**\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Assignment No1\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

**Name: Aniket Gholap**

**Class: SE -I Div:A**

**Roll No:205A079**

**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.**

#include<iostream>

using namespace std;

class Complex { public:

float x; float y;

Complex()

{ x = 0; y = 0;

}

friend istream& operator>>(istream&, Complex&); friend ostream& operator<<(ostream&, const Complex&);

Complex operator+(const Complex&);

Complex operator\*(const Complex&);

};

Complex Complex::operator+(const Complex& c)

{

Complex add; add.x = x + c.x; add.y = y + c.y; return add;

}

Complex Complex::operator\*(const Complex& c)

{

Complex mul;

mul.x = (x \* c.x) - (y \* c.y);

mul.y = (y \* c.x) + (x \* c.y); return mul;

}

istream& operator>>(istream& in, Complex& t)

{

cout << "\n Enter the Real Part: "; in >> t.x;

cout << " Enter the Imaginary Part: ";

in >> t.y; return in;

}

ostream& operator<<(ostream& out, const Complex& t)

{ out << t.x; if (t.y >= 0)

out << " + " << t.y << "i"; else

out << " - " << -t.y << "i"; return out;

}

int main()

{

Complex c1, c2, c3, c4; cout << "\n Default Constructor: "; cout << c1;

cout << "\n Enter 1st complex number: "; cin>>c1;

cout<<"\n Enter 2nd complex number:"; cin>>c2; cout<<"\n 1st Complex Number:"<<c1; cout<<"\n 2nd Complex number:"<<c2;

c3=c1+c2;

cout<<"\n Addition of two complex number:"<<c3; c4=c1\*c2;

cout<<"\n Multiplication of two complex number:"<<c4; return 0;

}

**Output:**



