Name: Simoni Patani

Class: SY Comp A

Roll No: 221039

GR No: 21810036

**Assignment 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 the Assignment:** In this assignment, we have to implement the concepts of operator overloading to add and multiply two given complex numbers. We also need to overload ‘<<’ and ‘>>’ operators to print and read the complex numbers respectively.

# **Description:**  First, we declare the real and imaginary parts of the complex number as floating point variables. The default constructor follows, in which both the variables are assigned an initial value of 0.0. Next, the ‘+’ and ‘\*’ operators are overloaded to perform addition and multiplication operations, and the ‘<<’ and ‘>>’ operators are overloaded for extraction and insertion respectively. The values are accepted, and the mathematical logic for addition and multiplication of complex numbers follows. Finally, in the main() method, values are accepted, mathematical operations are performed, and appropriate values are displayed.

**Algorithm**:

Step1: Start

Step2: Create Class complex and declare variables, constructor and operator overloading functions.

Step3: Define the overloading functions.

Step4: In main accept the numbers and perform the operations on them.

Step5: Display the addition and multiplication.

Step6:End

**Concepts Used**:

In order to solve the problem statement the concepts used were

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 an 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 can also 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 an object(instance of class) create. It is a special member function of the class.

Default Constructor:

Default constructor is a constructor which doesn’t take any arguments. 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 function name(arguments-list).

**Conclusion:** Hence, we have seen how inbuilt operators like ‘+’ and ‘\*’ can be overloaded to give them new meanings and extend their functionality. In this program, we have done this by using the aforementioned operators to add and multiply complex numbers.