

Data Types:

In C programming language, each variable has a data type associated with it. The data type specifies the type of data that the variable can store such as integer, character, floating, double, etc. Each data type requires different amounts of memory and has some specific operations that can be performed over it. The data type is a collection of data with values having fixed meanings and characteristics.

Different data types in C programming language have different ranges up to which they can store numbers. These ranges may vary from compiler to compiler. Below is a list of ranges along with the memory requirement and format specifiers on the 32-bit GCC compiler.

Data Type	Size (bytes)	Range	Format Specifier (Conversion Character)
short int	2	-32,768 to 32,767 (-2^{15} to $2^{15} - 1$)	%hd
unsigned short int	2	0 to 65,535 (0 to $2^{16} - 1$)	%hu
unsigned int	4	0 to 4,294,967,295 (0 to $2^{32} - 1$)	%u
int	4	-2,147,483,648 to 2,147,483,647 (-2^{31} to $2^{31} - 1$)	%d
long int	4	-2,147,483,648 to 2,147,483,647 (-2^{31} to $2^{31} - 1$)	%ld
unsigned long int	4	0 to 4,294,967,295 (0 to $2^{32} - 1$)	%lu
long long int	8	-2^{63} to $2^{63} - 1$	%lld
unsigned long long int	8	0 to 18,446,744,073,709,551,615 (0 to $2^{64} - 1$)	%llu
signed char	1	-128 to 127 (-2^7 to $2^7 - 1$)	%c
unsigned char	1	0 to 255 (0 to $2^8 - 1$)	%c
float	4	1.2E-38 to 3.4E+38 (1.2×10^{-38} to 3.4×10^{38})	%f
double	8	1.7E-308 to 1.7E+308 (1.7×10^{-308} to 1.7×10^{308})	%lf
long double	16	3.4E-4932 to 1.1E+4932 (3.4×10^{-4932} to 1.1×10^{4932})	%Lf

Note: The *long*, *short*, *signed* and *unsigned* are datatype modifier that can be used with some primitive data types to change the size or length of the datatype.

The following are some main primitive data types in C:

1. Integer data type:

The integer datatype in C is used to store the whole numbers without decimal values. Octal values, hexadecimal values, and decimal values can be stored in *int* data type in C.

Syntax: `int variable_name;`

Integer data type can also be used as *unsigned int*, *short int*, *long int* and *unsigned short int*.

2. Character data type:

Character data type allows its variable to store only a single character. It stores a single character and requires a single byte of memory.

Syntax: `char variable_name;`

3. Float data type:

Float in C is used to store decimal and exponential values. *float* data type is used to store floating-point values. It is used to store decimal numbers (numbers with floating point values) with single precision.

Syntax: `float variable_name;`

4. Double data type:

A Double data type in C is used to store decimal numbers (numbers with floating point values) with double precision. It is used to define numeric values which hold numbers with decimal values in C.

The double data type is basically a precision sort of data type that is capable of holding 64 bits of decimal numbers or floating points. Since double has more precision as compared to that float then it is much more obvious that it occupies twice the memory occupied by the floating-point type. It can easily accommodate about 16 to 17 digits after or before a decimal point.

Syntax: `double variable_name;`

Size of data types:

sizeof() operator in C can be used to check the size of a variable defined inside a C program. As size of some data types does depend upon the architecture, so one can verify the size using *sizeof()*.

The following example shows the use of *sizeof()* operator.

```
#include<stdio.h>
int main(){
    //Declaring variables
    int myIntegerVariable;
    float myFloatVariable;
    double myDoubleVariable;
    char myCharacterVariable;

    //Assigning values to the variables
    myIntegerVariable = 1234;
    myFloatVariable = 2.5f;
    myDoubleVariable = 12.293123;
    myCharacterVariable = 'A';

    //Printing the values and size of variables
```

```
printf("\nValue of Integer variable: %d",myIntegerVariable);
printf("\nSize of Integer variable: %d",sizeof(myIntegerVariable));
printf("\nValue of Float variable: %f",myFloatVariable);
printf("\nSize of Float variable: %d",sizeof(myFloatVariable));
printf("\nValue of Double variable: %lf",myDoubleVariable);
printf("\nSize of Double variable: %d",sizeof(myDoubleVariable));
printf("\nValue of Character variable: %c",myCharacterVariable);
printf("\nSize of Character variable: %d",sizeof(myCharacterVariable));
return 0;
}
```

OUTPUT OF THE CODE:

```
Value of Integer variable: 1234
Size of Integer variable: 4
Value of Float variable: 2.500000
Size of Float variable: 4
Value of Double variable: 12.293123
Size of Double variable: 8
Value of Character variable: A
Size of Character variable: 1
```