

type modifier in C++

Here we will discuss the following,

[1. C++ modifier](#)

[2. why we use modifier in C++?](#)

[3. Type of modifier in C++](#)

[3.1. signed in C++](#)

[3.1.1. Example of signed in C++](#)

[3.2. unsigned in C++](#)

[3.2.1. Example of unsigned in C++](#)

[4. long in C++](#)

[4.1. Example of long in C++](#)

[4.2. short in C++](#)

C++ modifier

Means- to modify. We use the modifier in C++ with the built-in data-type (except void). using a modifier, we can modify the property of a data-type, for example, we can increase the size of a variable, range or determine a variable which type of input will be store (positive or negative) by the end-user,

why we use modifier in C++?

It can be understood in such a way that the information given by the user is more than the size of the variable size In this situation, the variable can not hold that information, in that case, the modifier can be used with that data-type.

Let's see in other words,

For example, using modifier in C++ with `int` data type, we can store large form of a number (maybe mobile no) or define a condition where we need to store the only positive value in a variable e.g. a student's roll-no or age always a positive types value. so we can use modifier with them.

here is the syntax, to declare a modifier,

SYNTAX

```
modifier-type data-type variable-name;
```

where data-type can be of `int`, `float`, and `char` types and variable-name is an identifier.

Type of modifier in C++

The modifier in C++ are following type,

- signed
- unsigned
- long
- short

before proceeding modifier will reserve the space in the memory as follows,

signed in C++

This type of modifier used to store both negative and positive values. By default, an `int` data type is a signed type modifier which stores both types of value positive and negative.

SYNTAX

```
signed int variable-name;
```

or

```
signed variable-name;
```

such as,

```
signed int a = 5, b = -10;
```

```
signed x = 3, y = 2;
```

Let's try with an example,

Example of signed in C++

here we will go with `int` data-type, In the below Program we perform the subtraction operation with two number.

```
#include<iostream>

using namespace std;

int main()

{

    int num1 = 5, num2 = 6

    signed int sum;

    sum = num1 - num2;

    cout<<"Total : "<<sum;

    return 0;

}
```

OUTPUT

```
Total: -1
```

Explanation

as you can see in the above Program, we subtract **num1** from **num2** means **greater number – smaller number** so the result will be in a negative form which is store in a variable **sum**, a `signed` type modifier. but we already say `int` type of data-type a `signed` type modifier so,

```
int sum;
```

also, work.

unsigned in C++

By using it with a data-type, we can store only positive values in a variable. `char` data-type by default is an unsigned type modifier.

SYNTAX

```
unsigned int variable-name;
```

```
unsigned char variable-name;
```

Example

```
unsigned int a = 5; // possible
```

```
unsigned int b = -10;// not possible, garbage value printed
```

Let's try in Program,

Example of unsigned in C++

```
//header files
```

```
int main()
```

```
{
```

```
int num1 = 5, num2 = 6

signed int sum;

sum = num1 - num2;

cout<<"Total : "<<sum;

}
```

OUTPUT

```
Total: 43892
```

Explanation

a garbage value is printed because here, we declare an `int` data-type a signed type modifier, but when we replace 8th statement with the following statement

```
sum = num2 - num1;
```

than OUTPUT will be as follows,

```
1
```

because the result is in a positive form now.

long in C++

using this modifier you can extend the size of the variable. It is used only with the `int` and `double`.

SYNTAX

```
long int variable-name;
```

same as

```
long variable-name;
```

In general, the size of an `int` data type is 2 byte but `long` type modifier with `int` then `long int` size will be 4 bytes. `long` type modifier is used to store a large value.

Example of long in C++

```
//header files

int main()

{

    int x = 123456789101;

    long y =123456789101;

    cout<<"x    : "<<x;

    cout<<"\ny : "<<y;

}
```

OUTPUT

```
x : -23526357

y : 123456789101
```

short in C++

unlike the long type-modifier. using short type modifier with `int` data type decreases the range of `int` data-type size. short type- modifier will be used with `int` data-type

syntax

```
short int variable-name;
```

modifiers can be used with each other such as

```
signed short int variable;
```

```
unsigned long int variable;
```

The size of a data-type is also changed from a modifier. which will as follows,

Type	size in bytes	Range
<i>char</i>	1	-128 to 127
<i>signed char</i>	1	0 to 127
<i>unsigned char</i>	1	-128 to 127
<i>float</i>	4	3.4E-38 to 3.4E+38
<i>double</i>	8	1.7E-to 1.7E+308
<i>Long double</i>	10	3.4E-4932 to 1.1E+4932
<i>int</i>	2	-32768 to 32767
<i>unsigned int</i>	2	0 to 65535
<i>signed int</i>	2	-32768 to 32767
<i>short int</i>	2	-32768 to 32767

<i>Long int</i>	4	-2147483648 to 2147483647
-----------------	---	---------------------------