# TASK 3:

**HEADER FILE:**

#pragma once

#include<iostream>

using namespace std;

class Complex

{

float a;

float b;

public:

Complex();

Complex(float a, float b);

float mag(Complex c);

Complex add(Complex c);

Complex mul(Complex c);

Complex operator ++();

Complex operator ++(int);

Complex operator --();

Complex operator --(int);

friend ostream& operator <<(ostream& out, const Complex& c);

};

**FUNCTION FILE:**

#include"Header.h"

Complex::Complex()

{

a = 2;

b = 3;

}

Complex::Complex(float a, float b)

{

this->a = a;

this->b = b;

}

float Complex::mag(Complex c)

{

float temp;

temp = (c.a) \* (c.a) + (c.b) \* (c.b);

temp = sqrt(temp);

return temp;

}

Complex Complex::add(Complex c)

{

Complex temp;

temp.a = (c.a + this->a);

temp.b = (c.b + this->b);

return temp;

}

Complex Complex::mul(Complex c)

{

Complex temp;

temp.a = ((this->a \* c.a) - (this->b \* c.b));

temp.b = ((this->a \* c.b) + (c.a \* this->b));

return temp;

}

Complex Complex :: operator ++()

{

Complex temp;

temp.a = ++a;

temp.b = ++b;

return temp;

}

Complex Complex :: operator ++(int)

{

Complex temp;

temp.a = a++;

temp.b = b++;

return temp;

}

Complex Complex :: operator --()

{

Complex temp;

temp.a = --a;

temp.b = --b;

return temp;

}

Complex Complex :: operator --(int)

{

Complex temp;

temp.a = a--;

temp.b = b--;

return temp;

}

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

{

cout << "Complex No is: " << c.a << "+" << c.b << "i" << endl;

return out;

}

**SOURCE FILE:** #include"Header.h"

int main()

{

Complex c1;

cout << "Default "<< c1 << endl;

Complex c2(5, 8);

cout << "Overloaded " << c2 << endl;

cout << "Magnitude: " << c2.mag(c2) << endl;

cout << "Addition of " << c2.add(c2) << endl;

cout << "Multiplication of " << c2.mul(c2) << endl;

cout << "Pre Decrement of " << c2++ << endl;

cout << "Pre Increment of " << c2-- << endl;

}

# OUTPUT:

