The Complex class.

Class **Complex** is implemented in a way similar to that described by Stroustrup. In keeping with libg++ conventions, the class is named **Complex**, not **complex**. Complex arithmetic and relational operators are provided (**+**, **-**, *****, **/**, **+=**, **-=**, ***=**, **/=**, **==**, **!=**). Attempted division by (0, 0) triggers an exception.

Complex numbers may be constructed and used in the following ways:

Complex x;	Declares an uninitialized Complex.
Complex x = 2; Complex y(2.0);	Set x and y to the Complex value (2.0, 0.0);
Complex x(2, 3);	Sets x to the Complex value (2, 3);
Complex u(x); Complex v = x;	Set u and v to the same value as x.
double real(Complex& x);	returns the real part of x.
double imag(Complex& x);	returns the imaginary part of x.
double abs(Complex& x);	returns the magnitude of x.
double norm(Complex& x);	returns the square of the magnitude of x.
double arg(Complex& x);	returns the argument (amplitude) of x.
Complex polar(double r, double t = 0.0);	returns a Complex with abs of r and arg of t.
Complex conj(Complex& x);	returns the complex conjugate of x.

Complex cos(Complex& x);	returns the complex cosine of x.
Complex sin(Complex& x);	returns the complex sine of x.
Complex cosh(Complex& x);	returns the complex hyperbolic cosine of x.
Complex sinh(Complex& x);	returns the complex hyperbolic sine of x.
Complex exp(Complex& x);	returns the exponential of x.
Complex log(Complex& x);	returns the natural log of x.
Complex pow(Complex& x, long p);	returns x raised to the p power.
Complex pow(Complex& x, Complex& p);	returns x raised to the p power.
Complex sqrt(Complex& x);	returns the square root of x.
ostream << x;	prints x in the form (re, im).
istream >> x;	reads x in the form (re, im), or just (re) or re in which case the imaginary part is set to zero.