

Data Types in C Language

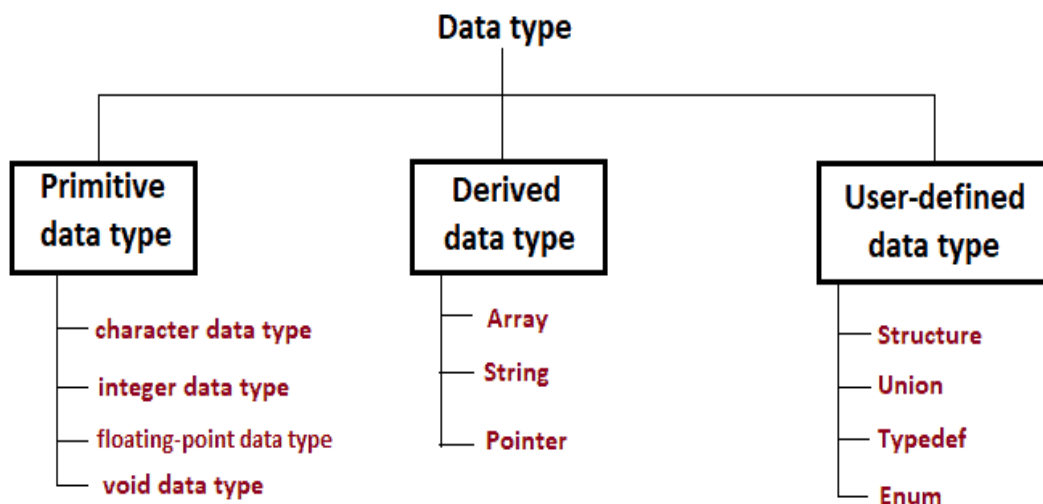
Introduction:

The data-type in a programming language is the collection of data with values having fixed meaning as well as characteristics. Some of them are an integer, floating point, character, etc. Usually, programming languages specify the range values for given data-type.

C Data Types are used to:

- Identify the type of a variable when it declared.
- Identify the type of the return value of a function.
- Identify the type of a parameter expected by a function.

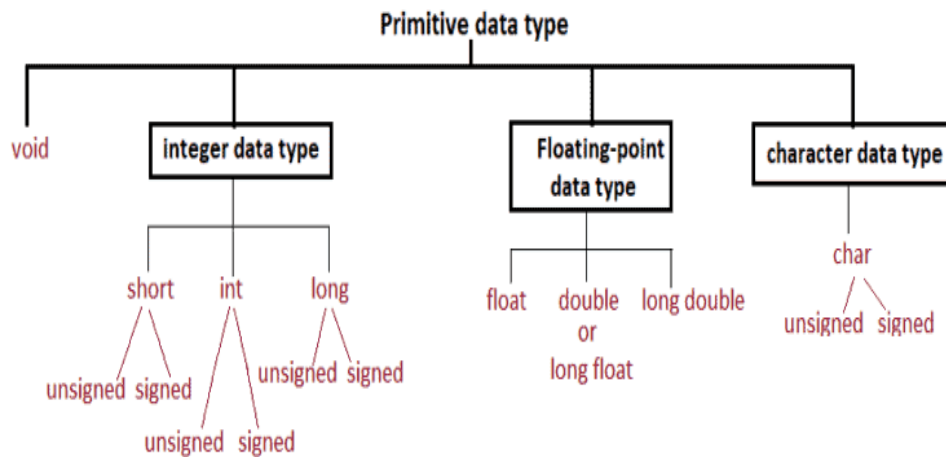
Classification of Data Types:



Modifiers:

- Modifiers are prefixed with basic data types to **modify (either increase or decrease)** the amount of **storage space** allocated to a variable.
- There are two types of type modifiers:
- Size modifiers - short, long (can be used with int and double)
- Sign modifiers - signed, unsigned (can be used with int and char data type)

Primitive (or Basic or Built in) Data types:



Integer Data Types:

Integers are used to store whole numbers. The keyword **int** is used to declare integer data type.

Size and range of Integer type on 16-bit machine:

Type	Size(bytes)	Range
int or signed int	2	-32,768 to 32767
unsigned int	2	0 to 65535
short int or signed short int	1	-128 to 127
unsigned short int	1	0 to 255
long int or signed long int	4	-2,147,483,648 to 2,147,483,647
unsigned long int	4	0 to 4,294,967,295

Formula for Range:

- Let the size of signed int is n bits then range is - 2^{n-1} to $+(2^{n-1} - 1)$
- For unsigned int- 0 to $2^n - 1$

Note:

- We can't store decimal values using int data type.
- If we use int data type to store decimal values, decimal values will be truncated and we will get only whole number.
- In this case, float data type can be used to store decimal values in a variable.

Example:

```
#include<stdio.h>
int main ()
{
    short a = 10; //short data type
    short int a1 = 20;
    int a2 = 1000; //int data type
    long a3 = 1000000000; //long data type
    long int a4 = 2000000000;
    long long a5 = 5000000000000; //long long data type
    long long int a6 = 9000000000000;
    printf("%d\n",a);
    printf("%d\n",a1);
    printf("%d\n",a2);
    printf("%ld\n",a3);
    printf("%ld\n",a4);
    printf("%lld\n",a5);
    printf("%lld\n",a6);
    return 0;
}
```

Output:-

```
10
20
1000
1000000000
2000000000
5000000000000
9000000000000
```

Float Data type:

- All numeric data type items with fractional part belong to float type.
- The keyword **float** is used to declare variables of float type.
Ex: **float** var1;
- By default, every floating-point number is treated as a double data type. Float and long double data type are also used for floating-point.
- Generally, the size of the float data type is 4 bytes and the double data type is 8 bytes.

- In floating point numbers, the precision describes the number of significant decimal places that a floating value carry.

Data types	Format Specifier	Size (In Bytes)	Digits of Precision
float	%f	4	6
double	%lf	8	15
long double	%Lf	10	19

Character data type:

- In C language, to store character data types keyword **char** is used.
- For character type variables and single-character constants, **1 byte** (8 bits) of memory space is allocated.
- It is classified under the integral data type as the storage occurs in the form of ASCII values which are used to represent every character.

For Ex: 'a' has the decimal value **97**.

- Example of char data types: - 'm', 'A', '5', '@', '?' e.t.c.

Data Type	Format Specifier	Size (in Byte)	Range
unsigned char	%c	1	0 to 255
char	%c	1	-128 to 127
signed char	%c	1	-128 to 127

Void Data Type:

- As the name indicates this type has no values.
- Most of the times it is used to indicate that a function does not return any value.
- All other primitive data types short, int, long int, float, double and long double can be used for both calculation (like storing values to a variable) and returning from a function but void can only be used for returning from a function.
- Void can't be used for storing and calculation in a program.

Example:

1. When used as a function return type:

the void keyword specifies that the function does not return a value.

```
void show()
{
    printf("This function has no return type");
}
```

2. When used for a function's parameter list:

void specifies that the function takes no parameters.

```
int sum(void)
{
    int a,b;
    printf("Enter Two number>> ");
    scanf("%d%d",&a,&b);
    return a+b;
}
```

Derived data types in C:

Those data types which are derived from the fundamental data types are called **derived data types**. Function, arrays, and pointers are derived data types in C programming language.

For example, an array is derived data type because it contains the similar types of fundamental data types and acts as a new data type for C.

User define data type in C:

C allows the feature called type definition which allows programmers to define their identifier that would represent an existing data type. There are three such types:

Structure: It is a package of variables of different types under a single name. This is done to handle data efficiently. "struct" keyword is used to define a structure.

Union: These allow storing various data types in the same memory location. Programmers can define a union with different members, but only a single member can contain a value at a given time.

Enum: Enumeration is a special data type that consists of integral constants, and each of them is assigned with a specific name. "enum" keyword is used to define the enumerated data type.

Typedef: The C programming language provides a keyword called typedef, which you can use to give a type a new name. Following is an example to define a term BYTE for one-byte numbers –

typedef unsigned char BYTE;

After this type definition, the identifier BYTE can be used as an abbreviation for the type unsigned char, for example-

BYTE b1, b2;