

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Lab Number:	5
Student Name:	Deep patel
Roll No :	8

Title:

To perform Operator Overloading using C++ for

- adding 2 complex numbers
- adding matrices

Learning Objective:

- Students will be able to perform user-defined overloading of built-in operators.

Learning Outcome:

- Understanding the overloading concept on built-in operators.

Course Outcome:

ECL304.2	Comprehend building blocks of OOPs language, inheritance, package and interfaces
-----------------	--

Theory:

Explain about operator overloading with respect to:

1. Constructor:

Constructors can be overloaded in a similar way as function overloading. Overloaded constructors have the same name as that of the class but the different number of arguments. Depending upon the number and type of arguments passed, the corresponding constructor will be called at that time.

A constructor is called depending upon the number and type of arguments passed. While creating the object, arguments must be passed to let compiler know, which constructor needs to be called.

2. Methods:

Overloading by different methods is the process of having two or more function with the same name, but different in parameters in C++. In this, the function is redefined by using either different types of arguments or a different number of arguments. It is only through these differences compiler can differentiate between the functions.

3. Operators:

Faculty: Ms. Deepali Kayande

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

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.

Types of Operator overloading:

1. Operator Overloading in Unary Operators
2. Operator Overloading in Binary Operators:

PROGRAM 1: To perform Operator Overloading using C++ for Multiplying 2 complex numbers.

Algorithm:

Step 1: START

Step 2: Create class complex

Step 3: Create multiply function to multiply two complex numbers

Step 4: Overloading increment operator to increment complex number

Step 5: Creating objects for two complex numbers and taking values from the user and printing it.

Step 6: Printing the output as multiplication of entered numbers

Step 7: EXIT.

Program:

```
//To perform Operator Overloading using C++ for multiplying 2 complex numbers
```

```
#include <iostream>
```

```
using namespace std;
```

```
class Complex
```

```
{
```

```
    public:
```

Faculty: Ms. Deepali Kayande

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

```
int real;
```

```
int imag;
```

```
Complex(){
```

```
    real=0;
```

```
    imag=0;
```

```
}
```

```
Complex(int r, int i)
```

```
{
```

```
    real=r;
```

```
    imag=i;
```

```
}
```

```
Complex operator*(Complex c){
```

```
    double real1,real2;
```

```
    real1=real;
```

```
    real2=c.real;
```

```
    Complex ans;
```

```
    real=(real*c.real)-(imag*c.imag);
```

```
    imag=(real1*c.imag)+(imag*real2);
```

```
    //if (a,b)(c,d) then formula of multiplication is (ac-bd,ad+bc)
```

```
    ans.real=real;
```

```
    ans.imag=imag;
```

```
    return(ans);
```

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

```
    }

    void display()
    {
        cout<<real<<"*"<<imag<<"i"<<endl;
    }

};

int main()
{
    Complex c1(3,2);
    Complex c2(1,7);
    Complex c3;
    c3=c1*c2;
    c3.display();
}
```

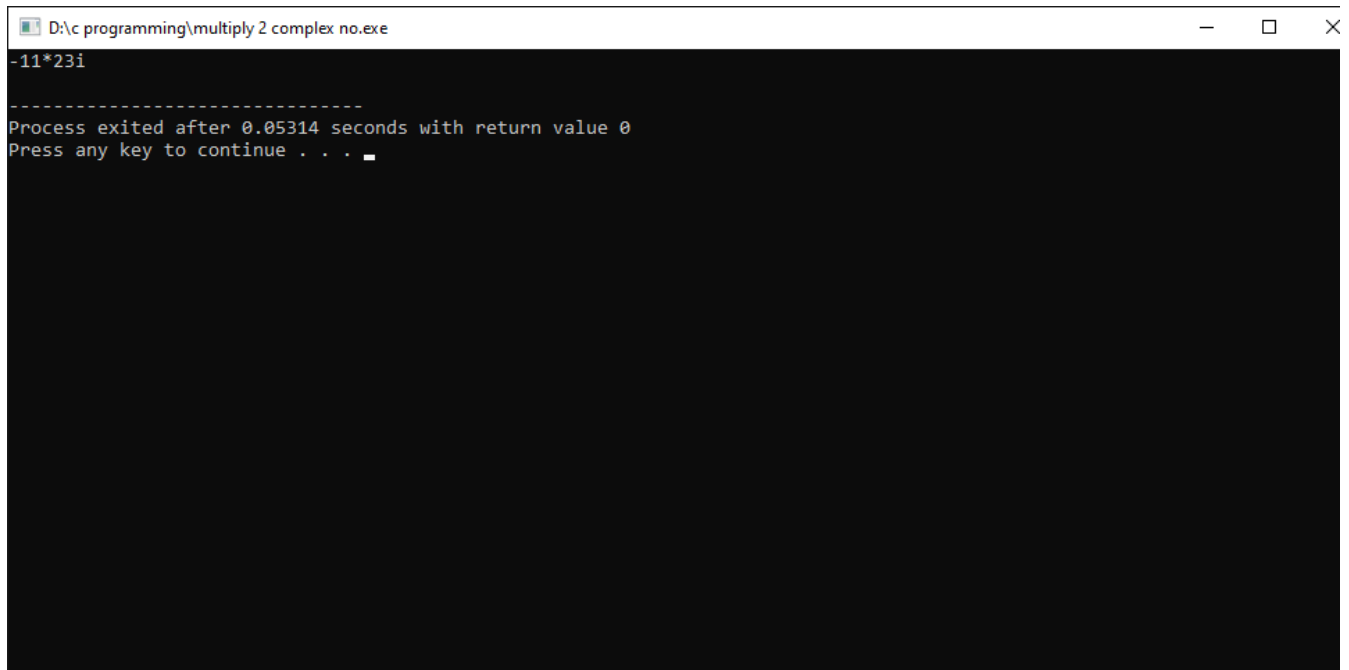
Input Given:

3 + 2i

1 + 7i

Output:

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22



The screenshot shows a Windows command prompt window titled "D:\c programming\multiply 2 complex no.exe". The output of the program is as follows:

```
-11*23i
-----
Process exited after 0.05314 seconds with return value 0
Press any key to continue . . .
```

PROGRAM 2: To perform Operator Overloading using C++ for Adding two matrices.

Algorithm:

Step 1: START

Step 2: Create class matrices

Step 3: Create 3 matrices for 2*2 matrix

Step 4: Creating get_elements() function, operator overloading, print the result as display function.

Step 5: Create functions outside class, using scope resolution.

Step 6: Creating objects, displaying matrices, and printing the result as addition of two matrices

Step 7: EXIT

Faculty: Ms. Deepali Kayande

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

Program:

//Write a C++ program to overload the '+' operator so that it can add two matrices.

```
#include<iostream>
```

```
using namespace std;
```

```
class matrices
```

```
{
```

```
    int a[2][2];
```

```
    int b[2][2];
```

```
    int c[2][2];
```

```
    public:
```

```
        void get_elements(); //take numbers from user
```

```
        matrices operator +(matrices m2); //operator overloading
```

```
        void display(); //print the result
```

```
};
```

```
//functions outside class, using scope resolution
```

```
void matrices::get_elements()
```

```
{
```

```
    cout<<"enter the elements";
```

```
    for(int i=0;i<2;i++) //for row
```

```
    {
```

```
        for(int j=0;j<2;j++) //for columns
```

```
            cin>>a[i][j];
```

```
    }
```

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

```
}
```

```
void matrices:: display()
```

```
{
```

```
    for(int i=0;i<2;i++)
```

```
    {
```

```
        for(int j=0;j<2;j++)
```

```
            cout<<a[i][j]<<" ";
```

```
            cout<<endl;
```

```
    }
```

```
}
```

```
matrices matrices::operator+(matrices m2)
```

```
{
```

```
    matrices m3;
```

```
    for(int i=0;i<2;i++)
```

```
    {
```

```
        for(int j=0;j<2;j++)
```

```
            m3.a[i][j]=a[i][j]+m2.a[i][j];
```

```
    }
```

```
    return(m3);
```

```
}
```

```
int main()
```

```
{
```

```
    matrices ob1,ob2;
```

```
    ob1.get_elements();
```

```
    ob2.get_elements();
```

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication Engineering
ECL304 - Skill Lab: C++ and Java Programming
Sem III
2021-22

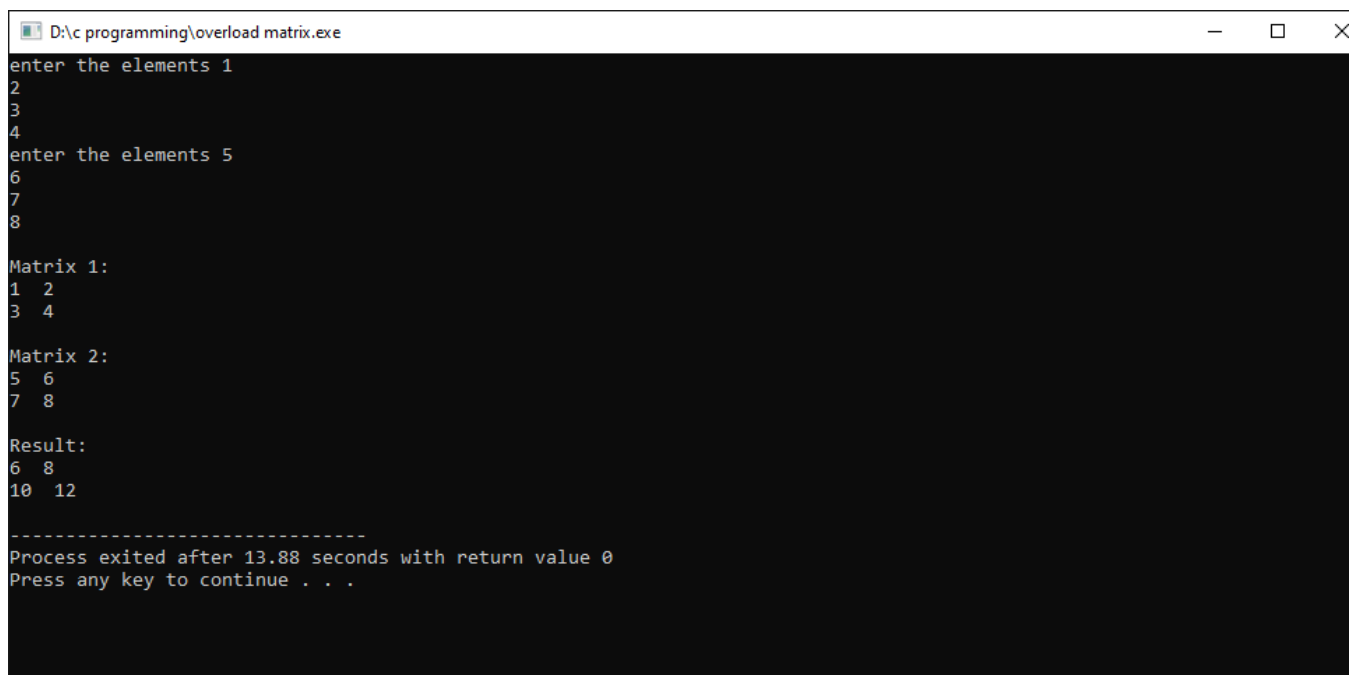
```
cout<<"\nMatrix 1:\n";  
  
ob1.display();  
  
cout<<"\nMatrix 2:\n";  
  
ob2.display();  
  
ob1=ob1+ob2;  
  
cout<<"\nResult:\n";  
  
ob1.display();  
  
}
```

Input Given:

1st Matrix: 1,2,3,4

2ND Matrix: 5,6,7,8

Output:



The screenshot shows a Windows command prompt window titled "D:\c programming\overload matrix.exe". The user enters the elements of two 2x2 matrices. The first matrix is 1 2 3 4, and the second matrix is 5 6 7 8. The program then displays the two matrices and the result of their addition, which is 6 8 10 12. The window also shows the process exit time and a prompt to press any key to continue.

```
D:\c programming\overload matrix.exe  
enter the elements 1  
2  
3  
4  
enter the elements 5  
6  
7  
8  
  
Matrix 1:  
1 2  
3 4  
  
Matrix 2:  
5 6  
7 8  
  
Result:  
6 8  
10 12  
  
-----  
Process exited after 13.88 seconds with return value 0  
Press any key to continue . . .
```