## **Objectives**

After performing this lab, students shall be able to learn:

* This pointer
* Operator overloading
* Binary operator overloading
* Unary operator overloading
* Operator overloading as member functions.
* Operator overloading as non-member functions.
* Operator overloading as friend functions.
* Static data member and Static function

**TASK 1:**

Implement following **ComplexNumber** class and write driver program to test the implementation. Do not change the class definition in any way or form.

class ComplexNumber

{

private:

int real;

int imaginary;

static int count; //will count the total number of objects created

public:

ComplexNumber(int, int); //with default arguments

~ComplexNumber(); //Does Nothing.

void Input();

void Output();

static int countDisplay; //will return the total number of by incrementing as the object is created

bool IsEqual(ComplexNumber);

ComplexNumber Conjugate();

// Adding two complex numbers ( a + bi ) and ( c + di ) yields ( (a+b) + (c+d)i )

ComplexNumber operator+ (const ComplexNumber & num);

//Subtracting two complex numbers (a + bi) and (c + di) yields ((a-b) + (c-d)i).

ComplexNumber operator- (const ComplexNumber & num);

//Multiplying two complex numbers(a + bi)and(c + di) yields ((ac-bd) + (ad+bc)i).

ComplexNumber operator\* (const ComplexNumber & num);

//Increment and decrement operators should only add 1 or subtract 1 from real part

ComplexNumber & operator ++(); // pre-increment

ComplexNumber & operator --(); // pre-decrement

ComplexNumber operator ++(int); // post-increment

ComplexNumber operator --(int); // post-decrement

bool operator>=(const ComplexNumber& num);

bool operator<=(const ComplexNumber& num);

bool operator!=(const ComplexNumber& num);

};

**TASK 2:**

Implement a class called Quadratic. The class will have three data members:

* int a;                       // First part of quadratic equation
* int b;                         // Second part of the equation
* int c                         //Third part of quadratic equation.

//It'll form a number as ax²+bx+c

You have to implement default constructor, overloaded constructor, copy constructor, destructor and overload the operators + , - , \* , << , >> , ==, != , = as described below:

* **+**    =>   Add 2 quadratic objects (**Member as well as friend function)**
* **-**     =>   Subtract one quadratic object from other **(Member as well as friend function)**
* **\***     =>   Multiply a constant with Quadratic object **(Member Function only)**
* **>>**   =>   Input a quadratic object  **(Friend Function)**
* **<<**   =>   Output a quadratic object (**Friend Function)**
* **==**   =>   Equality Operator **(Member Function only)**
* **!=**   =>   In-equality operator **(Member Function only)**
* =   =>   Assignment operator **(Member Function only)**

**Note**

* Follow all the code indentation, naming conventions and code commenting guidelines.