

Project 2 Grammar

- Variable Declaration
 - $\langle \text{var_dec} \rangle ::= \langle \text{var_type} \rangle \langle \text{var_name} \rangle$
 - $\langle \text{var_type} \rangle ::= \text{int} | \text{str} | \text{bool}$
 - $\langle \text{var_name} \rangle ::= \langle \text{str} \rangle$
- Variable Assignment
 - $\langle \text{var} \rangle ::= \langle \text{var_dec} \rangle = \langle \text{var_assignment} \rangle$
 - $\langle \text{var_assignment} \rangle ::= \langle \text{var} \rangle | \langle \text{int_expr} \rangle | \langle \text{str_expr} \rangle | \langle \text{bool_expr} \rangle$
- Letters, Characters and Strings
 - $\langle \text{str_root} \rangle ::= \langle \text{char} \rangle | \langle \text{char} \rangle \langle \text{str_root} \rangle$
 - $\langle \text{char} \rangle ::= \langle \text{symbol} \rangle | \langle \text{letter} \rangle | \langle \text{digit} \rangle$
 - $\langle \text{symbol} \rangle ::= ! | " | \# | \$ | \% | \& | ' | (|) | * | + | , | - | . | / | : | ; | < | > | ? | @ | [| \backslash |] | ^ | _ | ` | { | } | ~$
 - $\langle \text{letter} \rangle ::= \text{a} | \text{b} | \text{c} | \text{d} | \text{e} | \text{f} | \text{g} | \text{h} | \text{i} | \text{j} | \text{k} | \text{l} | \text{m} | \text{n} | \text{o} | \text{p} | \text{q} | \text{r} | \text{s} | \text{t} | \text{u} | \text{v} | \text{x} | \text{y} | \text{z} | \text{A} | \text{B} | \text{C} | \text{D} | \text{E} | \text{F} | \text{G} | \text{H} | \text{I} | \text{J} | \text{K} | \text{L} | \text{M} | \text{N} | \text{O} | \text{P} | \text{Q} | \text{R} | \text{S} | \text{T} | \text{U} | \text{V} | \text{W} | \text{X} | \text{Y} | \text{Z}$

String Expressions

 - $\langle \text{str_expr} \rangle ::= \langle \text{str_root} \rangle + \langle \text{str_root} \rangle | \langle \text{str_root} \rangle^* \langle \text{int_root} \rangle$
- Integers
 - $\langle \text{int_root} \rangle ::= \langle \text{int} \rangle | - \langle \text{int} \rangle$
 - $\langle \text{int} \rangle ::= \langle \text{digit} \rangle | \langle \text{digit} \rangle \langle \text{int_root} \rangle$
 - $\langle \text{digit} \rangle ::= 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9$

Integer Expressions

 - $\langle \text{int_expr} \rangle ::= \langle \text{int_root} \rangle \langle \text{op} \rangle \langle \text{int_root} \rangle | \langle \text{int_root} \rangle \langle \text{op} \rangle (\langle \text{int_expr} \rangle)$
 - $\langle \text{op} \rangle ::= + | - | * | / | \% | < | > | = | < = | > = | ! =$
- Booleans
 - $\langle \text{boolean_root} \rangle ::= \langle \text{signed_bool} \rangle | \langle \text{bool_expr} \rangle$
 - $\langle \text{signed_bool} \rangle ::= \langle \text{bool} \rangle | ! \langle \text{bool} \rangle$
 - $\langle \text{bool} \rangle ::= \text{true} | \text{false}$

Boolean Expressions

 - $\langle \text{bool_expr} \rangle ::= \langle \text{signed_bool} \rangle \langle \text{comp} \rangle \langle \text{signed_bool} \rangle | \langle \text{signed_bool} \rangle \langle \text{comp} \rangle (\langle \text{bool_expr} \rangle)$
 - $\langle \text{comp} \rangle ::= \& \& | || | < | > | = | < = | > = | ! =$

- Conditionals
 - `<if_stmt> ::= <if>|<if><elseif>|<if><elseif><else>`
 - `<if> ::= if (<bool_expr>)`
`{ <var_assignment>|<var_update>|<if_stmt> }`
 - `<elseif> ::= elseif (<bool_expr>) { <var_assignment> |`
`<var_update> | <if_stmt> } | <elseif><elseif>`
 - `<else> ::= else { <var_assignment>|<var_update>|<if_stmt> }`
 - `<var_update> ::= <var_name> = <var_name> <update_op>`
`<expr>`
 - `<expr> ::= <int_expr>|<str_expr>|<bool_expr>`
- Loops
 - `<while_loop> ::= while (<bool_expr>) { <var_assignment> |`
`<var_update> }`
 - `<for_loop> ::= for (<var_assignment>; <bool_expr>;`
`<var_update>) { <var_assignment>|<var_update> }`
- Command-Line Arguments
 - `<cmd_args> ::= cmd[<int_root>]`