

std::abs, std::labs, std::llabs, std::imaxabs

Defined in header `<cstdlib>`
 Defined in header `<cmath>`

<code>int</code>	<code>abs(int n);</code>	(1)	(constexpr since C++23)
<code>long</code>	<code>abs(long n);</code>	(2)	(constexpr since C++23)
<code>long long</code>	<code>abs(long long n);</code>	(3)	(since C++11) (constexpr since C++23)
Defined in header <code><cstdlib></code>			
<code>long</code>	<code>labs(long n);</code>	(4)	(constexpr since C++23)
<code>long long</code>	<code>llabs(long long n);</code>	(5)	(since C++11) (constexpr since C++23)
Defined in header <code><cinttypes></code>			
<code>std::intmax_t</code>	<code>abs(std::intmax_t n);</code>	(6)	(since C++11)
<code>std::intmax_t</code>	<code>imaxabs(std::intmax_t n);</code>	(7)	(since C++11)

Computes the absolute value of an integer number. The behavior is undefined if the result cannot be represented by the return type.

If `std::abs` is called with an unsigned integral argument that cannot be converted to `int` by integral promotion, the program is ill-formed.

Overload (6) of `std::abs` for `std::intmax_t` is provided in `<cinttypes>` if and only if `std::intmax_t` is an extended integer type. (since C++11)

Parameters

`n` - integer value

Return value

The absolute value of `n` (i.e. $|n|$), if it is representable.

Notes

In 2's complement systems, the absolute value of the most-negative value is out of range, e.g. for 32-bit 2's complement type `int`, `INT_MIN` is `-2147483648`, but the would-be result `2147483648` is greater than `INT_MAX`, which is `2147483647`.

Example

Run this code

```
#include <iostream>
#include <cstdlib>
#include <climits>

int main()
{
    std::cout << std::showpos
               << "abs(+3) = " << std::abs(3) << '\n'
               << "abs(-3) = " << std::abs(-3) << '\n';

    // std::cout << std::abs(INT_MIN); // undefined behavior on 2's complement systems
}
```

Output:

```
abs(+3) = +3
abs(-3) = +3
```

Defect reports

30.03.2022, 11:43

std::abs, std::labs, std::llabs, std::imaxabs - cppreference.com

The following behavior-changing defect reports were applied retroactively to previously published C++ standards.

DR	Applied to	Behavior as published	Correct behavior
LWG 2192 (https://cplusplus.github.io/LWG/issue2192)	C++98	overloads of std::abs were inconsistently declared in two headers	declared these overloads in both headers

See also

abs _(float)	
fabs	absolute value of a floating point value ($ x $)
fabsf	(C++11) (function)
fabsl	(C++11)

abs _(std::complex)	returns the magnitude of a complex number
	(function template)

abs _(std::valarray)	applies the function <code>abs</code> to each element of valarray
	(function template)

C documentation for **abs**, **labs**, **llabs**

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/numeric/math/abs&oldid=138111"