#include <iostream>

using namespace std;

class Complex {

private:

double real;

double imag;

public:

// Default constructor

Complex() : real(0), imag(0) {}

// Parameterized constructor

Complex(double r, double i) : real(r), imag(i) {}

// Overload operator+ to add two complex numbers

Complex operator+(const Complex& other) const {

return Complex(real + other.real, imag + other.imag);

}

// Overload operator\* to multiply two complex numbers

Complex operator\*(const Complex& other) const {

double newReal = real \* other.real - imag \* other.imag;

double newImag = real \* other.imag + imag \* other.real;

return Complex(newReal, newImag);

}

// Overload operator<< to print complex numbers

friend ostream& operator<<(ostream& out, const Complex& c) {

out << c.real << (c.imag >= 0 ? "+" : "") << c.imag << "i";

return out;

}

// Overload operator>> to read complex numbers

friend istream& operator>>(istream& in, Complex& c) {

cout << "Enter real part: ";

in >> c.real;

cout << "Enter imaginary part: ";

in >> c.imag;

return in;

}

};

int main() {

Complex c1, c2;

cout << "Enter the first complex number:" << endl;

cin >> c1;

cout << "Enter the second complex number:" << endl;

cin >> c2;

Complex sum = c1 + c2;

Complex product = c1 \* c2;

cout << "First complex number: " << c1 << endl;

cout << "Second complex number: " << c2 << endl;

cout << "Sum: " << sum << endl;

cout << "Product: " << product << endl;

return 0;

}