

# C++ Variable Types Explanation

Tulgar Cem Güngör

January 4, 2025

## 1 Auto

The `auto` keyword in C++ is used to automatically deduce the type of a variable at compile time based on the initializer expression. It eliminates the need to explicitly specify the type of a variable, which can be particularly useful in cases where the type is complex or verbose. This keyword was introduced in C++11. For instance:

- `std::shared_ptr<std::vector<MyClass<int>>>` *instead of writing this long code, we can use just `auto`.*

## 2 Const

The `const` keyword is used to define variables whose values cannot be modified after initialization. This ensures that the value remains constant throughout its lifetime. It can be applied to variables, pointers, and function arguments. It operates at the run-time; however, `constexpr` operates at compile-time.

- `const int x = 10;`
- `const int* ptr = &x;`

## 3 Constexpr

The `constexpr` keyword is used to declare variables and functions whose values are computed at compile time. The main idea is a performance improvement of programs by doing computations at compile time rather than run time. It guarantees that the value is constant and can be used in constant expressions. `constexpr` was introduced in C++11. If we specify a variable's value at the beginning of the program, we can use `constexpr`; however, if the value comes after the program starts, we need to use `const`.

- `constexpr int square(int n) { return n * n; }`
- `constexpr int max_size = 100;`

## 4 Fixed-width int & Fixed-width float

These types are designed to have a known size and provide a way to ensure the exact size of integer or floating-point variables. It is independent from the platform. For example, `int32_t` is a 32-bit signed integer, and `int64_t` is a 64-bit signed integer. Also, `float32_t` is a 32-bit floating-point type, and `float64_t` is a 64-bit floating-point type.

- `int32_t x = 42;`
- `int64_t y = 1000000;`
- `float32_t x = 3.14f;`
- `float64_t y = 3.141592653589793;`

## 5 `size_t`

The `size_t` type is an unsigned integer or unsigned long type used to represent the size of objects/variables in bytes. It is often used in functions like `sizeof()` to return the size of data structures, arrays, and other objects. The size of `size_t` depends on the platform: it is typically 32 bits on 32-bit systems and 64 bits on 64-bit systems. If the compiler is 32-bit, it is typedef of unsigned int. If the compiler is 64-bit, it is typedef of unsigned long long. `size_t` is more effective than unsigned int or unsigned long because `size_t` can be understood every system, so it is flexible. Moreover, many function in the C++ return `size_t` type instead of int or long; thus, the program can easily fit in to the system thanks to the `size_t`.

- `size_t x = sizeof(int);`
- `size_t y = sizeof(double);`

## 6 `wchar_t`

The `wchar_t` type is a wide character type used to represent wide characters (e.g., Unicode characters). It is typically used for internationalization (i18n) purposes, where characters outside of the ASCII character set need to be represented. `char` can only represent ASCII characters; however, `wchar_t` can represent other type of characters. The size of `wchar_t` depends on the platform; it is usually 16 bits on Windows and 32 bits on Linux.

- `wchar_t ch = L'ä';`

VARIABLES' SIZES	
Type	Size
-----	
int	4
int8_t	1
int16_t	2
int32_t	4
int64_t	8
unsigned int	4
uint8_t	1
int8_t	1
uint16_t	2
uint32_t	4
uint64_t	8
long int	8
unsigned long int	8
long	8
unsigned long	8
long long	8
unsigned long long	8
short int	2
unsigned short int	2
short	2
unsigned short	2
double	8
long double	16
float	4

VARIABLES' MAX AND MIN VALUES		
Type	Max	Min
-----		
int	2147483647	-2147483648
int8_t	127	-128
int16_t	32767	-32768
int32_t	2147483647	-2147483648
int64_t	9223372036854775807	-9223372036854775808
unsigned int	4294967295	0
uint8_t	255	0
uint16_t	65535	0
uint32_t	4294967295	0
uint64_t	18446744073709551615	0
long int	9223372036854775807	-9223372036854775808
unsigned long int	18446744073709551615	0
long	9223372036854775807	-9223372036854775808
unsigned long	18446744073709551615	0
long long	9223372036854775807	-9223372036854775808
unsigned long long	18446744073709551615	0
short int	32767	-32768
unsigned short int	65535	0
short	32767	-32768
unsigned short	65535	0
double	1.79769e+308	2.22507e-308
long double	1.18973e+4932	3.3621e-4932
float	3.40282e+38	1.17549e-38