*Introduction to Programming *

Basics of C: Keywords, Identifiers, Datatypes, variable constants, Input/output, operators, storage close format specifiers

• Control Statement: if else, for, while do while loop, switch. break, continue

• Functions & Reccursion : Actual & formal arguments, Parameter Passing technique ,Scoping

• concept of painter : Introduction Pointer & Array, Pointer & function, Dynamic memory allocatin

Strings: Library functions, Pointer & strings..

-Structure & union.

-File Handling.

Features of C

1. High level lang.

2. Small level long having 32 keywords 3. Core lang. as many other programming language are dependent on C

4. Portable

5. Built in functions & operators.

6. Structured lang.

7. Pointers

8. Extensible long.

9. Compilation and execution is faster.

10. Dynamic memory allocation

11. Platform dependent

Structure of C

1. Documentation section—

(includes //, /* */) comments.

2. Link section-- (#include <stdio.h>) ➝➝ (printf scanf)

++ (Standard input output)

#include <conio.h> -> getch()

++(Console input output)

#include <math.h> (√, power).

#include <string.h> ->(strlen, stramp, etc)

Definition Section - # define pi 3.14159

# define max 200 etc.

4. Global declaration section -- int a, void sum(), void sub()

5. Main function- int main(),

6. Sub Program Section

Variables in C

A **variable** is a name of the memory location. It is used to store data. Its value can be changed, and it can be reused many times.

It is a way to represent memory location through symbol so that it can be easily identified.

Let's see the syntax to declare a variable:

A= 50

B= 60

int a: 10

float a= 3.14

Char as 'a'

```
→ void main()
{
Int a ;
a = 10
print f("%d", a);
getch();
}
```

*Rules for Constructing variable names.

Variables

Average

Sum 12

Sum _12

_sum

Etc.

different variables

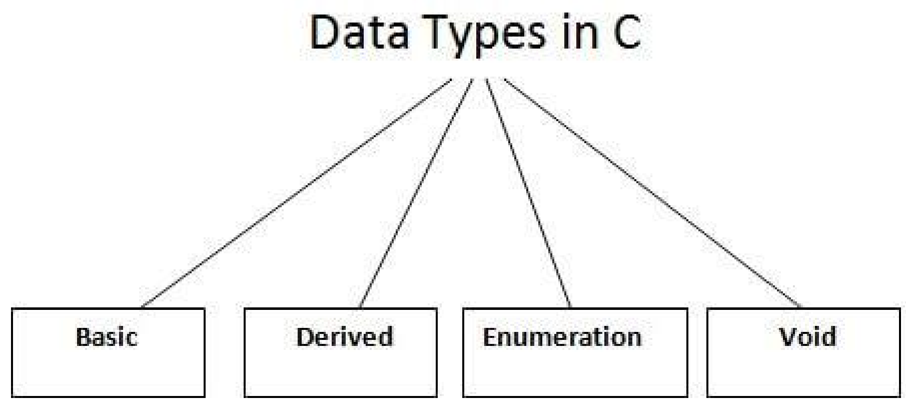
NUM

Num

num

Data Types in C

A data type specifies the type of data that a variable can store such as integer, floating, character, etc.



There are the following data types in C language.

Types	Data Types
Basic Data Type	int, char, float, double
Derived Data Type	array, pointer, structure, union
Enumeration Data Type	enum
Void Data Type	void

Keywords in C

A keyword is a **reserved word**. You cannot use it as a variable name, constant name, etc. There are only 32 reserved words (keywords) in the C language.

A list of 32 keywords in the c language is given below:

auto	break	case	char	const	continue	default	do
double	else	enum	extern	float	for	goto	if
int	long	register	return	short	signed	sizeof	static
struct	switch	typedef	union	unsigned	void	volatile	while

C Identifiers

C identifiers represent the name in the C program, for example, variables, functions, arrays, structures, unions, labels, etc. An identifier can be composed of letters such as uppercase, lowercase letters, underscore, digits, but the starting letter should be either an alphabet or an underscore. If the identifier is not used in the external linkage, then it is called as an internal identifier. If the identifier is used in the external linkage, then it is called as an external identifier.

Rules for constructing C identifiers

- The first character of an identifier should be either an alphabet or an underscore, and then it can be followed by any of the character, digit, or underscore.
- It should not begin with any numerical digit.
- In identifiers, both uppercase and lowercase letters are distinct. Therefore, we can say that identifiers are case sensitive.
- Commas or blank spaces cannot be specified within an identifier.

- Keywords cannot be represented as an identifier.
- The length of the identifiers should not be more than 31 characters.
- Identifiers should be written in such a way that it is meaningful, short, and easy to read.

Example of valid identifiers

1. total, sum, average, _m _, sum_1, etc.

C Operators

An operator is simply a symbol that is used to perform operations. There can be many types of operations like arithmetic, logical, bitwise, etc.

There are following types of operators to perform different types of operations in C language.

- Arithmetic Operators
- Relational Operators
- Shift Operators
- Logical Operators
- Bitwise Operators
- Ternary or Conditional Operators
- Assignment Operator
- Misc Operator

Category	**Operator**	**Associativity**
Postfix	() [] -> . ++ - -	Left to right
Unary	+ - ! ~ ++ - - (type)* & sizeof	Right to left
Multiplicative	* / %	Left to right
Additive	+ -	Left to right
Shift	<< >>	Left to right

<table>
<tr><td>Relational</td><td>< <= > >=</td><td>Left to right</td></tr>
<tr><td>Equality</td><td>== !=</td><td>Left to right</td></tr>
<tr><td>Bitwise AND</td><td>&</td><td>Left to right</td></tr>
<tr><td>Bitwise XOR</td><td>ˆ</td><td>Left to right</td></tr>
<tr><td>Bitwise OR</td><td>|</td><td>Left to right</td></tr>
<tr><td>Logical AND</td><td>&&</td><td>Left to right</td></tr>
<tr><td>Logical OR</td><td>||</td><td>Left to right</td></tr>
<tr><td>Conditional</td><td>?:</td><td>Right to left</td></tr>
<tr><td>Assignment</td><td>= += −= *= /= %=>>= <<= &= ˆ= |=</td><td>Right to left</td></tr>
<tr><td>Comma</td><td>,</td><td>Left to right</td></tr>
</table>