**Name – Ashish Bhangale**

**Class – FYMCA**

**Division – A, Batch- A1**

**Roll No- 07**

**Subject – Object Oriented Programming.**

**Assignment No – 4**

**Assignment Title -** Design a class ‘Complex ‘with data members for real and imaginary part. Provide default and parameterized constructors. Write a program to

perform arithmetic operations of two complex numbers using operator

overloading.

i. Addition and subtraction using friend functions

ii. Multiplication and division using member functions

**Input –**

**#include <iostream>**

**using namespace std;**

**class complex {**

**public:**

**double real, imaginary;**

**// Default constructor**

**complex() {**

**real = 0.0;**

**imaginary = 0.0;**

**}**

**// Parameterized constructor**

**complex(double r, double i) {**

**real = r;**

**imaginary = i;**

**}**

**complex operator\*(const complex& b) const {**

**return complex((real \* b.real) - (imaginary \* b.imaginary),**

**(real \* b.imaginary) + (imaginary \* b.real));**

**}**

**complex operator/(const complex& b) const {**

**double denominator = (b.real \* b.real) + (b.imaginary \* b.imaginary);**

**return complex(((real \* b.real) + (imaginary \* b.imaginary)) / denominator,**

**((imaginary \* b.real) - (real \* b.imaginary)) / denominator);**

**}**

**friend complex operator+(const complex& b1, const complex& b2) {**

**return complex(b1.real + b2.real, b1.imaginary + b2.imaginary);**

**}**

**friend complex operator-(const complex& b1, const complex& b2) {**

**return complex(b1.real - b2.real, b1.imaginary - b2.imaginary);**

**}**

**void show() {**

**cout << real << "+" << imaginary << "i" << endl;**

**}**

**};**

**int main() {**

**double realPart, imaginaryPart;**

**// Input for num5**

**cout << "Enter real part for num1: ";**

**cin >> realPart;**

**cout << "Enter imaginary part for num1: ";**

**cin >> imaginaryPart;**

**complex num5(realPart, imaginaryPart);**

**// Input for num6**

**cout << "Enter real part for num2: ";**

**cin >> realPart;**

**cout << "Enter imaginary part for num2: ";**

**cin >> imaginaryPart;**

**complex num6(realPart, imaginaryPart);**

**complex Mul = num5 \* num6;**

**cout << "Multiplication is: ";**

**Mul.show();**

**//complex Div = num5 / num6;**

**//cout << "Division is: ";**

**//Div.show();**

**// Addition**

**complex Add = num5 + num6;**

**cout << "Addition is: ";**

**Add.show();**

**// Subtraction**

**complex Sub = num5 - num6;**

**cout << "Subtraction is: ";**

**Sub.show();**

**// Division**

**complex num7(2, 3);**

**complex num8(4, 5);**

**complex Div = num7 / num8;**

**cout << "Division is: ";**

**Div.show();**

**return 0;**

**}**

Output-

