**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 operator overloading using operators like +,\*,<< and >> and use the constructors.

**Description:** A class called complex is defined with two float member variables called real and imaginary. It also has two operator overloaded functions as its members to add and multiply complex numbers. Two friend functions are used to overload << and >> operator to accept and print the values. Friend function is used because it can be called without object.

**OOP Concept used:**

1. Operator Overloading:

Operator overloading is a compile-time polymorphism in which the operator is overloaded to provide the special meaning to the user-defined data type. Operator overloading is used to overload or redefines most of the operators available in C++. It is used to perform the operation on the user-defined data type. For example, C++ provides the ability to add the variables of the user-defined data type that is applied to the built-in data types.

Syntax:

return\_type class\_name  : : operator op(argument\_list)

{

     // body of the function.

}

Overloading insertion and extraction operator:

C++ is able to input and output the built-in data types using the stream extraction operator >> and the stream insertion operator <<. The stream insertion and stream extraction operators also can be overloaded to perform input and output for user-defined types like an object.

Here, it is important to make operator overloading function a friend of the class because it would be called without creating an object.

1. Constructor:

A constructor is a member function of a class which initializes objects of a class. In C++, Constructor is automatically called when object(instance of class) create. It is special member function of the class.

Default Constructor:

Default constructor is the constructor which doesn’t take any argument. It has no parameters.

1. Friend Function:

A friend function of a class is defined outside that class' scope but it has the right to access all private and protected members of the class. Even though the prototypes for friend functions appear in the class definition, friends are not member functions. Friend function doesn’t require class object to call. It can be called with functionname(agruments-list).

**Conclusion:** The concept of operator overloading and default constructor is implemented successfully.

Program:

#include<iostream>

using namespace std;

class complex

{

float a,b,c;

public:

complex()

{

a=0;

b=0;

}

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

{

cout<<"\n Enter Real part:";

in>>c.a;

cout<<"\n Enter Imaginary part:";

in>>c.b;

return in;

}

friend ostream &operator <<(ostream &out, complex &c)

{

out<<"\n"<<c.a<<"+"<<c.b<<"i";

return out;

}

complex operator+ (complex);

complex operator\* (complex);

};

complex complex::operator+ (complex c1)

{

complex t;

t.a=a+c1.a;

t.b=b+c1.b;

//cout<<"the sum is: "<<t.c;

return(t);

}

complex complex::operator\* (complex c1)

{

complex t;

t.a=a\*c1.a;

t.b=b\*c1.b;

//cout<<"the multiplication is"<<t.c;

return(t);

}

int main()

{

complex t1,r1,i,k;

cout<<"\n Enter 1st number:";

cin>>t1;

cout<<"\n Enter 2nd number:";

cin>>r1;

i=t1+r1;

k=t1\*r1;

cout<<"\n First No. is:";

cout<<t1;

cout<<"\n Second No. is:";

cout<<r1;

cout<<"addition: "<<i;

cout<<"multiplication: "<<k;

return 0;

}

