CSC 372 Project 2 Grammar

integers

<digit> ::= 0 | 1| 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
<integer> ::= <digit> | <integer> <digit>

variables

<letter> ::= A | B | C | D | E | F | G | H | I | J | K | L | M | N | O | P | Q | R | S | T | U | V | W | X | Y | Z | a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z
<variable> ::= <letter> | <variable> <letter>

variable assignment

<var_asgmt> ::= integer <variable> assign <int_expr> | boolean <variable> assign <bool_expr> | <variable> assign <expr>
<expr> ::= <int_expr> | <bool_expr>

integer expression

<int_expr> ::= <mult_expr> | <int_expr> sub <mult_expr> | <int_expr> add <mult_expr>
<mult_expr> ::= <negate_expr> | <mult_expr> mult <negate_expr> | <mult_expr> div <negate_expr> | <mult_expr> mod <negate_expr>
<negate_expr> ::= <int_root> | negate <int_root>
<int_root> ::= <integer> | <variable> | <cmd_arg_var>

booleans

<boolean> ::= true | false

boolean expression

<bool_expr> ::= <and_expr> | <bool_expr> or <and_expr>
<and_expr> ::= <not_expr> | <and_expr> and <not_expr>
<not_expr> ::= <bool_root> | not <bool_root> | <int_expr> <comparator> <int_expr>
<comparator> ::= lt | gt | lte | gte | equal | diff
<bool_root> ::= <boolean> | <variable>

conditionals

<conditional> ::= if <bool_expr>
<mult_cond_loop>
end |
if <bool_expr>
<mult_cond_loop>
else
<mult_cond_loop>
end

<mult_cond_loop> ::= <root_cond_loop> |
<mul_cond_loop>
<root_cond_loop>

<root_cond_loop> ::= <var_asgmt> | <print> | <loop> | <conditional>

loops

<loop> ::= while <bool_expr>
<mult_cond_loop>
end

printing to output

<print> ::= printout <print_root> | print <print_root>
<print_root> ::= <int_expr> | <bool_expr> | <string_literal>

command line arguments

<cmd_arg_define> ::= use <integer> cmd args
<cmd_arg_var> ::= arg<integer>

string literals

<string_literal> ::= “<string>”
<string> ::= <character> | <string><character>
<character> ::= Any ASCII character

comments

<comment> ::= #<comment_line>
<comment_line> ::= <comment_char> | <comment_line> <comment_char>
<comment_char> ::= Any ASCII character except for double quote character