

OOP Lab Assignment 2

Page No.: |

Jaynam Modi. PG-43. G3. Aug 05, 2020

- Problem Statement: Define a class Complex consisting following:
 - Data members:
 1. Real
 2. Imaginary part
 - Member Functions:
 1. One default constructor
 2. Function setComplex () to set the value of real and imaginary part.
 3. Function printComplex () to display
- Three overloaded operator member functions

1. Operator + to add two complex numbers.

2. Overloaded << and >> to print and read Complex Numbers.

- Friend function to add two complex number by taking two reference variables of class complex and returning another reference.

- Objectives:

1. To learn to create classes in C++
2. To learn constructor, function and operator overloading in C++
3. To learn friend function in C++

- Theory:

- Operator Overloading (Unary operator & Binary Operator)

C++ provides a special function to change the current functionality of some operators within its class which is often called as operator overloading. Operator Overloading is the method by which we can change the function of some specific operators to do some different task.

In unary operator function, no arguments should be passed. It works only with one class objects. It is a overloading of an operator operating on a single operand.

In binary operator overloading function, there should be one argument to be passed. It is overloading of an operator operating on two operands.

- 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.

- Algorithm / Class Diagram / Implementation:

1. Take Input from User.
2. Process the Input to form a Complex number as defined in the Complex Class.
3. Perform Addition and Multiplication with Given Input.
4. Display Results.

- Platform: 64 bit Open source Linux
- Input: Any 2 complex numbers: $(6+4i, 3+2i)$
- Output: Sum $(9+6i)$ and Product $(10+24i)$ of the 2 input complex numbers
- Conclusion: Hence, learned to create classes, friend function and performed operator overloading in C++ successfully.
- FAQs:

1. What is inline function?

> Inline function is powerful concept that is commonly used with classes. If a

function is inline, the compiler places a copy of the code of that function at each point where the function is called at compile time.

Any change to an inline function could require all clients of the function to be recompiled because compiler would need to replace all the code once again otherwise it will continue with old functionality.

To inline a function, place the keyword inline before the function name and define the function before any calls are made to the function. The compiler can ignore the inline qualifier in case defined function is more than a line.

2. Which operator links class to a member?

> Dot (.) operator is known as "Class Member Access Operator" in C++ programming language, it is used to access public members of a class. Public members contain data members (variables) and member functions (class methods) of a class.

3. What is default access specifier in C++?

- > By default access to members of a C++ class is private. The private members are not accessible outside the class; they can be accessed only through methods of the class.


```

1  #include <iostream>
2
3  using namespace std;
4
5  class Complex{
6
7      public:
8          int r;
9          int i;
10
11         void setComplex(int a, int b){
12             r = a;
13             i = b;
14         }
15         void printComplex(){
16             cout << r << " + " << i << "i" << endl;;
17         }
18
19         Complex operator + (Complex c2){
20             Complex result;
21             result.r = r + c2.r;
22             result.i = i + c2.i;
23             //result.printComplex();
24             return result;
25         }
26
27         friend ostream &operator<<(ostream &output,
Complex &c){
28             output << c.r << " + " << c.i << "i";
29             return output;
30         }
31
32         friend istream &operator>>(istream &input, Complex
&c){
33             input >> c.r >> c.i;
34             return input;
35         }
36
37         friend Complex multiply(Complex * , Complex *);
38 };
39
40 Complex * multiply(Complex &c1, Complex &c2){
41     Complex result;
42     result.r = (c1.r * c2.r) - (c1.i * c2.i);
43     result.i = (c1.r * c2.i) + (c2.r * c1.i);
44     return &result;
45 }
46
47
48 int main(){
49     Complex c;
50     c.setComplex(5,4);
51     c.printComplex();
52
53     Complex c1;
54     cin >> c1;
55     cout << c1 << endl;
56
57     Complex c2;
58     cin >> c2;
59     cout << c2 << endl;
60
61     Complex c3 = c1 + c2;
62
63     cout << c3 << endl;
64
65     Complex c4 = *multiply(c1, c2);
66     cout << c4 << endl;
67 }

```

```
u0_a362@localhost:~$ g++ oop_assignment2.cpp
oop_assignment2.cpp:44:10: warning: address of stack memory associated with local
    variable 'result' returned [-Wreturn-stack-address]
    return &result;
           ^~~~~~
```

1 warning generated.

```
u0_a362@localhost:~$ ./a.out
```

5 + 4i

6

4

6 + 4i

3

2

3 + 2i

9 + 6i

10 + 24i

```
u0_a362@localhost:~$ █
```