



**Bahria University**  
Discovering Knowledge

# ***Language Specification Document***

## ***Submitted By:***

Mania Imam (02-134212-013)

Shavana Yusuf (02-134212-022)

Tayyaba Imam (02-134202-056)

## ***Submitted To:***

Miss Saba Imtiaz

***Bachelor's in Computer Science***

***5th Semester***

***11th December 2023***



**Bahria University**  
Discovering Knowledge

**Motivation:** To build a compiler that uses C++ as its reference language. We made changes in the language to make it easier to understand. Increase its scope and, make it simple.

**Targeted Audience:** People unfamiliar with C++.

**Language Paradigm:** Our language is structured. Our language allows the programmer to divide the whole program into modules and functions.

**Case sensitivity:** Yes

**Keywords:**

Keyword	Keyword	Keyword
loop (for)	con (if)	end (stoping the loop)
num (int)	non (else)	ghost (return)
dec (float)	depression (void function)	sike (break)
rizz (String)	fosho (while)	twocents (cin)
calm (char)	ASCII (getch)	yap (cout)
slay (switch)	dawg (do)	bin (bool)
girlboss (array)	gaslight (if else)	gatekeep (do)
bussin (continue)	boujee (case)	drip (default)
purr (include)	basic (main)	levels (class)
facts (true)	cap (false)	chill (finally)
salty (import)	Rent-free (public)	vibe (private)
woke (protected)	Base (structure)	



## Data Types:

int- **num**

float- **dec**

string- **rizz**

char- calm

bool- bin

array- girlboss

## Iterative statements:

**for loop** = loop (int a=1;a<=10;a++)  
{....}

**while loop** = fosho(true)  
{statements};

## Conditional Statements:

con (true): statement            →if  
non: statement                    →else  
slay (true)                        →switch-case  
{  
boujee 1:  
{.....};  
boujee 2:  
{.....} ...  
drip:  
{.....}



}

**Comments (Multi line + Single line):** Multi line ~\ \~  
Single line \\\

**Line terminator : -**

**Operators:**

<b>Arithmetic Operator</b>	
+	addition
-	Subtraction
*	Multiplication
/	Division
%	Modulus
<b>Assignment Operator</b>	
=	Equal to
+=	Add + assign
-=	Subtract + assign
*=	Multiply + assign
/=	Divide + assign
%=	Modulus + assign
<b>Logical Operator</b>	
&	and
	or
!	not
<b>Comparison Operator</b>	
==	Is Equal
!=	Is not equal
<	Is less than
>	Is greater than



<=	Is less than equal to
>=	Is greater than equal to

### **Punctuators:**

( )
{ }
[ ]
...    or -->(for comment)
/n (for line terminator)
!
// // (use in string)
/ / (use in char)
\t
- (instead of ;)

**Identifiers :** Can be made using characters a to z, A to Z, no special character and numbers are allowed except underscore but only in between of characters. identifier must start with \$

Valid = \$a, \$ab, \$a\_a

Invalid = \$\_a, \_\$a, \$aa\_, \$ a, \$ a\_b, 9\$a, \$9b, \$a9



## Syntax Specification:

### **Functions** → keyword difference

Func a()                      →function  
{.....}

### **Array** →

#### **1) 1D ARRAY:**

arr \*\*datatype\*\* arrayName [ arraySize ]- //initialization

arr \*\*datatype\*\* arrayName [ arraySize ] equals { }- //declaration

#### **2) MULTIDIMENSIONAL ARRAY:**

arr \*\*datatype\*\* arrayName[size1][size2]....[sizeN] –  
//initialization

arr \*\*datatype\*\* arrayName[size1][size2] ....[sizeN] equals {{},  
{}, {}},  
//declaration

### **Structure**

base name  
{  
Datatype, identifier, value(optional);  
...  
}-



**Bahria University**  
Discovering Knowledge

## Pointers:

<b>Access operator</b> (sub element)	.
<b>pointer</b>	^
<b>Address</b>	#&