Assignment No : 4

Assignment Title: Write a C++ program to implement the concept of operator overloading and friend functions

Objective:

To study the concept of operator overloading and friend functions.

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 subtract two complex numbers 4. Overloaded operator* to multiply two complex numbers. 5. Overloaded << and >> to print and read Complex Numbers**.**

Software/ Hardware requirements:

1. **64 –bit open source Linux (Ubuntu** 12.04 64 bit)
2. **Open source C++ Programming tool (Eclipse 3.7.2 IDE)**

Theory:

Theory same as quadratic except complex number code example

```
+ operator overloading using member function
Complex Complex::operator+(Complex cNo2)
{
        Complex temp;
        temp.fReal = fReal + cNo2.fReal;
        temp.fImaginary = fImaginary + cNo2.fImaginary;
        return temp;
}
* operator overloading using member function
Complex Complex::operator*(Complex cNo2)
{
        Complex ret;
        ret.fReal = fReal*cNo2.fReal - fImaginary*cNo2.fImaginary;
        ret.fImaginary = fReal*cNo2.fImaginary +          fImaginary*cNo2.fReal;
        return ret;
}
<<  and >> operator is overloaded using friend function
It is written in the class as follows:-
friend ostream& operator<<(ostream&, Complex &);
friend istream& operator>>(istream&, Complex &);

ostream& operator<<(ostream& out, Complex &ret)
{
        out<< ret.fReal;
        cout<<" + ";
        out<< ret.fImaginary;
        cout<<'i';
```

```
	return out;
}
```
Similarly >> is defined as
```
istream& operator>>(istream& instr, Complex &ret)
{
	cout<<"Enter real part: ";
	instr>> ret.fReal;
	cout<<"Enter imaginary part: ";
	instr>> ret.fImaginary;
	cout<<ret<<endl;
	return instr;
}
```

Algorithm:

Step 1: Enter 2 complex numbers

Step 2: Display options “1. Add complex numbers, 2. Subtract complex numbers, 3. Multiply complex numbers”

Step 3: Check for input. If input ‘1’ goto step 4. if input ’2’ goto step 5. if input ’3’ goto step 6.

Step 4: Print sum of both complex numbers. Goto step 7

Step 5: Print difference of both complex numbers. Goto step 7

Step 6: Print product of both complex numbers. Goto step 7

Step 7: End

Input:

```
Enter a complex number:
Enter real part: 1
Enter imaginary part: 2
1 + 2i
Enter another number:
Enter real part: -1
Enter imaginary part: 2
-1 + 2i
```

Output:

```
1. Add the complex numbers
2. Subrtact the numbers
3. Multiply the complex numbers
Sum is: 0 + 4i
Do you want to continue ?(y)
y

1. Add the complex numbers
2. Subrtact the numbers
3. Multiply the complex numbers
3
Product is: -5 + 0i
```

Do you want to continue ?(y)
n

Conclusion:

Hence in this way we have implemented the complex number and studied the concepts of operator overloading and friend functions.