

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:	Yash J Sonavane
Roll No :	45

Title:

To perform Operator Overloading using C++ for

- adding 2 complex numbers
- adding matrices
- Students will be able to perform user-defined overloading of built-in operators.

Learning Objective:

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.**

Operator overloading provides the ability to a mathematical operator to perform other manipulation operations other than its original operation . Operator overloading is one of the most valuable concepts introduced by C++ language . C++ frequently uses user-defined data types such as classes and structures that are a combination of one or more basic data types. User-defined data types created from class or structure are nothing but a combination of one or more variables of basic data types. The compiler knows how to perform various operations using operators for the built-in types; however, for the objects those are instance of the class, the operation routine must be defined by the programmer.

For example, in traditional programming languages the operators such as +, -, <=, >=, etc. can be used only with basic data types such as int or float. The operator + (plus) can be used to perform addition of two variables, but the same is not applicable for objects. The compiler cannot perform addition of two objects. The compiler would throw an error if addition of two objects is carried out. The compiler must be made aware of the addition process of two objects. When an expression including operation with objects is encountered, a compiler searches for the definition of the operator, in which a code is written to perform an operation

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

with two objects. Thus, to perform an operation with objects we need to redefine the definition of various operators. For example, for addition of objects A and B, we need to define operator + (plus). Redefining the operator plus does not change its natural meaning. It can be used for both variables of built-in data type and objects of user-defined data type and this is called as operator overloading .

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.

Overloaded constructors have the same name (name of the class) but the different number of arguments. Depending upon the number and type of arguments passed, the corresponding constructor is called.

Algorithm :	<ol style="list-style-type: none">1. Start2. Define functions for get_element(), display(), and overload the '+' operator.3. Take user input for matrices.4. Decide on two variables of the Matrix type.5. Use the get_element() function to receive the matrix 6. Use the display_element() function to display the matrices.7. Add them using the overloaded '+' operator.8. Print the result.
Program:	https://github.com/yashsonavane/skill-labs-with-oopm/commit/0acbfc056862ba80552cb1b8b4e460ab544cc4a6
Input given:	M1=5432 M2=16810

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

Output Screenshot:	<pre> enter the elements5 4 3 2 enter the elements1 6 8 10 Matrix 1: 5 4 3 2 Matrix 2: 1 6 8 10 Result: 6 10 11 12 ----- Process exited after 19.72 seconds with return value 0 Press any key to continue . . . </pre>
---------------------------	---

Algorithm:	<p>Step 1 - Start</p> <p>Step 2 - Int real , imag</p> <p>Step 3 - real = r , imag = i</p> <p>Step 4 - Define overload * operator</p> <p>Step 5 - Use temporary variable with name 'temp'</p> <p>Step 6 - temp.real = real*c.real temp.imag = imag*c.imag</p> <p>Step 7 = Enter the values of complexNo</p> <p>STEP 8 = Multiply them using overload * operator</p> <p>Step 9 = Display the caluculated value</p> <p>Step 10 = Stop</p>
-------------------	--

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:	https://github.com/yashsonavane/skill-labs-with-oopm/commit/0acbfc056862ba80552cb1b8b4e460ab544cc4a6
Input given:	8,3 5,6
Output Screenshot:	