**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 Number

**PROGRAM/SOURCE CODE:**

***#include* <iostream>**

**class Complex**

**{**

**private:**

**double real;**

**double imag;**

**public:**

***// Constructors***

**Complex() : real(0.0), imag(0.0) {}**

**Complex(double real, double imag) : real(real), imag(imag) {}**

***// 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 result\_real = real \* other.real - imag \* other.imag;**

**double result\_imag = real \* other.imag + imag \* other.real;**

***return* Complex(result\_real, result\_imag);**

**}**

***// Overload the << operator to print Complex Numbers***

**friend std::ostream &operator<<(std::ostream &os, const Complex &complex)**

**{**

**os << complex.real;**

***if* (complex.imag >= 0)**

**{**

**os << " + " << complex.imag << "i";**

**}**

***else***

**{**

**os << " - " << -complex.imag << "i";**

**}**

***return* os;**

**}**

***// Overload the >> operator to read Complex Numbers***

**friend std::istream &operator>>(std::istream &is, Complex &complex)**

**{**

**std::cout << "Enter real part: ";**

**is >> complex.real;**

**std::cout << "Enter imaginary part: ";**

**is >> complex.imag;**

***return* is;**

**}**

**};**

**int main()**

**{**

**Complex c1, c2;**

**std::cin >> c1;**

**std::cin >> c2;**

**Complex sum = c1 + c2;**

**Complex product = c1 \* c2;**

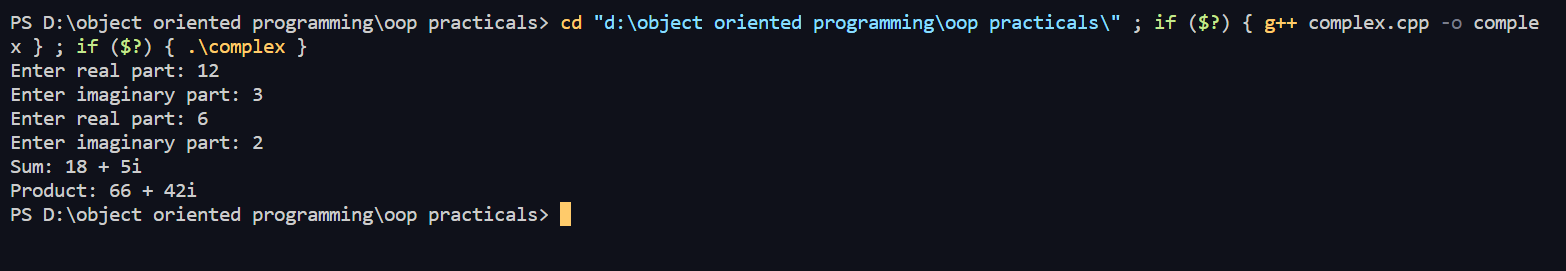
**std::cout << "Sum: " << sum << std::endl;**

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

***return* 0;**

**}**

**OUTPUT:**

****