**Assignment No: 1**

**Title:** Arithmetic operations on complex numbers using operator overloading.

**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**

**Prerequisites:**

Object Oriented Programming

**Objectives:**

To learn the concept of constructor, default constructor, operator overloading using member function and friend function.

**Theory:**

Operator Overloading

It is a specific case of polymorphism where different operators have different implementations depending on their arguments. In C++ the overloading principle applies not only to functions, but to operators too. That is, of operators can be extended to work not just with built-in types but also classes. A programmer can provide his or her own operator to a class by overloading the built-in operator to perform some specific computation when the operator is used on objects of that class.

An Example of Operator Overloading

Complex a(1.2,1.3); //this class is used to represent complex numbers

Complex b(2.1,3); //notice the construction taking 2 parameters for the real and imaginary part

Complex c = a+b; //for this to work the addition operator must be overloaded

Arithmetic Operators

Arithmetic Operators are used to do basic arithmetic operations like addition, subtraction, multiplication, division, and modulus.

With C++ feature to overload operators, we can design classes able to perform operations using standard operators. Here is a list of all the operators that can be overloaded:

|  |
| --- |
| **Over loadable operators** |
| + - \* / = <> += -= \*= /= <<>>  <<=  >>=  ==   !=   <=   >=   ++  --   % & ^ ! |  ~ &= ^= |= && || %= [] |

* To overload an operator in order to use it with classes we declare *operator functions*, which are regular functions whose names are the operator keyword followed by the operator sign that we want to overload. The format is:
* type operator operator-symbol (parameters) {/\*...\*/ }
* The **operator** keyword declares a function specifying what *operator-symbol* means when applied to instances of a class. This gives the operator more than one meaning, or "overloads" it. The compiler distinguishes between the different meanings of an operator by examining the types of its operands.

**Syntax:**

**return\_typeclass\_name :: operator op(arg\_list)**

**{**

**//function body**

**}**

where,

* Return type is the value returned by the specified operation
* op is the operator to be overload.
* op is proceeding by the keyword operator.
* operator op is the function name

**Process of the overloading has 3 steps**

1. Create a class that define a data types that is used in the overloading operation
2. Declare the operator function operator op() in the public part of the class. It may be either a member function or a friend function.
3. Define the operator function to implement the required operation

*e.g.*

**Overloading Binary operators:**

A statement like

C = sum (A, B); // functional notation

This functional notation can be replaced by a natural looking expression

|  |
| --- |
| C = A+B; // arithmetic notation  by overloading the + operator using an operator+ () function. |
| **Algorithm:** |
| Step 1: Start the program  Step 2: Create a class complex  Step 3: Define the default constructor.  Step 4: Declare the operator function which are going to be overloaded and display function  Step 5: Define the overloaded functions such as +, -,/,\* and the display function  For Addition:  (a+bi) + (x + yi) = ((a+x)+(b+y)i) For Multiplication:  (a+bi) \* (x + yi) = (((a\*x)-(b\*y)) + ((a\*y) + (x\*b))i)  Step 6: Create objects for complex class in main() function  Step 7:Create a menu for addition, multiplication of complex numbers and display the result  Step 8: Depending upon the choice from the user the arithmetic operators will invoke the overloaded operator automatically and returns the result  Step 9: Display the result using display function. |

**Input**

Complex numbers with real and imaginary values for two complex numbers.

Example :

Complex No 1:  Real Part :   5

Imaginary part : 4 Complex No 2: Real Part :   5

|  |
| --- |
| Imaginary part :   4 |
| **Output:** |
| Default constructor value=0+0i  Enter the 1st number Enter the real part2  Enter the imaginary part4  Enter the 2nd number Enter the real part4  Enter the imaginary part8  The first number is 2+4i The second number is 4+8i  The addition is 6+12i  The multiplication is -24+32i |
| **Conclusion:** |
| Hence, we have studied concept of operator overloading. |
| **Important Questions:** |
| 1. What is operator overloading? 2. What are the rules for overloading the operators? 3. State clearly which operators are overloaded and which operator are not overloaded? 4. State the need for overloading the operators. 5. Explain how the operators are overloaded using the friend function. 6. What is the difference between “overloading” and “overriding”? 7. What is operator function? Describe the syntax? 8. When is Friend function compulsory? Give an example? |