C Programming Basic Concepts

Character Set

Character set refers to a set of all the valid characters that we can use in the source program for forming words, expressions, and numbers.

C has the following character set:

Type of Character	Description	Characters
Lowercase Alphabets	a to z	a, b, c, d, e, f, g, h, i, j, k, l, m, n, o, p, q, r, s, t, u, v, w, x, y,z
Uppercase Alphabets	A to Z	A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U,V, W, X, Y, Z
Digits	0 to 9	0, 1, 2, 3, 4, 5, 6, 7, 8, 9
Special Characters	–	` ~ @ ! $ # ^ * % & ( ) [ ] { } < > + = _ – | / \ ; : ‘ “ , . ?
White Spaces	–	Blank Spaces, Carriage Return, Tab, New Line

C Tokens

“Every smallest individual unit of a c program is called token.”

Tokens are used to construct cprograms and they are said to the basic building blocks of a c program.

Some of the C tokens are given below:

- Keywords
- Identifiers
- Constants
- Operators
- Expression
- Separators

Keywords

Keywords are the reserved words with predefined meaning which already known to the compiler.

Properties of Keywords

- All the keywords in C programming language are defined as lowercase letters so they must be used only in lowercase letters
- Every keyword has a specific meaning, users can not change that meaning.
- Keywords cannot be used as user-defined names like variable, functions, arrays, pointers,etc...
- Every keyword in C programming language represents something or specifies some kind of action to be performed by the compiler.

Example

```
int marks;
float num;
```

Here, **int** and **float** are keywords.

In the C programming language, there are **32 keywords**. All the 32 keywords have their meaningwhich is already known to the compiler.

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

Identifiers

The identifier is a user-defined name given to various programming elements such as variables, constants, arrays, pointers, functions etc.

Example

int marks;

float num;

Here, **marks** and **num** are identifiers.

Rules for naming Identifiers

- An identifier can contain **letters** (UPPERCASE andlowercase), **numerics** & **underscore** symbol only.
- An identifier should not start with a numerical value. It can start with a letter or an underscore.
- We should not use any special symbols in between the identifier even whitespace.However, the only underscore symbol is allowed.
- Keywords should not be used as identifiers.
- Both uppercase and lowercase letters are permitted.

Constants

Constants are values that do not change during the execution of the program.

Types of Constants:

Numeric Constants

- Integer constants
- Float (real) constants

Character Constants

- Single character constants
- String constants

Numeric Constants:

- **Integer constants**:

It refers to a sequence of digits without a decimal point. It can be represented in

three different number systems.

Decimal constants:
Decimal Integers consist of digits 0 to 9 preceded by an optional + or - sign.
Spaces, commas and non-digit characters are not permitted between digits.

Examples
123 -31 0 562321 + 78

Octal constants:
Octal Integers constant consists of any combination of digits from 0 through 7
with an 0 at the beginning.

Examples
026, 011, 0347, 0676

Hexadecimal constants:

0X or 0x precedes a hexadecimal integer constant; they may contain alphabets from A to F. The alphabets A to F refer to 10 to 15 in decimal digits.

Examples
0X2, 0x8C, 0XBCD, 0x12

- **Float/ Real constants**

A floating-point constant must contain both integer and decimal parts.

Example
3.14, 123.45

Character Constants

- **Single character constant:**

It is an alphanumeric or any symbol enclosed within single quotation marks.

Examples ‘5’, ‘x’, ‘;’

But C contains a few character two but are treated as one character.

Example ‘\n’ , ’\t’

- **String constants**

A string constant is a set of characters enclosed in double quotation marks.

The characters in a constant string sequence may be an alphabet, number, special characters and blank space.

Examples "SURESH" ,"1234", "God Bless", “%^&”

Creating constants in C

In a c programming language, constants can be created using two concepts...

- Using the 'const' keyword
- Using '#define' preprocessor

Using the 'const' keyword

We create a constant of any datatype using **const** keyword. To create a constant, we prefix the variable declaration with **const** keyword.

Syntax const datatype variable_name = value; **Example** const int x = 10 ;

Here, 'x' is a integer constant with fixed value 10.

Example Program

```
#include<stdio.h>

int main()

{

const int x = 10

printf(“ The value of x= %d”,x) ;
return 0;
}
```

Output

The value of x=10;

Using '#define' preprocessor

#define is a useful C component that allows the programmer to give a name to a constant value before the program is compiled.

Syntax

#define CONSTANT_NAME value

Example

#define PI 3.14

Here, PI is a constant with value 3.14

Example Program

```
#include<stdio.h>
#define PI 3.14
int main()

{
int r, area ;
printf("Enter the radius of circle : \n") ;
scanf("%d", &r) ;
area = PI * (r * r) ;
printf("Area of the circle = %d", area) ;
return 0;
}
```

Output

Enter the radius of circle : 6

Area of the circle = 113

Variables

Variable is a name given to a memory location where we can store different values of the same datatype during the program execution.

Rules to specify a variable name...

- Variable name should not start with a digit.
- Keywords should not be used as variable names.
- A variable name should not contain any special symbols except underscore(_).
- A variable name can be of any length but compiler considers only the first 31 characters of the variable name.

Declaration of Variable

Declaration of a variable tells the compiler to allocate the required amount of memory with the specified variable name and allows only specified data type values into that memory location.

Declaration Syntax:

datatype variable_name;

Example:

int number;

The above declaration tells to the compiler that allocates **2 bytes** of memory with the name **number** and allows only integer values into that memory location.

Assigning values to variables

Assigning a value to variable at the time declaration is called initialization .

Syntax

datatype variable_name= value;

Example

int a=10;

Values can be assigned to variable using the assignment operator =

Variables and Memory Address

All the variables of the C program will have a memory address.

This address depends on the computer storage (hard disk). So, this address varies from computer to computer.

We have a unary operator & (ampersand) to see the address of the variable.

int num; // is a variable.

Then **&num** will be the memory address of a variable.

```
#include<stdio.h>
int main ()
{
int num=5;
printf ("Value of num is = %d",num);
printf ("Address of num is = %u", &num);
return 0;
}
```

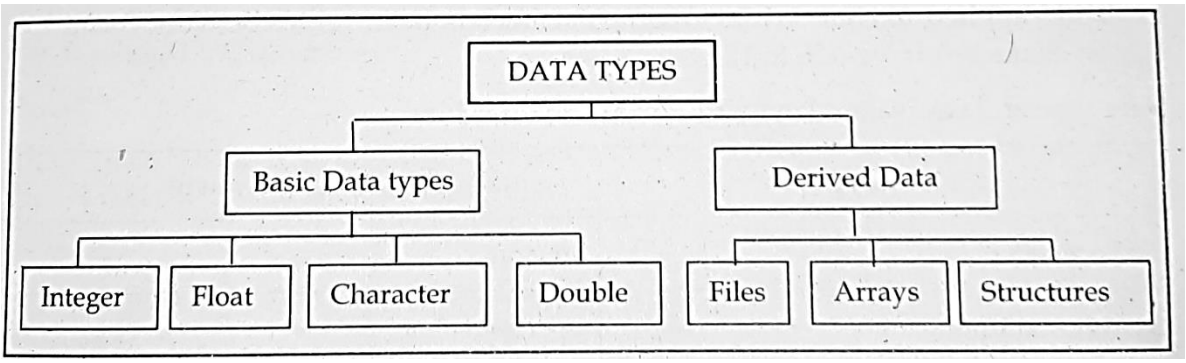
Datatypes

Data types in the c programming language are used to specify what kind of value can be stored in a variable.

The memory size and type of the value of a variable are determined by the variable data type.

In the c programming language, data types are classified as follows...

- Primary data types (Basic data types OR Predefined data types)
- Derived data types (Secondary data types OR User-defined data types)



Integer Data types

- The integer data type is a set of whole numbers. Every integer value does not have the decimal value. It can hold only whole number.
- Keyword **int** used to represent integer data type in c. It takes 2 bytes of memory space.
- The integer data type is used with different type modifiers like short ,long, singed, unsinged.

Type	Size (Bytes)	Range	Specifier
int (signed short int)	2	-32768 to +32767	%d
short int (signed short int)	2	-32768 to +32767	%d
long int (signed long int)	4	-2,147,483,648 to +2,147,483,647	%d
unsigned int (unsigned short int)	2	0 to 65535	%u
unsigned long int	4	0 to 4,294,967,295	%u

Floating Point data types

- Floating-point data types are a set of numbers with the decimal value. The floating-point data types has two variants
  float

  double
- Keyword **float** used to represent floating-point data type in c. It takes 4 bytes of memory space.
- It can hold both integer and fractions. It also called as single precision floating number. It has 6 places of decimal accuracy.
- Keyword **double** used to represent double data type in c. It takes 8 bytes of memory space.
- It can hold too small and too big numbers. It also called as double precision floating number. It has 15 places of decimal accuracy.

Type	Size (Bytes)	Range	Specifier
float	4	1.2E - 38 to 3.4E + 38	%f
double	8	2.3E-308 to 1.7E+308	%ld
long double	10	3.4E-4932 to 1.1E+4932	%ld

Character data type

- The character data type is a set of characters enclosed in single quotations.
- Keyword **char** used to represent character data type in c. It takes 1 bytes of memory space.
- It can hold any single character from the C-language character set.

Type	Size (Bytes)	Range	Specifier
char (signed char)	1	-128 to +127	%c
unsigned char	1	0 to 255	%c