Data Types in C

In this tutorial you will master the basic data types like character, integer and decimal used in C with example programs.Also you will learn about modifiers used in C to alter the data types.

How can we define Data types in C ?

Compared to any other programming languages, C has more simple and compact data types. Data type can be defined as a kind of data that a variable or object can hold and perform operations based on the data. It also helps to determine the type and size of the data a programmer intends to compile or interpret. Based on the data type, memory can be allocated which means we can allot the space required for the data and for its operation in memory.

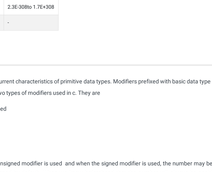
All types of data input taken by the C processor comprise of mainly three types :

- character
- integer number
- decimal number.

In C language these three types are represented as '**char**', '**int**', '**float**' respectively, called **primary data types**. For instance :

1. int a, int b , int c /* a, b, c are variables of data type integer.*/
2. char x, y, z; /* x, y, z are variables of data type character.*/
3. float pi = 3.14; /*pi is a variable of datatype float*/

In fact, the number of data types in this language is infinite as the user can define new data types according to his need. In the next section, we will talk about these primary data types in detail.



Data types in C

As we have already seen, data types are of various types and hence have different specifications. Below table gives the format specifier, memory size and value range corresponding for each data type in a 32 bit architecture system.

Datatype	Format specifier	Memory Size	Value Range
char	%c	1 byte	-128 to 127
int	%d	2 bytes	-32,768 to 32,767
float	%f	4 bytes	1.2E-38 to 3.4E+38
double	%lf	8 bytes	2.3E-308to 1.7E+308
void	-	-	-

What are modifiers in C

Modifiers are keywords used to alter the current characteristics of primitive data types. Modifiers prefixed with basic data type can either increase or decrease the size or sign of the data type. Basically there are two types of modifiers used in c. They are

- Sign Modifiers - signed or unsigned
- Size Modifiers – short or long

Sign Modifiers

The number is always **positive** when the unsigned modifier is used and when the signed modifier is used, the number may be **positive or negative**. If the modifier is not mentioned then a signed qualifier is assigned by default. When we know beforehand that the number will always be positive we usually use the unsigned modifier. Moreover , these modifiers can only be used with int and char types.

Size Modifiers

Size Modifiers help to increase or decrease the size of the data types. When a short modifier is used it reduces the range of data type while the long modifier does the reverse, i.e increases the range of data type. The important note to be considered is that these size modifiers work well with int data type , double can use only long . Both char and float do not use any of these modifiers.

As in C size of data type depends on the machine used,for a 16 bit machine the size of the type is 2 bytes while for a 32 or 64 bit machine the size of the type is 4 bytes.

C - INTEGER DATA TYPES

The data type 'int' implies any positive or negative integer number within a specific range. The range varies according to the compiler, for example: In case of 16-bit compilers like Turbo C/C++, the range is -32768 to 32767 whereas 32-bit compilers like Visual C++ it is - 2147483648 to 2147483647. In general a 32-bit processor like Intel Pentium successfully runs a 16-bit compiler but the vice versa is not allowed.

Data Type	Format Specifier	Memory Size	Range
short int / short signed int	%d	2 bytes	-32,768 to 32,767
unsigned short int	%hu	2 bytes	0 to 65,535
int / signed int	%d	4 bytes	-2147483648 to 2147483647
unsigned int	%d	4 bytes	0 to 4294967295
short int	%hd	2 bytes	-32,768 to 32,767
long int / signed long int	%ld	8 bytes	-2,147,483,648 to 2,147,483,647
Unsigned long int	%lu	8 bytes	0 to 4,294,967,295

'int' can be defined in two different subsets: short and long. They can be declared as 'short int' or 'short' and 'long int' or 'long' respectively. The range of the former type is -32768 to 32767 and the later is 2147483648 to 2147483647. Use of 'short' data type wherever applicable, significantly increases the processing speed. 'int' can be also divided into two subcategories signed and unsigned. Changing data type from 'int' to 'unsigned int' doubles its number storing capacity (0 to 65535) omitting all the negative ones.

```
#include <stdio.h>
#include <stdlib.h>
void main()
{

int a;
int short b;
short c;
int long d;
long e;
 printf(" Size of  a is %d\n",sizeof(a));
 printf(" Size of  b is %d\n",sizeof(b));
 printf(" Size of  c is %d\n",sizeof(c));
 printf(" Size of  d is %d\n",sizeof(d));
 printf(" Size of  e is %d\n",sizeof(e));

 printf(" Signed int - INT_MAX     :   %d\n", INT_MAX);
 printf(" Signed int - INT_MIN     :   %d\n", INT_MIN);
 printf(" Unsigned int - UINT_MAX  :   %u\n", (unsigned int) UINT_MAX);

}
```

Output:

```
Size of  a is 4
 Size of  b is 2
 Size of  c is 2
 Size of  d is 4
 Size of  e is 4

 Signed int - INT_MAX     :   2147483647
 Signed int - INT_MIN     :   -2147483648
 Unsigned int - UINT_MAX  :   4294967295
```

C- FLOATING-POINT DATA TYPE:

If you need to use any fractional or decimal number you must have declared Float data type. It allocates 4 bytes of memory in RAM and its value ranges from -3.4e38 to 3.4e38. For bigger numbers there is another data type 'double' which occupies 8 bytes of memory. It's value ranges from -1.7e308 to 1.7e308. If the required number is even bigger, you can use 'long double' which occupies 10 bytes of memory having a range of -1.7e4932 to 1.7e4932.A tabular representation is shown below.

Type	Format Specifier	Memory size	Value range	Precision
float	%f	4 byte	1.2E-38 to 3.4E+38	6 decimal places
double	%lf	8 byte	2.3E-308 to 1.7E+308	15 decimal places
long double	%Lf	10 byte	3.4E-4932 to 1.1E+4932	19 decimal places

Illustrations of these days type's declaration are as follows:

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <float.h>
void main()
{

float x;
double y;
long double z;

 printf(" Size of  x is %d\n",sizeof(x));
 printf(" Size of  y is %d\n",sizeof(y));
 printf(" Size of  z is %d\n\n",sizeof(z));

 printf(" Float Minimum Range     :   %g\n", (float) FLOAT_MIN);
 printf(" Float Maximum Range     :   %g\n", (float) FLOAT_MAX);
 printf(" Double Maximum Range    :   %g\n", (double) DBL_MAX);

}
```

Output:

```
Size of  x is 4
 Size of  y is 8
 Size of  z is 16

Float Minimum Range     :   1.17549e-038
Float Maximum Range     :   3.40282e+038
Double Maximum Range    :   1.79769e+308
```

C- CHAR DATA TYPE

Simply the data type 'char' has been designed to take a single input (single key depression) through a keyboard like a, b, c etc. Although It may seem to be a bit confusing but there are two types of 'char' as well - **signed and unsigned**. Actually every character of the keyboard corresponds to a certain ASCII value. A simple program will show it.

```
#include <stdio.h>
#include <stdlib.h>
#include <limits.h>
#include <float.h>

void main()
{

char c;

        printf("Enter any character -->");
        scanf("%c", &c);
        printf("ASCII value of %c is %d\n", c, c);
        printf("Size of  c is %d\n\n",sizeof(c));

        printf("Character Minimum Range   :   %d\n", CHAR_MIN);
        printf("Character Maximum Range   :   %d\n", CHAR_MAX);

}
```

In this program '%c' will show the exact character and '%d will display the corresponding ASCII value. For example, if you enter 'C' the output would be ' The ASCII value of C is 67.

```
Enter any character -->C
ASCII value of C is 67
Size of  c is 1

Character Minimum Range   :   -128
Character Maximum Range   :   127
```

Here comes the concept of signed and unsigned char. As its name suggests the former has these numerical values ranging from -128 to +127, whereas the later have 0 to 255. Later we will have to work with numerical values of a character. In those cases assigning a data type to be 'unsigned char' helps a lot, as calculation of positive integers is much easier. The following table shows the size and range of char data types in C programming.

Type	Format Specifier	Memory size	Value range
float	%c	1 byte	-128 to 127
double	%c	1 byte	0 to 255

C - VOID DATA TYPE

'Void' in C represents 'nothing' or 'null'. It is solely different from any other data types like 'int or 'char' which correspond to an entity with a specific property. It is mostly used in pointer related operations where we need to change the value of a variable at address level. To understand this we have to know about 'pointers' which will be discussed later.

It will be easier to understand 'void' thorough analyzing function return type. If a function is defined as 'int check();' or char check();, it means that return value will be an integer or a character respectively. Here is a simple example:

```
int check();
{
int a;
a = 5;
return  a;
}
```

But when a function has to return nothing, rather doing a simple task like printing a statement void can be used. For example:

```
void check_it();
{
printf(" Void is printing a statement ");
}
```

This void tells the compiler not to take any argument.