

Compilers DSL

Sasank - CS21BTECH11019

Avinash - CS21BTECH11039

Yagnesh - CS21BTECH11003

Asli - CS21BTECH11011

Contents

1	Lexical Conventions	2
1.1	Comments	2
1.2	White Space	2
1.3	Reserved Keywords	2
1.4	Special Symbols	2
1.5	Operators	2
1.6	Identifiers	3
2	Declarations:	3
2.1	Variable declaration:	3
2.2	Constants:	3
2.2.1	String constants:	3
2.2.2	Numerical constants:	3
2.3	Array declarations:	4
2.4	Function declarations :	4
3	Statements:	4
3.1	Non control flow Statements	4
3.2	Control flow statements	5
3.2.1	Conditional statement:	5
3.2.2	Loop statements:	5
3.2.3	Break statement:	5
3.2.4	Continue statement:	6
4	Domain	7
4.1	Encryption and Decryption	7
5	included libraries	7
5.1	String operations	7
5.2	Matrix Operations	7

1 Lexical Conventions

1.1 Comments

Single line Comments in this language begin with # symbol.

```
# This is a single line comment
int a = 10 .
```

Multi line Comments would begin with /* and end with */.

```
/* This is first line of multi comment
   This is the second line of multiline comment */
int arr[10] .
```

1.2 White Space

- White space includes tabs , spaces
- White spaces will be ignored except in places where spaces are required mainly for separation of tokens

1.3 Reserved Keywords

- Primitive Data Types
 - int , char , bool , void , float , double , struct
- Non Primitive Data Types
 - matrix , graph , set , vector
- Other Keywords
 - iterate , return , in , continue , break

1.4 Special Symbols

- [] { } () : . ,

1.5 Operators

- Assignment Operator
 - =
- Arithmetic Operators
 - add , sub , mul , div
- Conditional Operators

- `geq` , `gt` , `leq` , `lt` , `and` , `or` , `neg` , `neq` , `eq`

Conditional and arithmetic operators are also reserved keywords.

1.6 Identifiers

The identifiers in this language can contain either an alphabet or a digit. But the identifier must begin with an alphabet. It can not start with a digit.

Correct Syntax for identifiers

```
int a2 . # correct
string T2emP . # correct
```

Wrong Syntax for identifiers

```
int 2a . # wrong
string -T . # wrong
```

2 Declarations:

2.1 Variable declaration:

- Variable declaration must start with a datatype and can have multiple declarations in it.

```
int var = 100.
int a,b,c = 10.
float real = 37f6.
double var1 = 365d8979.
```

2.2 Constants:

2.2.1 String constants:

String constants must be encoded within double quotes(" ").

```
string str = "Hello World!".
```

2.2.2 Numerical constants:

- real numbers must be declared in floats and doubles datatypes.
- Float variables has 'f' in place of decimal '.'
- Double variables had 'd' in place of decimal '.'

```
float litres = 22f4.
double pi = 3d141.
```

2.3 Array declarations:

We could declare 1-D array or 2-D array in this language. The syntax is similar to C language.

2.4 Function declarations :

The function declaration begins with the return type of function then followed by name of the function. The arguments of the function are to be written in parenthesis. The function could have zero parameters as input.

Example :

```
# A function that has 2 input paramters and returns an integer.
int addition(int a,int b)
{
    int c = a+b .
    return c .
}
# A function that does not have any input parameters.
void print()
{
    printf("Hello World\n") .
}
```

3 Statements:

- Every statement that can be used in this language should be from one of the following statements.
 - Declaration statement
 - Expression statement
 - Call statement
 - Control flow statements
 - * Conditional statement
 - * Loop statement
 - Return statement
 - Assignment statement

3.1 Non control flow Statements

- Every statement except control flow statements should end with fullstop (.).
Syntax is as follows:

```
int var = 10.
int var1,var2,var3 = 10.
```

- Declaration statements are mentioned in part 2.
- Expression statements must contain '=' symbol in it and must terminate with full stop and both sides of equal to must contain one of variables or operations on variables.

- Call statement can be call to a function or call to the attributes of that object.
- Return statement must be used inside a function and the data type of the returned variable or method must match with function datatype.
- Variable initialization can be done at the declarations itself or it can be initialized after wards

3.2 Control flow statements

3.2.1 Conditional statement:

- If statements should contain condition in it followed by statements which are enclosed with in curly braces.
- If statements can be followed by else if statements with same syntax as if statements and can end with optional else statement.
- Condition in the if or else if should be a boolean statement (statements should be evaluated to true or false)

Syntax for if, else if, else statements:-

```
if(boolean expression){statements}
else if(boolean expression){statements}
else{statements}
```

3.2.2 Loop statements:

loop statement should start with 'iterate' keyword followed by three statements seperated by colons (:) enclosed by braces ('(',')') ,three statements are

- Intialization statement (optional)
- Conditonal statement (compulsory)
- Update statement (optional)

Syntax is as follows:

```
iterate(int i=0:i<10:i++){statements}
```

3.2.3 Break statement:

This statement must be inside iterate loop ,this statement ends the loop immediately.

Syntax is as follows:

```
iterate(int i=0:i<10:i++){
    Statements.
    break.
    statements.
}
```

3.2.4 Continue statement:

This statement must be inside iterate loop ,this statement skips the current iteration of the loop immediately. Syntax is as follows:

```
iterate(int i=0:i<10:i++){  
    Statements.  
    continue.  
    statements.  
}
```

4 Domain

4.1 Encryption and Decryption

We want to abstract out some functions of encryption and decryption of data. We will take some inputs from user regarding the level of encryption, etc, and will try to find the optimal encryption.

5 included libraries

5.1 String operations

We are planning to implement some basic string operations like concatenation, sub-strings, etc.as it will help user to give formatted inputs as he want with strings.

5.2 Matrix Operations

We are planning to implement some basic matrix operations like transpose, multiplication, etc.