#include <iostream>

using namespace std;

class Complex {

public:

float real, imag;

// Constructors

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

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

// Overloaded operators

Complex operator+(const Complex& other) const {

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

}

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

return Complex((real \* other.real) - (imag \* other.imag), (real \* other.imag) + (other.real \* imag));

}

// Overloaded input and output operators

friend ostream& operator<<(ostream& os, const Complex& complexNumber) {

os << complexNumber.real;

if (complexNumber.imag >= 0) {

os << "+";

}

os << complexNumber.imag << "i";

return os;

}

friend istream& operator>>(istream& is, Complex& complexNumber) {

cout << "Enter real part: ";

is >> complexNumber.real;

cout << "Enter imaginary part: ";

is >> complexNumber.imag;

return is;

}

};

int main() {

// Test the Complex class

Complex a, b, c, d;

cout << "Default value of complex number a: " << a << endl;

cout << "Enter the first complex number (real and imaginary part): ";

cin >> a;

cout << "First number is: " << a << endl;

cout << "Enter the second complex number (real and imaginary part): ";

cin >> b;

cout << "Second number is: " << b << endl;

// Addition and multiplication

c = a + b;

cout << "Addition is: " << c << endl;

d = a \* b;

cout << "Multiplication is: " << d << endl;

return 0;

}