

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication
Engineering ECL304 - Skill Lab: C++ and Java
Programming Sem 111 2021-22

Lab Number:	5
Student Name:	Disha Maggu
Roll No :	09

Title:

To perform Operator Overloading using C++ for

- Multiplying 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:

- Constructor : Constructor overloading is a concept in which one class can have multiple constructors with different parameters. The main thing to note here is that the constructors will run according to the arguments for example if a program consists of 3 constructors with 0, 1, and 2 arguments, so if we pass 1 argument to the constructor the compiler will automatically run the constructor which is taking 1 argument.
- methods : Method overloading is the process of overloading the method that has the same name but different parameters. C++ provides this method of overloading features. Method overloading allows users to use the same name to another method, but the parameters passed to the methods should be different. The return type of methods can be the same or different.
- Operators : In C++, it can add special features to the functionality and behaviour of already existing operators like arithmetic and other operations. The mechanism of giving special meaning to an operator is known as operator overloading. For example, we can overload an operator '+' in a class like string to concatenate two strings by just using +.

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 111 2021-22

Algorithm:

1-Start

2-Creating class of name complex

3-Declaring attributes- real , img

4-Declaring methods- 1)get_elements()-to take input from user

2)display()- to print the result

5-Operator overloading function to overload "*"+"----for performing operation

6-Defining methods outside the class

7-Creating an objects of class in main function 8-Calling

the methods using object of class

9-Displaying the result

10-End

Program :

```
# include<iostream>
using namespace std;
class complex
{
    float real;
    float img;
    public:
    void get_elements();           //take numbers from user
        complex operator *(complex c1); //operator overloading
        void display();           //print the result
};

void complex::get_elements()
{
    cout<<"Enter the real and img of complex no.\n";
    cout<<"Real :";
    cin>>real;
    cout<<"Img :";
```

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication
Engineering ECL304 - Skill Lab: C++ and Java
Programming Sem 111 2021-22

```
cin>>img;
}

void complex::display()
{
cout<<"("<<real<<")"<<"+"<<"("<<img<<")"<<"i";
}

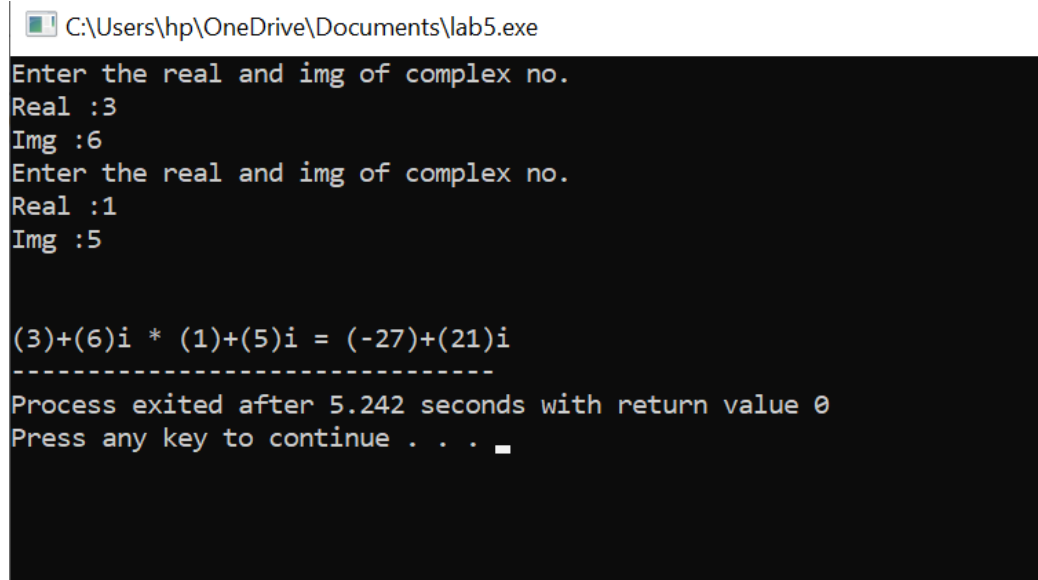
complex complex::operator*(complex c1)
{
complex mul;
mul.real = ((real*c1.real)-(img*c1.img));
mul.img = ((real*c1.img)+(c1.real*img));
return(mul);
}

int main()
{
complex obj1,obj2,obj3;
obj1.get_elements();
obj2.get_elements();
obj3= obj1*obj2;

cout<<"\n\n";
obj1.display();
cout<<" * ";
obj2.display();
cout<<" = ";
obj3.display();
}
```

Don Bosco Institute of Technology, Kurla(W)
Department of Electronics and Tele-Communication
Engineering ECL304 - Skill Lab: C++ and Java
Programming Sem 111 2021-22

Output :



```
C:\Users\hp\OneDrive\Documents\lab5.exe
Enter the real and img of complex no.
Real :3
Img :6
Enter the real and img of complex no.
Real :1
Img :5

(3)+(6)i * (1)+(5)i = (-27)+(21)i
-----
Process exited after 5.242 seconds with return value 0
Press any key to continue . . .
```

Algorithm :

1-Start

2-Creating class of name matrices

3-Declaring $a[2][2], b[2][2], c[2][2]$

4-Declaring methods- 1)get_elements()-to take input from user

2)display()- to print the result

5-Operator overloading function to overload "+"----for performing operation

7-Creating an objects of class in main function

8-Calling the methods using object of class

9-Displaying the result

10-End

Program :

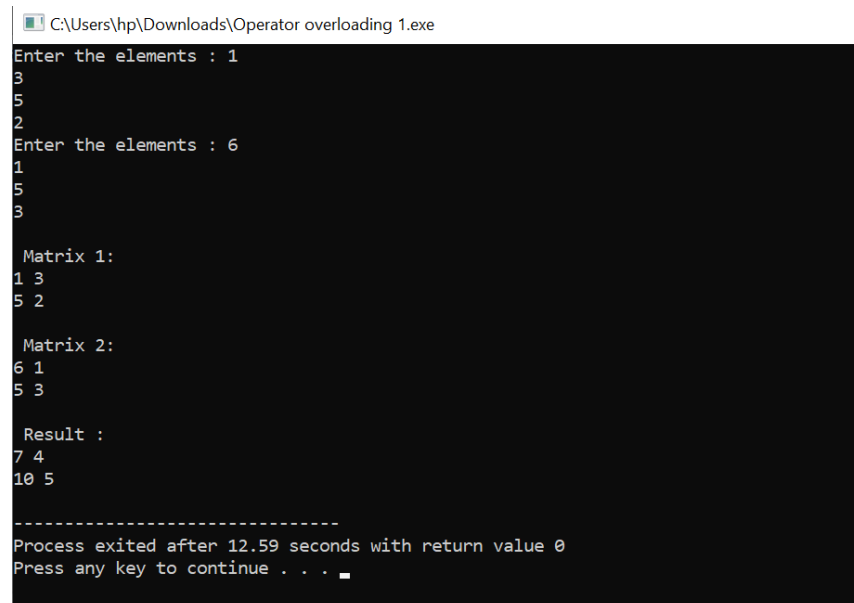
```
#include<iostream>
using namespace std;

class matrices
{
public:
int a[2][2];
int b[2][2];
int c[2][2];
void get_elements()
{
    cout<<"Enter the elements : ";
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        {
            cin>>a[i][j];
        }
    }
}
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);
}
void display()
{
    for(int i=0;i<2;i++)
    {
        for(int j=0;j<2;j++)
        {
            cout<<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 111 2021-22

```
        }  
        cout<<endl;  
    }  
    }  
};  
  
int main()  
{  
    matrices ob1,ob2;  
    ob1.get_elements();  
    ob2.get_elements();  
    cout<<"\n Matrix 1:\n";  
    ob1.display();  
    cout<<"\n Matrix 2:\n";  
    ob2.display();  
    ob1=ob1+ob2;  
    cout<<"\n Result : \n";  
    ob1.display();  
  
}
```

Output :



The screenshot shows a Windows command prompt window titled "C:\Users\hp\Downloads\Operator overloading 1.exe". The program prompts the user to enter elements for two 2x2 matrices. For Matrix 1, the inputs are 1, 3, 5, and 2. For Matrix 2, the inputs are 6, 1, 5, and 3. The program then displays the two matrices, adds them, and shows the resulting matrix. The output is as follows:

```
C:\Users\hp\Downloads\Operator overloading 1.exe  
Enter the elements : 1  
3  
5  
2  
Enter the elements : 6  
1  
5  
3  
  
Matrix 1:  
1 3  
5 2  
  
Matrix 2:  
6 1  
5 3  
  
Result :  
7 4  
10 5  
  
-----  
Process exited after 12.59 seconds with return value 0  
Press any key to continue . . .
```