



Data type modifiers in C

Dr. Sheak Rashed Haider Noori
Professor & Associate Head
Department of Computer Science

Data type modifiers in C



- In c language Data Type Modifiers are keywords used to change the properties of current properties of data type. Data type modifiers are classified into following types.
 1. **long**
 2. **short**
 3. **unsigned**
 4. **Signed**
- Modifiers are prefixed with basic data types to modify (either increase or decrease) the amount of storage space allocated to a variable.
- For example, storage space for **int** data type is 4 byte for 32 bit processor.
 - We can increase the range by using **long int** which is 8 byte.
 - We can decrease the range by using **short int** which is 2 byte.

long

- This can be used to increased size of the current data type to 2 more bytes, which can be applied on **int** or double data types. For example **int** occupy 2 byte of memory if we use **long** with integer variable then it occupy 4 byte of memory.

long int a; → occupies 4 bytes of memory space



2 + 2 → 4 bytes

long double b; → occupies 10 bytes of memory space



2 + 8 → 10 byte

Tutorial4us.com

Syntax

long a; --> by default which represent **long int**.

short



- In general int data type occupies different memory spaces for a different operating system; to allocate fixed memory space short keyword can be used.

Syntax

```
short int a; --> occupies 2 bytes of memory space in every operating system.
```

unsigned



- This keyword can be used to make the accepting values of a data type is positive data type.

Syntax

```
unsigned int a =100;    // right
unsigned int a=-100;    // wrong
```

This declaration allows the **a** variable to store only positive values. An immediate effect is that the range changes from (-32768 to 32767) to (0 to 65536)

Signed



- This keyword accepts both negative or positive value and this is default properties or data type modifiers for every data type.
- **Note:** in real time no need to write signed keyword explicitly for any data type.

Example

```
int a=10; // right
int a=-10; // right
signed int a=10; // right
signed int a=-10; // right
```

Type	Bits	Minimal Range
char	8	-127 to 127
unsigned char	8	0 to 255
signed char	8	-127 to 127
int	16 or 32	-32,767 to 32,767
unsigned int	16 or 32	0 to 65,535
signed int	16 or 32	Same as int
short int	16	-32,767 to 32,767
unsigned short int	16	0 to 65,535
signed short int	16	Same as short int
long int	32	-2,147,483,647 to 2,147,483,647
long long int	64	$-(2^{63} - 1)$ to $2^{63} - 1$ (Added by C99)
signed long int	32	Same as long int
unsigned long int	32	0 to 4,294,967,295
unsigned long long int	64	$2^{64} - 1$ (Added by C99)
float	32	1E-37 to 1E+37 with six digits of precision
double	64	1E-37 to 1E+37 with ten digits of precision
long double	80	1E-37 to 1E+37 with ten digits of precision