COMPILER DESIGN

Grammar

BNF Grammar

This file contains the BNF grammar for a custom mini-language with support for integers, floats, strings, arithmetic and logical operators, assignment, conditionals and loops

➢ Program Structure:

- <program> ::= <stmt_list>

- <stmt_list> ::= <stmt>
  | <stmt> <stmt_list>

➢ Statements:

- <stmt> ::= <var_decl>
  | <assignment>
  | <if_stmt>
  | <return_stmt>
  | <while_stmt>
  | <comment>
- <stmt_block> ::= “{“ <stmt_list> “}”

➢ Variable Declarations and Assignments:

- <var_decl> ::= <var_type> <identifier> "=" <expr> ";"
- <var_type> ::= "integer"
  | "float"
  | "string"
  | "bool"
- <assignment> ::= <identifier> "=" <expr> " “;”

➢ Control Flow Statements

- <if_stmt> ::= "if" "(" <expr> ")" <stmt_list>
- <for_stmt> ::= "while" "(" <expr")" <stmt_block>

➢ Expressions

- <expr>::= "integer"
  | "float"
  | "string"
  | "bool"
  | <expr> <arith_op> <expr>
  | <expr> <rel_op> <expr>
  | <expr> <logical_op> <expr>
  | "(" <expr> ")"
- <integer> ::= [0-9]+
- <var> ::= <identifier>
- <identifier> ::= [a-zA-Z_][a-zA-Z_0-9]*
- <float> ::= [0-9]+"."<integer>
- <bool> ::= "true" | "false"
- <string> ::= "\"" <string_content> "\""
- <string_content> ::= [a-zA-Z_0-9]* | <string_content> <string_content>
- <arith_op> ::= "+" | "-" | "*" | "/"
- <rel_op> ::= "==" | "!=" | "<" | ">" | "<=" | ">="
- <logical_op> ::= "&&" | "||"

➢ Return Statements

- <return_stmt> ::= "return" <expr> ";"

➢ Comments

- <comment> ::= "#" <string_content>

➢ Punctuations

<punctuation> ::= ";" | "(" | ")" | "{" | “}”