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:	Aniket shrimant pawar
Roll No :	04

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:

- constructor,
- methods and
- operators.

Algorithm :	1) Start 2) Create class matrices 3) Take input from user for matrix 1and matrix 2 4) Matrix 3=matrix 1+matrix 2 5) Create obj1, obj2 for matrix1, matrix2 6) Print obj1, obj2 7) obj1=obj1+obj2 8) Print obj1
Program:	#include<iostream>

Faculty: Ms. Deepali Kayande

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

	ob1.display();
	}

Input given:	Matrix 1-8, 8,9,4 Matrix 2-7, 8,3,4
Output Screenshot:	**Compile Result** enter the elements8 8 9 4 enter the elements7 8 3 4 Matrix 1: 8 8 9 4 Matrix 2: 7 8 3 4 Result: 15 16 12 8

- Adding 2 complex numbers

Program- #include<iostream>

using namespace std ;

class complexno

```
{
    public :

            int real, imag;
            complexno()
            {
                    real = 0;
                    imag = 0;

            }
            complexno(int r, int i)
            {
                    real = r;
                    imag = i;
            }
            void display()
            {
                    cout << real << "*" << imag << "i" << endl;
              }
            complexno operator *(complexno c)
            {
                    complexno temp;
                    temp.real = real * c.real;
                    temp.imag =   imag * c.imag;

                    return temp;
            }
};
```

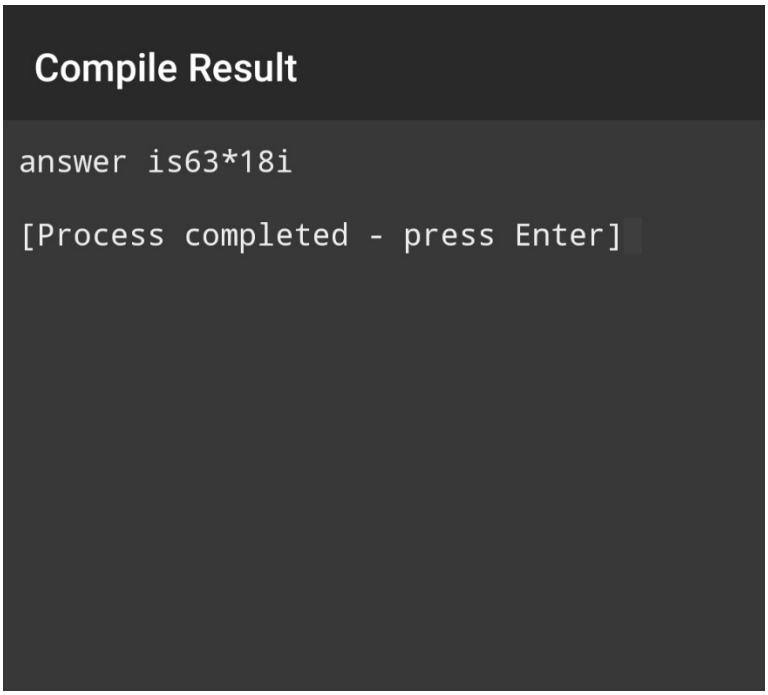
```
int main()
{
    complexno c3;
    complexno c1(7,3);
    complexno c2(9,6);
    c3 = c1*c2;
cout<<"answer is";
    c3.display();
    return 0;
}
```

Input given:-

C1=7+3i

C2=9+6i

Output-


Compile Result
answer is63*18i
[Process completed - press Enter]