Built-In Mathematical Functions

C++ name	operation	Example Use		Value
`sqrt`	square root	`sqrt(2.0)`	=	`1.414214`
`pow`	to-the-power-of	`pow(10.0,3.0)`	=	`1000.0`
`exp`	*e* to-the-power-of	`exp(1.0)`	=	`2.718282`
`log`	natural logarithm	`log(2.718282)`	=	`1.0`
`log10`	base-10 logarithm	`log10(1000.0)`	=	`3.0`
`sin`	sine (see note)	`sin(0.785398)`	=	`0.707107`
`cos`	cosine (see note)	`cos(1.047198)`	=	`0.5`
`tan`	tangent (see note)	`tan(0.785398)`	=	`1.0`
`asin`	arc-sine (see note)	`asin(0.707107)`	=	`0.785398`
`acos`	arc-cosine (see note)	`acos(0.5)`	=	`1.047198`
`atan`	arc-tangent (see note)	`atan(1.0)`	=	`0.785398`
`atan2`	direction (see note)	`atan2(1.0,1.0)`	=	`0.785398`
		`atan2(1.0,-1.0)`	=	`2.356194`

Notes:

Natural Logarithm is the logarithm base-*e* that is preferred in calculus, usually called "*ln*".

Base-10 Logarithm is the kind of logarithm commonly used in human calculations.

The trigonometric functions use radians and not degrees. If you want the sine of 45º, do not use `sin(45.0)`; to convert degrees to radians multiply by π/180, so the sine of 45º is `sin(45*3.14159/180)`. Similarly for the arc-functions, the results are in radians, so although the arc-sine of 0.5 is 30º, `asin(0.5)` is 0.5236. To get the answer in degrees, you must use `asin(0.5)*180/3.14159.`

`atan2(x,y)` calculates the direction from the point (0,0) to the point (x,y). Straight up, or North, or the `+y` direction is 0. Right, or East, or `+x` is 90º. Left, West, or `-x` is -90º. North-East is 45º, etc. Down, South, or `-y` can be either 180º or `-`180º. Naturally, the results are really in radians, so for degrees, again use `asin(0.5)*180/3.14159.`

Special Constants

Constant	Correct Value	How to get it in C++
π, pi	`3.14159265358979323846`	`acos(-1.0)`
e, base of natural logarithms	`2.71828182845904523354`	`exp(1.0)`

It is not exactly convenient to have to remember those formulaæ, but that is a lot better than the alternative of having to memorize the values exactly.