C Data Types

**In this tutorial, you will learn about basic data types such as int, float, char etc. in C programming.**

In C programming, data types are declarations for variables. This determines the type and size of data associated with variables. For example,

1. int myVar;

Here, myVar is a variable of int (integer) type. The size of int is 4 bytes.

**Basic types**

Here's a table containing commonly used types in C programming for quick access.

| Type | Size (bytes) | Format Specifier |
| --- | --- | --- |
| int | at least 2, usually 4 | %d |
| char | 1 | %c |
| float | 4 | %f |
| double | 8 | %lf |
| short int | 2 usually | %hd |
| unsigned int | at least 2, usually 4 | %u |
| long int | at least 4, usually 8 | %li |
| long long int | at least 8 | %lli |
| unsigned long int | at least 4 | %lu |
| unsigned long long int | at least 8 | %llu |
| signed char | 1 | %c |
| unsigned char | 1 | %c |
| long double | at least 10, usually 12 or 16 | %Lf |

**int**

Integers are whole numbers that can have both zero, positive and negative values but no decimal values. For example, 0, -5, 10

We can use int for declaring an integer variable.

1. int id;

Here, id is a variable of type integer.

You can declare multiple variables at once in C programming. For example,

1. int id, age;

The size of int is usually 4 bytes (32 bits). And, it can take 232 distinct states from -2147483648 to 2147483647.

**float and double**

float and double are used to hold real numbers.

1. float salary;
2. double price;

In C, floating-point numbers can also be represented in exponential. For example,

1. float normalizationFactor = 22.442e2;

What's the difference between float and double?

The size of float (single precision float data type) is 4 bytes. And the size of double (double precision float data type) is 8 bytes.

**char**

Keyword char is used for declaring character type variables. For example,

1. char test = 'h';

The size of the character variable is 1 byte.

**void**

void is an incomplete type. It means "nothing" or "no type". You can think of void as **absent**.

For example, if a function is not returning anything, its return type should be void.

Note that, you cannot create variables of void type.

**short and long**

If you need to use a large number, you can use a type specifier long. Here's how:

1. long a;
2. long long b;
3. long double c;

Here variables a and b can store integer values. And, c can store a floating-point number.

If you are sure, only a small integer ([−32,767, +32,767] range) will be used, you can use short.

short d;

You can always check the size of a variable using the sizeof() operator.

1. #include <stdio.h>
2. int main() {
3. short a;
4. long b;
5. long long c;
6. long double d;
7. printf("size of short = %d bytes\n", sizeof(a));
8. printf("size of long = %d bytes\n", sizeof(b));
9. printf("size of long long = %d bytes\n", sizeof(c));
10. printf("size of long double= %d bytes\n", sizeof(d));
11. return 0;
12. }

**signed and unsigned**

In C, signed and unsigned are type modifiers. You can alter the data storage of a data type by using them. For example,

1. unsigned int x;
2. int y;

Here, the variable x can hold only zero and positive values because we have used the unsigned modifier.

Considering the size of int is 4 bytes, variable y can hold values from -231 to 231-1, whereas variable x can hold values from 0 to 232-1.

Other data types defined in C programming are:

* bool Type
* Enumerated type
* Complex types

**Derived Data Types**

Data types that are derived from fundamental data types are derived types. For example: arrays, pointers, function types, structures, etc.