**Gr no 21810819**

**Assignment no: 2**

**Problem statement:**

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

Implement the following operations:

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

2. Overloaded operator + to add two complex numbers.

3. Overloaded operator \* to multiply two complex numbers.

4. Overloaded << and >> to print and read Complex Numbers.

**Aim of Assignment:**

To implement a Complex class and implement operations like create default constructor, and perform arithmetic operations on complex numbers using operator overloading.

**Description:**

One class named complex are used in which two constructors are created that is one is default constructor and another is parameterized constructor.

Arithmetic operations are perform by using operator overloading

Insertion and extraction operators are overloaded using istream and ostream class by declaring it as friend.

**OOP concept used:**

1. Constructor:

A constructor is a special function of a class which initialize objects of class.

Here, in this program contains two constructors : default & parameterized constructor.

2. Operator Overloading:

C++ provides a special function to change the current functionality of some operators within its class which is often called as operator overloading. Operator Overloading is the method by which we can change the function of some specific operators to do some different task.

**Code :**

#include <iostream>

using namespace std;

class complex //creation of class for complex numbers

{

private :

float real,imag;

public :

complex() //default constructor

{

real=0;

imag=0;

}

complex(float r,float i)

{

real=r;

imag=i;

}

friend ostream &operator<<(ostream &out,const complex &c) //overloaded operator <<

{

out << c.real;

out << "+i" << c.imag << endl;

return out;

}

friend istream &operator>>(istream &in, complex &c) //overloaded operator >>

{

cout << "Enter Real Part ";

in >> c.real;

cout << "Enter Imaginary Part ";

in >> c.imag;

return in;

}

complex setdata(float r,float i)

{

real=r;

imag=i;

}

complex operator +(complex c) //overloaded operator +

{

complex t;

t.real= real + c.real;

t.imag= imag + c.imag;

return t;

}

complex operator \*(complex c) //overloaded operator \*

{

complex t;

t.real= real\*c.real - imag\*c.imag;

t.imag= real\*c.imag + imag\*c.real;

return t;

}

};

int main()

{

cout<<" Complex Number";

cout<< endl;

complex c1,c2(1.5,2.5),c3,c4,c5,c6,c7; //object of class creation

cin>>c1;

c3 = c1+c2;

cout<<"Addition Of Complex Number is ";

cout<<c3;

cin>>c4;

c5.setdata(2.5,3.0); //calling of setdata() function

c6 = c4\*c5;

cout<<"Multiplication Of Complex Number is ";

cout<<c6;

c7 = c1+c2\*c3;

cout<<"Addition and Multiplication Of The Complex Number is ";

cout<<c7;

return 0;

}

**Conclusion:**

Various operations are implemented successfully on complex numbers.

**Screenshots:**

