

(Enhanced) Grammar for Basic SPLAT 1.3

```
<program> ::= program <decls> begin <stmts> end ;

<decls> ::= ( <decl> )*

<decl> ::= <var-decl>
        | <func-decl>

<var-decl> ::= <label> : <type> ;

<func-decl> ::= <label> ( <params> ) : <ret-type> is <loc-var-decls> begin <stmts> end ;

<params> ::= <param> ( , <param> )*
        | ε

<param> ::= <label> : <type>

<loc-var-decls> ::= ( <var-decl> )*

<stmts> ::= ( <stmt> )*

<stmt> ::= <label> := <expr> ;
        | while <expr> do <stmts> end while ;
        | if <expr> then <stmts> else <stmts> end if ;
        | if <expr> then <stmts> end if ;
        | <label> ( <args> ) ;
        | print <expr> ;
        | print_line ;
        | return <expr> ;
        | return ;
```

$\langle \text{expr} \rangle ::=$
 $(\langle \text{expr} \rangle \langle \text{bin-op} \rangle \langle \text{expr} \rangle)$
 $|$
 $(\langle \text{unary-op} \rangle \langle \text{expr} \rangle)$
 $|$
 $\langle \text{label} \rangle (\langle \text{args} \rangle)$
 $|$
 $\langle \text{label} \rangle$
 $|$
 $\langle \text{literal} \rangle$

$\langle \text{bin-op} \rangle ::=$
 $\text{and} \mid \text{or} \mid > \mid < \mid == \mid >= \mid <= \mid + \mid - \mid * \mid / \mid \%$

$\langle \text{unary-op} \rangle ::=$
 $\text{not} \mid -$

$\langle \text{args} \rangle ::=$
 $\langle \text{expr} \rangle (, \langle \text{expr} \rangle)^*$
 $|$
 ϵ

$\langle \text{label} \rangle ::=$
...sequence of alphanumeric characters and underscore, not starting with a digit, which are not keywords...

$\langle \text{ret-type} \rangle ::=$
 $\langle \text{type} \rangle \mid \text{void}$

$\langle \text{type} \rangle ::=$
 $\text{Integer} \mid \text{Boolean} \mid \text{String}$

$\langle \text{literal} \rangle ::=$
 $\langle \text{int-literal} \rangle \mid \langle \text{bool-literal} \rangle \mid \langle \text{string-literal} \rangle$

$\langle \text{int-literal} \rangle ::=$
...sequence of decimal digits...

$\langle \text{bool-literal} \rangle ::=$
 $\text{true} \mid \text{false}$

$\langle \text{string-literal} \rangle ::=$
"...sequence of characters and space that do not contain double-quotes, backslashes, or newlines..."