

THE C CHARACTER SET: A character set is a set of characters which can be used to represent the information in C language.

A character denotes any alphabet, digit or special symbol used to represent information in C language which are:

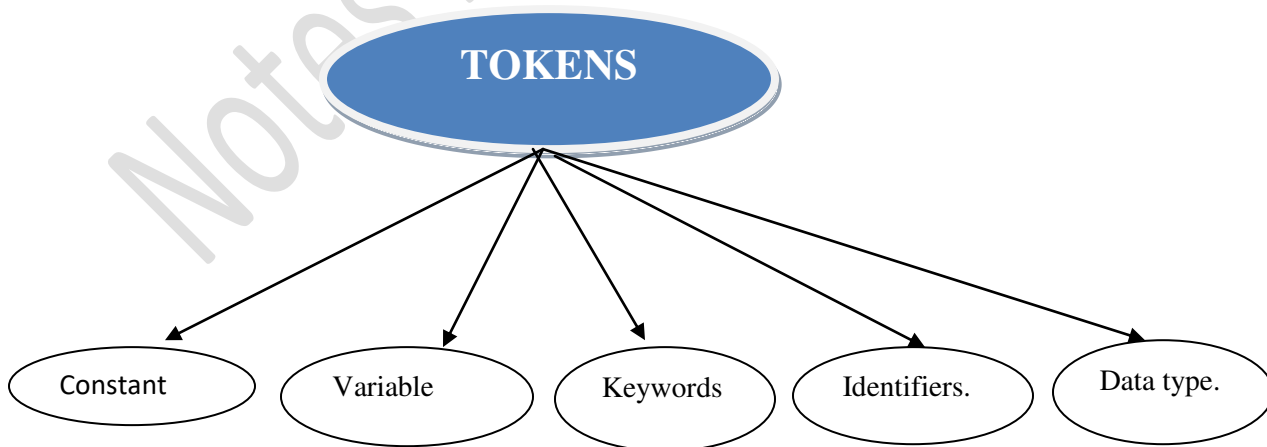
- | | |
|--------------------|--------------------------|
| 1. ALPHABETS | A, B-----Z
a, b-----z |
| 2. DIGITS | 0,1,2,3,4,5,6,7,8,9 |
| 3. SPECIAL SYMBOLS | ~, %, #, 0, *, +-----etc |
| 4. WHITE SPACES | |

A sequence set of instructions, which are written to the user according to their need to achieve a specific goal, is called a program.

Tokens: --Tokens are the smallest elements of a program, which are meaningful to the compiler. In C programs, each word and punctuation is referred to as a token. C Tokens are the smallest building block or smallest unit of a C program.

Types of tokens

- Constant
- Variable
- Identifiers.
- Keywords
- Data type.
- Operators.



CONSTANT:

Constant refers to a value that does not change during the execution of a program. It is similar to mathematical constants like "Pie" whose value is always assumed to be equal to $22/7$.

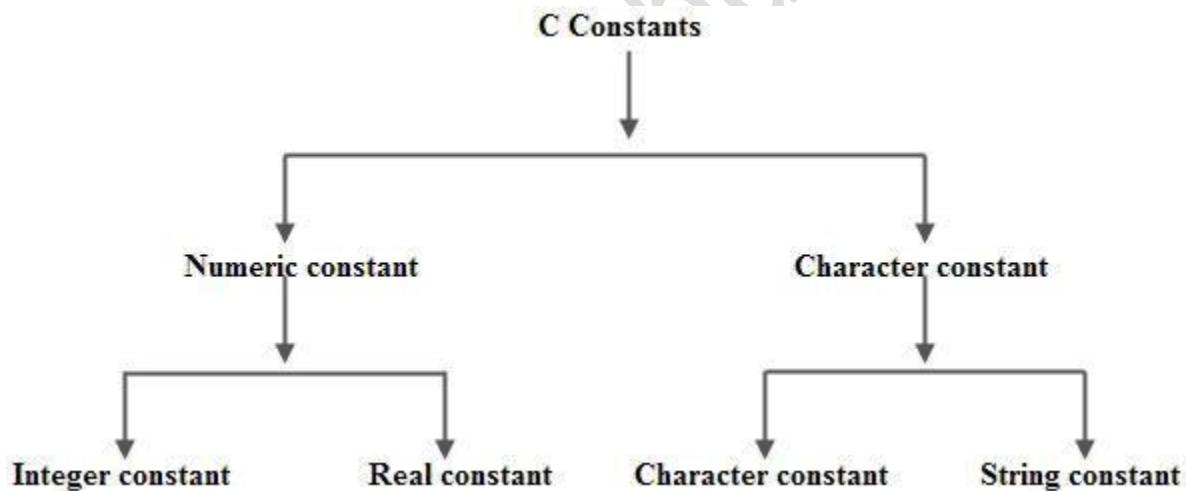
C Constants is the most fundamental and essential part of the C programming language. Constants in C are the fixed values that are used in a program, and its value remains the same during the entire execution of the program.

Constants may be numbers, characters or strings and are also known as literals. .

E.g. $x=2y + 5z$. is a statement. in which 2,5 is constants

Syntax: `const type constant_name;`

Types of Constant



Integers Constant - An integers constant is an integer valued number. It combines a sequence of digits. It is any combination of digits from 0 to 9.

Rules for constructing Integer Constant

1. An integers constant must have at least one digit.
2. It must not have a decimal point.
3. It may be either positive or negative. E.g. 1,4,45. -34, -89, -6
4. No commas or blank space are allowed in an integers constant.

5. It used to store 2 byte memory.

Float Constant-> The numbers containing fractional parts like 99.25 are called real or floating points constant.

A float constant is a float value which contains a decimal point. It is a combination of decimal no.

e.g. 12.5, 2.4, 5.6.

Rules for constructing Float Constant

- Float constant must have at least one digit.
- It must contain decimal no.
- if we do not use decimal no it automatically contains decimal no.
- It may be positive or negative. Default is positive.
- It assumes 4 byte computer memory.

Character Constant -> A character Constant is a single alphabet, single digits or single special symbol which is enclosed in single inverted commas. It simply contains a single character enclosed within ' and ' (a pair of single quote).

e.g. 'A' 'a' '%' '6' '*'

String Constant -> A string constant is a set of characters enclosed in double quotation marks. The characters in a string constant sequence may be an alphabet, number, special character and blank space.

E.g. "PALVI" "1234" "God Bless""!.....?"

VARIABLES

Variables are those à values that can change any time during the program execution.

The primary purpose of variables is to store data in memory for later use. Unlike **constants** which do not change during the program execution, variables value may change during execution.

These locations can contain integer, real or character constants.

Ex:- $6x+3y=77$.

Here x and y are variables and 6, 3 and 77 are constants.

RULES FOR CONSTRUCTING VARIABLE NAMES:

- A variable name is any combination of alphabets, digits and underscores.
- No commas or blanks are allowed within a variable name.
- No special symbol, except the underscore is allowed.
- Special characters like #, \$ are not allowed.
- C keywords cannot be used as variable names.
- The first character in the variable name must be an alphabet
- Values of the variables can be numeric or alphabetic.
- Variable type can be char, int, float, double, or void.

Declaration of variables:

Every variable used in the program should be declared to the compiler. The declaration does two things :

- Tell the compiler the variable names.
- Specifies what type of data the variable will hold.

The general format of any declaration.

type variable_name;

or

Data type v1,v2,v3-----vn;

Where v1.v2,v3-----vn are variable names.

variables are separated by commas. A declaration statement must end with a semicolon.

Ex-int sum, avg;

The declaration for a variable can also contain an initial value.

Ex- int i=10, a=200;

KEYWORDS:

Keywords are the words whose meaning has already been explained to the C compiler.

Keywords cannot be used as a variable name. The meaning and working of these keywords are already known to the compiler. These are also called "**RESERVED WORDS**".

There are 32 keywords available in C.

Falling table shows:

Auto	double	static	Break
else	int	strict	Case
enum	long	switch	Char
extern	near	typeset	Const
float	register	union	Continue
For	return	unsigned	Default
for	short	void	Do
goto	signed	while	if

They are specifically used by the compiler for its own purpose and they serve as building blocks of a program. They cannot be redefined by the programmer.

IDENTIFIERS:

Identifiers are the names given to the various elements of a program such as variables, function, arrays. structure, unions etc. Their meaning has not been defined to the c compiler.

Rules For Naming Identifiers:

- An identifier can only have alphanumeric characters (a-z , A-Z , 0-9) (i.e. letters & digits) and underscore(_) symbol.
- Identifier names must be unique
- The first character must be an alphabet or underscore.
- You cannot use a keyword as identifiers.
- It must not contain white spaces.

Example:

```
int amount;  
double totalbalance;
```

In the above example, amount and totalbalance are identifiers and int, and double are keywords.