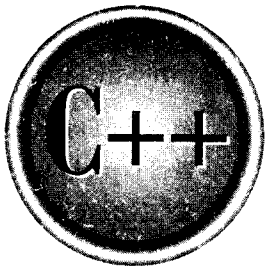
The Complete Reference



Chapter 27

The Mathematical Functions

The standard function library contains several mathematical functions, which fall into the following categories:

- Trigonometric functions
- Hyperbolic functions
- Exponential and logarithmic functions
- Miscellaneous functions

All the math functions require the header **<cmath>.** (C programs must use the header file **math.h.**) In addition to declaring the math functions, this header defines the macro called **HUGE_VAL**. The macros **EDOM** and **ERANGE** are also used by the math functions. These macros are defined in the header **<cerrno>** (or the file **errno.h**). If an argument to a math function is not in the domain for which it is defined, an implementation-defined value is returned, and the built-in global integer variable **errno** is set equal to **EDOM**. If a routine produces a result that is too large to be represented, an overflow occurs. This causes the routine to return **HUGE_VAL**, and **errno** is set to **ERANGE**, indicating a range error. If an underflow happens, the function returns zero and sets **errno** to **ERANGE**.

All angles are in radians.

Originally, the mathematical functions were specified as operating on values of type **double**, but Standard C++ added overloaded versions to explicitly accommodate values of type **float** and **long double**. The operation of the functions is otherwise unchanged.

acos

```
#include <cmath>
float acos(float arg);
double acos(double arg);
long double acos(long double arg);
```

The **acos( )** function returns the arc cosine of *arg*. The argument to **acos( )** must be in the range –1 to 1; otherwise a domain error will occur.

Related functions are **asin( )**, **atan( )**, **atan2( )**, **sin( )**, **cos( )**, **tan( )**, **sinh( )**, **cosh( )**, and **tanh( )**.

asin

```
#include <cmath>
float asin(float arg);
```

```
double asin(double arg);
long double asin(long double arg);
```

The **asin( )** function returns the arc sine of *arg*. The argument to **asin( )** must be in the range –1 to 1; otherwise a domain error will occur.

Related functions are **acos( )**, **atan( )**, **atan2( )**, **sin( )**, **cos( )**, **tan( )**, **sinh( )**, **cosh( )**, and **tanh( )**.

atan

```
#include <cmath>
float atan(float arg);
double atan(double arg);
long double atan(long double arg);
```

The **atan( )** function returns the arc tangent of *arg*.

Related functions are **asin( )**, **acos( )**, **atan2( )**, **tan( )**, **cos( )**, **sin( )**, **sinh( )**, **cosh( )**, and **tanh( )**.

atan2

```
#include <cmath>
float atan2(float y, float x);
double atan2(double y, double x);
long double atan2(long double y, long double x);
```

The **atan2( )** function returns the arc tangent of *y/x*. It uses the signs of its arguments to compute the quadrant of the return value.

Related functions are **asin( )**, **acos( )**, **atan( )**, **tan( )**, **cos( )**, **sin( )**, **sinh( )**, **cosh( )**, and **tanh( )**.

ceil

```
#include <cmath>
float ceil(float num);
double ceil(double num);
long double ceil(long double num);
```

The **ceil( )** function returns the smallest integer (represented as a floating-point value) not less than *num*. For example, given 1.02, **ceil( )** would return 2.0. Given –1.02, **ceil( )** would return –1.

Related functions are **floor( )** and **fmod( )**.

cos

```
#include <cmath>
float cos(float arg);
double cos(double arg);
long double cos(long double arg);
```

The **cos( )** function returns the cosine of *arg*. The value of *arg* must be in radians.

Related functions are **asin( )**, **acos( )**, **atan2( )**, **atan( )**, **tan( )**, **sin( )**, **sinh( )**, **cos( )**, and **tanh( )**.

cosh

```
#include <cmath>
float cosh(float arg);
double cosh(double arg);
long double cosh(long double arg);
```

The **cosh( )** function returns the hyperbolic cosine of *arg*.

Related functions are **asin( )**, **acos( )**, **atan2( )**, **atan( )**, **tan( )**, **sin( )**, **cosh( )**, and **tanh( )**.

exp

```
#include <cmath>
float exp(float arg);
double exp(double arg);
long double exp(long double arg);
```

The **exp( )** function returns the natural logarithm base *e* raised to the *arg* power.

A related function is **log( )**.

fabs

```
#include <cmath>
float fabs(float num);
double fabs(double num);
long double fabs(long double num);
```

The **fabs( )** function returns the absolute value of *num*.
A related function is **abs( )**.

floor

```
#include <cmath>
float floor(float num);
double floor(double num);
long double floor(long double num);
```

The **floor( )** function returns the largest integer (represented as a floating-point value) not greater than *num*. For example, given 1.02, **floor( )** would return 1.0. Given –1.02, **floor( )** would return –2.0.
Related functions are **fceil( )** and **fmod( )**.

fmod

```
#include <cmath>
float fmod(float x, float y);
double fmod(double x, double y);
long double fmod(long double x, long double y);
```

The **fmod( )** function returns the remainder of *x*/*y*.
Related functions are **ceil( )**, **floor( )**, and **fabs( )**.

frexp

```
#include <cmath>
float frexp(float num, int *exp);
```

```
double frexp(double num, int *exp);
long double frexp(long double num, int *exp);
```

The **frexp( )** function decomposes the number *num* into a mantissa in the range 0.5 to less than 1, and an integer exponent such that *num* = *mantissa* * 2^{exp}. The mantissa is returned by the function, and the exponent is stored in the variable pointed to by *exp*.

A related function is **ldexp( )**.

ldexp

```
#include <cmath>
float ldexp(float num, int exp);
double ldexp(double num, int exp);
long double ldexp(long double num, int exp);
```

The **ldexp( )** returns the value of *num* * 2^{exp}. If overflow occurs, **HUGE_VAL** is returned.

Related functions are **frexp( )** and **modf( )**.

log

```
#include <cmath>
float log(float num);
double log(double num);
long double log(long double num);
```

The **log( )** function returns the natural logarithm for *num*. A domain error occurs if *num* is negative, and a range error occurs if the argument is zero.

A related function is **log10( )**.

log10

```
#include <cmath>
float log10(float num);
double log10(double num);
long double log10(long double num);
```

The **log10( )** function returns the base 10 logarithm for *num*. A domain error occurs if *num* is negative, and a range error occurs if the argument is zero.

A related function is **log( )**.

modf

```
#include <cmath>
float modf(float num, float *i);
double modf(double num, double *i);
long double modf(long double num, long double *i);
```

The **modf( )** function decomposes *num* into its integer and fractional parts. It returns the fractional portion and places the integer part in the variable pointed to by *i*.

Related functions are **frexp( )** and **ldexp( )**.

pow

```
#include <cmath>
float pow(float base, float exp);
float pow(float base, int exp);
double pow(double base, double exp);
double pow(double base, int exp);
long double pow(long double base, long double exp);
long double pow(long double base, int exp);
```

The **pow( )** function returns *base* raised to the *exp* power ($base^{exp}$). A domain error may occur if *base* is zero and *exp* is less than or equal to zero. It will also happen if *base* is negative and *exp* is not an integer. An overflow produces a range error.

Related functions are **exp( )**, **log( )**, and **sqrt( )**.

sin

```
#include <cmath>
float sin(float arg);
double sin(double arg);
long double sin(long double arg);
```

The **sin( )** function returns the sine of *arg*. The value of *arg* must be in radians.

Related functions are **asin( )**, **acos( )**, **atan2( )**, **atan( )**, **tan( )**, **cos( )**, **sinh( )**, **cosh( )**, and **tanh( )**.

sinh

```
#include <cmath>
float sinh(float arg);
double sinh(double arg);
long double sinh(long double arg);
```

The **sinh( )** function returns the hyperbolic sine of *arg*.

Related functions are **asin( )**, **acos( )**, **atan2( )**, **atan( )**, **tan( )**, **cos( )**, **tanh( )**, **cosh( )**, and **sin( )**.

sqrt

```
#include <cmath>
float sqrt(float num);
double sqrt(double num);
long double sqrt(long double num);
```

The **sqrt( )** function returns the square root of *num*. If it is called with a negative argument, a domain error will occur.

Related functions are **exp( )**, **log( )**, and **pow( )**.

tan

```
#include <cmath>
float tan(float arg);
double tan(double arg);
long double tan(long double arg);
```

The **tan( )** function returns the tangent of *arg*. The value of *arg* must be in radians.

Related functions are **acos( )**, **asin( )**, **atan( )**, **atan2( )**, **cos( )**, **sin( )**, **sinh( )**, **cosh( )**, and **tanh( )**.

tanh

```
#include <cmath>
float tanh(float arg);
double tanh(double arg);
long double tanh(long double arg);
```

The **tanh( )** function returns the hyperbolic tangent of *arg*.

Related functions are **acos( )**, **asin( )**, **atan( )**, **atan2( )**, **cos( )**, **sin( )**, **cosh( )**, **sinh( )**, and **tan( )**.