CSE4061– Compiler Design Project Assignment #1

Grammar Rules for Our Language:

stmts → stmt stmts | ε

stmt → decl_stmt | compound_stmt | if_stmt | for_stmt | loop_stmt | function_init_stmt | func_call_stmt

decl_stmt → just_decl_stmt | decl_w_init_stmt | assign_stmt

just_decl_stmt → variable_type **ID ;**

decl_w_init_stmt → variable_type assign_stmt

assign_stmt → **ID** assignment_operators arithm_simple_expr **;**

compound_stmt → **{** stmts **}**

if_stmt → **if (** bool_expr **)** stmt | **if (** bool_expr **)** stmt **else** stmt

for_stmt → **for (** for_decl **;** arithm_simple_expr **:** arithm_simple_expr **;** iter_spec **)** stmt

for_decl → for_variable_type **ID** | **ID**

for_variable_type → **int** | **double**

iter_spec → arithm_operators **(** arithm_expr **)** | **ID** unary_operators

loop_stmt → **loop (** bool_expr **)** stmt

func_init_stmt → **func ID (** params_list **)** compound_stmt

params_list → param_decl params_list_continue | ε

params_list_continue → **,** param_decl params_list_continue |ε

param_decl → variable_type **ID**

func_call_stmt → **ID (** param_values_list **) ;**

params_values_list → simple_expr params_value_list_continue | ε

params_value_list_continue → **,** simple_expr params_value_list_continue |ε

arithm_simple_expr → **ID** | **INT_VAR** | **DOUBLE_VAR** | arithm_expr | **(** arithm_expr **)**

arithm_expr → arithm_simple_expr arithm_operators arithm_expr |
arithm_simple_expr

simple_expr → arithm_simple_expr | **STRING_VAR**

bool_expr → simple_bool_expr bool_expr_continue

bool_expr_continue → logical_operators simple_bool_expr bool_expr_continue | ε

simple_bool_expr → arithm_expr relational_operators arithm_expr

assignment_operators → '=' | '+=' | '-=' | '*=' | '/='

logical_operators → '&&' | '||'

arithm_operators → '+' | '-' | '*' | '/'

relational_operators → '<' | '>' | '<=' | '>=' | '==' | '!='

unary_operators → '++' | ' --'

variable_type → **int** | **double** | **string**

Lexical Structure of Our Language:

Comments: Comments starts with # character and ends with the same (#) character.

Keywords: The keywords in grammar of our programming language :

- int, double, string, if, else, func, loop, for

Identifiers: An identifier starts with a letter and continues with a letter or digit or an underscore character. An identifier can not be the same with a keyword exactly but it can contain it. Length of an identifier should not exceed 64 characters.

Delimiters: whitespace, tab and new line supported by our language.

Operators:

- **Assignment operators:**
  - = | += | -=
- **Logical Operators:**
  - && | ||
- **Relational Operators:**
  - < | > | <= | >= | == | !=
- **Unary Operators:**
  - ++ | --

Numbers:

- **digit** : 0 - 9 → 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9
- **INT_VAR (int) :** digit+
- **DOUBLE_VAR (double) :** digit+ exponent | digit+ fraction (exponent | ε)
- **exponent :** e (+ | - | ε) digit+ (the maximum exponent value is 128)
- **fraction :** .digit+

Texts (strings):

- **STRING_VAR (string) :** Strings can contain any characters that are supported by UTF-8 format. It consists of quote characters (“) at the start and at the end.

Group Members :

İlker FENER 150115024
Cem GÜLEÇ 150117828
Büşra GÖKMEN 150116027