OOP Lab Practical 1

Problem Statement :

Implement a class Complex which represents the Complex Number data type. Implement the following

1. Constructor (including a default constructor which creates the complex number 0+0i).

2. Overload operator+ to add two complex numbers.

3. Overload operator* to multiply two complex numbers.

4. Overload operators << and >> to print and read Complex Numbers.

#include <iostream>

using namespace std;

// Class to represent a complex number

class Complex

{

    private:

        double real;   // Real part of the complex number

        double img;   // Imaginary part of the complex number

    public:

        // Default constructor: Initializes complex number to 0 + 0i

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

        // Parameterized constructor: Initializes complex number with given real and

        //imaginary parts

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

        // Overload + operator to add two complex numbers

        Complex operator +(const Complex& other) const

        {

            // Add the real parts and the imaginary parts separately

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

        }

        // Overload * operator to multiply two complex numbers

```cpp
        Complex operator*(const Complex& other) const
        {
                // Use the formula (a + bi) * (c + di) = (ac - bd) + (ad + bc)i
                double realPart = real * other.real - img*other.img;
                double imgPart = real * other.img + img*other.real;
                return Complex(realPart,imgPart);
        }


        // Overload << operator to print complex numbers in the format "a + bi"
        friend ostream& operator<<(ostream& os, const Complex& c)
        {
                if (c.img >= 0)
                        os<<c.real<<"+"<<c.img<<"i";
                else
                        os<<c.real<<"-"<<-c.img<<"i";


                return os;
        }


        // Overload >> operator to read complex numbers from input
        friend istream& operator>>(istream& is, Complex& c)
        {
                cout<<"Enter real part :";
                is>>c.real;
                cout<<"Enter img part :";
                is>>c.img;
                return is;
        }
};


int main()
{
        Complex c1, c2, sum, product;
```

OOP Lab Practical 1

```cpp
// Read two complex numbers from the user
cout<<"Enter first complex number :\n"<<endl;
cin>>c1;
cout<<"Enter second complex number :\n"<<endl;
cin>>c2;

// Compute the sum and product of the two complex numbers
sum = c1+c2;
product = c1*c2;

// Display the results
cout<<"Sum:"<<sum<<endl;
cout<<"Product:"<<product<<endl;

return 0;
}
```

// OUTPUT

```
Enter first complex number :

Enter real part :2
Enter img part :2
Enter second complex number :

Enter real part :1
Enter img part :1
Sum:3+3i
Product:0+4i


=== Code Execution Successful ===
```