_colon_list
130    |  ε
131
132 ---
133
134 //Direct Declaration
135
136 direct_declaration
137    : type ID semi_colon_list
138    | SIGNED type ID semi_colon_list
139    | UNSIGNED type ID semi_colon_list
140    | VOID ID semi_colon_list
```

```
141    | CHAR ID '=' STRING_LITERAL semi_colon_list
142
143 ---
144
145 //Statement and Statement List
146
147 statement
148    : compound_statement //Sub statements to loops, conditional, and code blocks
149    | expression_statement //Assignment, Boolean, Arithmetic Expressions
150    | selection_statement //IF Statements
151    | iteration_statement  //Loops
152    | semi_colon_list // End of statement one or more ;
153    | direct_declaration
154
155 statement_list
156    : statement statement_list_tail
157
158 statement_list_tail
159    : statment_list
160    | ε
161
162
163
164 compound_statement
165    : '{' '}'
166    | '{' statement_list '}'
167
168
169
170 expression_statement
171    : ';'
172    | expression ';'
173
174 //TODO Add Expression
175 expression
176
177 selection_statement
178    : IF '(' expression ')' statement
179    | IF '(' expression ')' statement ELSE statement
180
181
182
183 iteration_statement
184    : WHILE '(' expression ')' statement
185    | FOR '(' expression_statement expression_statement ')' statement
186    | FOR '(' expression_statement expression_statement expression ')' statement
```

$$program \Rightarrow body \;\; EOF \tag{1}$$

$$\begin{aligned} body \Rightarrow\; & stmt \\ & \mid stmt \;\; body \end{aligned} \tag{2}$$

$$\begin{aligned} stmt \Rightarrow\; & ID \;\; '=' \;\; expr \;\; ';' \\ & \mid IF \;\; '(' \;\; bexp \;\; ')' \;\; stmt \\ & \mid WHILE \;\; '(' \;\; bexp \;\; ')' \;\; stmt \\ & \mid '\{' \;\; substmts \\ & \mid ';' \end{aligned} \tag{3}$$

$$\begin{aligned} substmts \Rightarrow\; & stmt \;\; '\}' \\ & \mid stmt \;\; substmts \\ & \mid '\}' \end{aligned} \tag{4}$$

$$assignment \Rightarrow\; ID \;\; '=' \;\; expression \;\; ';' \tag{5}$$

$$\begin{aligned} expression \Rightarrow\; & expr \\ & \mid bexpr \end{aligned} \tag{6}$$

Notes - Continue down this route
TODO

- expr - arithmetic expressions. Make sure to get precedence (greater precedence last) and associativity correct. From `http://pages.cs.wisc.edu/~fischer/cs536.s08/course.hold/html/NOTES/3.CFG.html#assoc` Remove POW

```
exp      --> exp PLUS term        |  exp MINUS term      |  term
term     --> term TIMES factor    |  term DIVIDE factor  |  factor
```

```
factor   --> exponent POW factor  |  exponent
exponent --> MINUS exponent       |  final
final    --> INTLITERAL           |  LPAREN exp RPAREN
```

- bexpr - boolean expressions

```
bexp --> TRUE
bexp --> FALSE
bexp --> bexp OR bexp
bexp --> bexp AND bexp
bexp --> NOT bexp
bexp --> LPAREN bexp RPAREN
```

- stmt - add if and while loops to it. Need to make sure no ambiguity in control statements.