

Title: Demonstrate use of operator overloading for complex numbers.

Objectives:

- 1) To understand overloading.

- 2) To demonstrate overloading of binary, insertion & extract operators.

Problem Statement: Implement a class complex which represents the complex number data type-

- 1) Constructor (which includes $0 + 0i$)

- 2) Overloaded operator +.

- 3) Overloaded operator x.

- 4) Overloaded operators << & >>.

System & Hardware req: 64 bit OS C++ programming tool G++.

Theory: C++ allows you to specify more than one ^{definition for a} function name or an operator in the same scope, which is called function overloading & operator overloading respectively. An overloaded declaration is a declaration that had been declared with same name as a previously declared declaration in same scope, except that both declaration have different argument & obviously different definition. The process of selecting the most appropriate overloaded function operator is called operator ~~overloading~~ resolution.

Algorithm:

1. Start
2. Declare 4 objects c_1, c_2, c_3, c_4 of complex class
3. Input c_1, c_2 using '>>' overloaded extraction operator.
4. Set $c_3 = c_1 + c_2$ using overloaded '+' operator.
5. Print c_3 using overloaded '<<' insertion operator.
6. Set $c_4 = c_1 * c_2$ using overloaded '*' operator.

- 7 print $c4$ using overloaded ' $<<$ ' insertion operator.
 8. End.

Test cases

Test case	Description	Input	Exp O/P	Actual O/P	Result
1	Addition of two complex numbers	$c1 + c2$	$c1 + c2$	$2+3i + 1+4i = 3+7i$	Pass
2	Numeric input	5	5	5	Pass
3	Using default value				Pass
4	Invalid input & >>ab	invalid out	invalid	invalid Comp No	Pass
5	Invalid input & >>b	"	"	"	Pass
6	Invalid choice of no. 5 (all invalid)	choice	choice	choice	Pass
7	Addition	1+1i	1+1i	1+1i	Pass
8	Multiplication	0+i0	0+i0	0+i0	Pass
9	Multiplication	-23-i38	-23-i38	-23-i38	Pass

CONCLUSION:

1. Learnt to implement class & use of default constructor.
2. Learnt about operator overloading & friend class.