The Math Library Functions <math.h>

Mathematics is relatively straightforward library to use again. The math functions are enclosed in header file. `#include <math.h>`
A common source of error is in forgetting to include the `<math.h>` file . So including the file <math.h> we can use the math functions from library.

Math Functions

Below is the list of some common math functions structures and their functions.

`double acos(double x)` -- Compute and returns arc cosine of x.
`double asin(double x)` – Compute and returns arc sine of x.
`double atan(double x)` – Compute and returns arc tangent of x.
`double atan2(double y, double x)` – *Compute and returns arc tangent of y/x, using the signs of both arguments to determine the quadrant of the return value.*
`double ceil(double x)` -- Get smallest integral value that exceeds x.
`double cos(double x)` -- Compute cosine of angle in radians.
`double cosh(double x)` -- Compute the hyperbolic cosine of x.
`div_t div(int number, int denom)` -- Divide one integer by another.

`div_t` is structure defined in <stdlib.h> as

```
typedef struct
{
      int quot;
      int rem;
 }div_t;
```

`double exp(double x` -- Compute exponential of x
`double fabs (double x )` -- Compute absolute value of x.
`double floor(double x)` -- Get largest integral value less than x.
`double fmod(double x, double y)` -- Divide x by y with integral quotient and return remainder.
`double frexp(double x, int *expptr)` -- Breaks down x into mantissa and exponent of no.

/* frexp and frexpl examples */

```
#include <math.h>
#include <iostream.h>

void main(void)
{
     double mantissa, number;
     int exponent;

     number = 8.0;
     mantissa = frexp(number, &exponent);
```

```
      cout<<"The number " <<number<<" is";
      cout<<mantissa<<" times two to the ";
      cout<<"power of "<<exponent;
}

output: The number 8 is 0.5 times two to the power of 4
```

`long labs(long n)` -- Find absolute value of long integer n.
`double ldexp(double x, int exp)` -- Reconstructs x out of mantissa and exponent of two.

```
/* ldexp example */

#include <iostream.h>
#include <math.h>

int main(void)
{
      double value;
      double x = 5;

      /* ldexp raises 2 by a power of 3
           then multiplies the result by 5   */
      value = ldexp(x,3);
      cout<<"The ldexp value is:"<<value;
}

OutPut: 40.
```

`ldiv_t ldiv(long number, long denom)` -- Divide one long integer by another.

// `ldiv_t` is structure defined in <stdlib.h> as

```
  typedef struct
  {
        int quot;
        int rem;
   }ldiv_t;
```

`double log(double x)` -- Compute log(x).
`double log10 (double x )` -- Compute log to the base 10 of x.
`double modf(double x, double *intptr)` -- Breaks x into fractional and integer parts.

/* modf example */

```
#include <math.h>
#include <iostream.h>
void main(void)
{
```

```
  double fraction, integer;
  double number = 100000.567;

  fraction = modf(number, &integer);
  cout<<"The whole and fractional parts of "<<number<<" are " ;
  cout<<integer<<" and "<<fraction<<endl;
}
```

`double pow (double x, double y)` -- Compute x raised to the power y.
`double sin(double x)` -- Compute sine of angle in radians.
`double sinh(double x)` - Compute the hyperbolic sine of x.
`double sqrt(double x)` -- Compute the square root of x.
`double tan(double x)` -- Compute tangent of angle in radians.
`double tanh(double x)` -- Compute the hyperbolic tangent of x.

Math Constants

The `math.h` library defines many constants. Some of the defined constants are:

`M_E` -- The base of natural logarithms (e).
`M_LOG2E` -- The base-2 logarithm of e.
`M_LOG10E` - The base-10 logarithm of e.
`M_LN2` -- The natural logarithm of 2.
`M_LN10` -- The natural logarithm of 10.
`M_PI` -- π.
`M_PI_2` -- $\pi/2$.
`M_PI_4` -- $\pi/4$.
`M_1_PI` -- $1/\pi$.
`M_2_PI` -- $2/\pi$.
`M_2_SQRTPI` -- $2/\sqrt{\pi}$.
`M_SQRT2` -- The positive square root of 2.
`M_SQRT1_2` -- The positive square root of 1/2.

There are also a number a machine dependent values defined in `<value.h>` see `value.h` for further details.