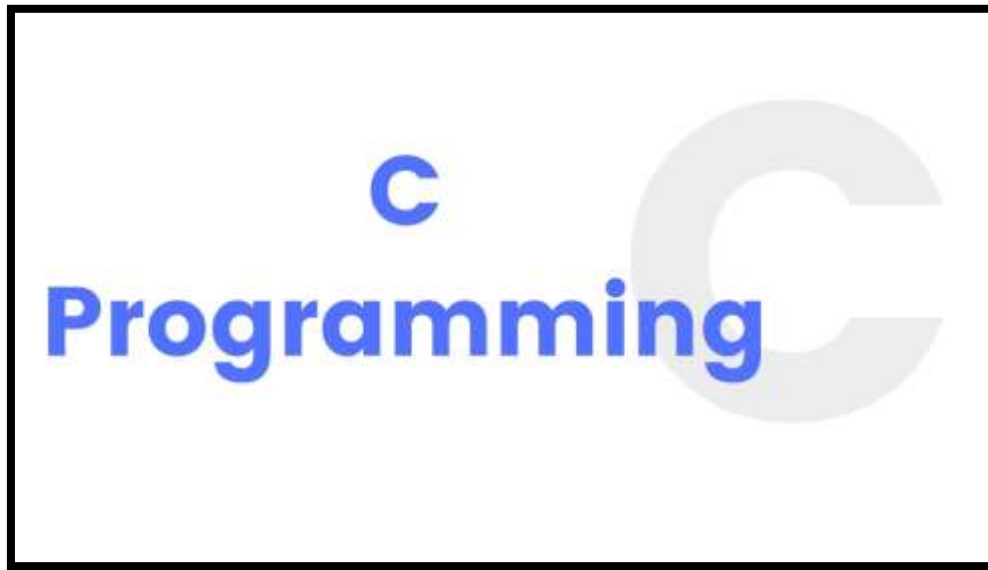


Introduction



What is C language?

C is a mid-level and procedural programming language. The Procedural programming language is also known as the structured programming language.

C is a general-purpose programming language that is extremely popular. It can develop software like game, OS, database, compilers and etc.

Why to Learn:

1. Easy to learn.
2. Structured language.
3. It produces efficient programs.
4. It can handle low-level activities.
5. It can be compiled on a variety of computer platforms.

History:



Founder and Father of C Language

C language was developed in 1972 by American computer scientist Dennis Ritchie at Bell Laboratories.

It was created from 'ALGOL', 'BCPL' and 'B' programming languages. Initially, C language was developed and 'C' is a powerful programming language which is strongly associated with the UNIX OS. UNIX Operating System is coded in C language.

Programming languages that were developed before C language:

Language	Year	Developed By
Algol	1960	International Group
BCPL	1967	Martin Richard
B	1970	Ken Thompson
Traditional C	1972	Dennis Ritchie
K & R C	1978	Kernighan & Dennis Ritchie

ANSI C	1989	ANSI Committee
ANSI/ISO C	1990	ISO Committee
C99	1999	Standardization Committee

GYANSABHA.IN

Keywords and Identifiers:

Keyword are reserved words used in programming that have special meanings to the compiler. keywords are part of the syntax.

You cannot be used a variable name, constant name, etc.

There are only 32 keywords in the C Language.

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

C identifiers represent the name given to entities such as variables, functions, structures and etc. It is and user-define names consisting of C Program standard character set.

It is must be unique. They are created to give a unique name to an entity to identify.

For Example:

```
int roll_no;
```

```
float name;
```

In this example, **int** and **float** are keyword.

Roll_no and **name** are identifiers.

Rules must be followed for identifiers:

1. Keywords are not allowed to be used as identifiers.
2. It is case-sensitive in the language.

3. The first letter of an identifier should be either a letter or an underscore.
4. The valid identifier can have uppercase and lowercase letter, underscores and digits.

Variable and constants:

A variable is a container to store data that helps to execute the program. Constants can never change at the time of execution.

A variable can be defined in both ways, like the uppercase and lowercase letters, digits and the underscores. A variable can store the value of the different data types like int, float, char.

Rules:

1. A variable name can only have letters, digits and underscore.
2. There should consist not be a white space in a variable name.
3. There should consist not be a keyword in a variable name.

Example:

```
//declare the variable in C.
```

```
Datatype variable_name;
```

```
Datatype variable_name1, variable_name2, variable_name3;
```

Data Types:

Integers: An integer is a numeric literals without any fractional or exponential part.

There are 3 types of integer literals:

- octal
- decimal
- hexadecimal

Floating: It can hold a floating point value which means a number is having a fraction and a decimal part.

Characters: This data types are used to store a single character value enclosed in single quotes.

Constants:

Constants are the fixed values that never change during the execution of a program.

They are the basic data types integer, float, character and string:

Integer:

An integer constant prefix specifies the base. An integer constant can be an octal, decimal and hexadecimal constant.

Example: 0, 1, 2, 123, 5767, 0X23, 0XFFF, etc.

Floating:

A float constant has an integer part, a decimal point, fractional form and exponential form.

Example: 0.5, 32.02, PI, etc.

Character:

Character constants are enclosed in single quotation marks. 'y' can be stored in a simple variable of char type.

String:

String constants are enclosed in a double quote. A character may be letters, numbers, special symbols and some blank space.

Data Type:

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

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

Basic Data Type:

The basic data type are integer-based and floating-point based.

The memory size of the basic data types may change according to 32 or 64-bit operating system.

Data Types	Memory Size	Range
char	1 byte	−128 to 127
signed char	1 byte	−128 to 127
unsigned char	1 byte	0 to 255
short	2 byte	−32,768 to 32,767
signed short	2 byte	−32,768 to 32,767
unsigned short	2 byte	0 to 65,535
int	2 byte	−32,768 to 32,767
signed int	2 byte	−32,768 to 32,767
unsigned int	2 byte	0 to 65,535
short int	2 byte	−32,768 to 32,767
signed short int	2 byte	−32,768 to 32,767
unsigned short int	2 byte	0 to 65,535

long int	4 byte	-2,147,483,648 to 2,147,483,647
signed long int	4 byte	-2,147,483,648 to 2,147,483,647
float	4 byte	
double	8 byte	
long double	10 byte	

Derived Data Types:

The fundamental data types are derived types. For example: arrays, pointers, function types, structures, etc.

Input/output

You will learn to use printf() function to display output to the user and scanf() function to take input from the user.

printf():

It is an output function. It is print the five n statement to the console.

Example:

```
#include <stdio.h>

int main()
{
    printf("Gyansabha");
}
```

Output:

Gyansabha

scanf():

It is an input function. It reads the input data from the console.

Example:

```
#include <stdio.h>

int main()
{
    int a,b,c;
    printf("Enter the two Numbers: ");
    scanf("%d%d", &a, &b);
    c=a+b;
```

```
printf("The Sum of Two Numbers: %d",c);  
}
```

Output:

Enter the two Numbers: 12

12

The Sum of Two Numbers: 24

GYANSABHA.IN

Operators

An operator is a symbol that is used to perform the operations. They are different types of operations:

1. Arithmetic Operators
2. Increment and Decrement Operators
3. Assignment Operators
4. Relational Operators
5. Logical Operators
6. Bitwise Operators

1. **Arithmetic Operators:** It is perform the mathematical operations (like addition, subtraction, multiplication, division, etc.).

Operator	Meaning of Operator
+	Addition
-	Subtraction
*	Multiplication
/	Division
%	remainder after division (modulo division)

2. **Increment and Decrement Operators:** This two operators increment and decrement to change the value of an operand by 1.
prefix: The value of `var(++var)` is incremented by 1 then it returns the value.
postfix: The value of `var(var++)` is incremented by 1.

3. **Assignment Operators:** An operator is used for assigning a value to a variable.

Operator	Type	Example
=	a=b	a = b
+=	a+b	a += b
-=	a-b	a -= b
*=	a*b	a *= b
/=	a/b	a /= b
%=	a%b	a = a%b

4. **Relational Operators:** This operator checks the relationship between two operands.

Relation true then return 1 and Relation false return 0.

Operator	Meaning of Operator
==	Equal to
>	Greater than
<	Less than
!=	Not equal to
>=	Greater than or equal to
<=	Less than or equal to

5. **Logical Operators:** logical operator returns either 0 or 1 depending upon whether expression results true or false.

Operator	Meaning
&&	Logical AND. True only if all operands are true.
	Logical OR. True only if either one operand is true.

!	Logical NOT. True only if the operand is 0.
---	---

6. **Bitwise Operators:** arithmetic operation are converted to bit-level operation.

Operators	Meaning
&	Bitwise AND
	Bitwise OR
^	Bitwise exclusive OR
~	Bitwise complement
<<	Shift left
>>	Shift right