

```

#include <iostream>
using namespace std;

class Complex {
private:
    double real, imag;
public:
    // Default constructor (0 + 0i)
    Complex() : real(0), imag(0) {}

    // Constructor to initialize with real and imaginary parts
    Complex(double r, double i) : real(r), imag(i) {}

    // Overload operator+ to add two complex numbers
    Complex operator+(const Complex& other) {
        return Complex(real + other.real, imag + other.imag);
    }

    // Overload operator* to multiply two complex numbers
    Complex operator*(const Complex& other) {
        double r = real * other.real - imag * other.imag;
        double i = real * other.imag + imag * other.real;
        return Complex(r, i);
    }

    // Overload operator<< to print a complex number
    friend ostream& operator<<(ostream& out, const Complex& c) {
        out << c.real;
        if (c.imag >= 0) out << "+" << c.imag << "i";
        else out << c.imag << "i";
        return out;
    }

    // Overload operator>> to read a complex number
    friend istream& operator>>(istream& in, Complex& c) {
        in >> c.real >> c.imag;
        return in;
    }
};

int main() {
    Complex c1, c2, result;

    // Reading complex numbers
    cout << "Enter first complex number (real and imaginary parts): ";
    cin >> c1;
    cout << "Enter second complex number (real and imaginary parts): ";
    cin >> c2;

    // Adding complex numbers
    result = c1 + c2;
    cout << "Sum: " << result << endl;

    // Multiplying complex numbers
    result = c1 * c2;
    cout << "Product: " << result << endl;

    return 0;
}

```

}