

DEPARTMENT OF COMPUTER SCIENCE
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY
M.Sc. (Five year integrated) in Computer Science (AI & DS)
21-805-0107: Lab 2 – C++ Programming Lab

LAB EXPERIMENTS – SET 2

1. Write a C++ program to overload ==, !=, <, <=, > and >= operators as member operator functions for a **vector** object.
2. Write a C++ program to design a class representing **complex** numbers and having the functionality of performing addition & multiplication of two complex numbers using operator overloading (Use friend operator functions).
3. Write a C++ program to overload operators like *, <<, >> using friend function. The following overloaded operators should work for a class **vector**.
4. Write a C++ program for developing a **matrix** class which can handle integer matrices of different dimensions. Also overload the operator for addition and multiplication of matrices. Use double pointers in your program to dynamically allocate memory for the matrices.
5. Write a C++ program to demonstrate the concept of Multiple and Multilevel inheritance including constructors with parameters.
6. Write a C++ program to design a **student** class representing student roll no. and a **test** class (derived class of student) representing the scores of the student in various subjects and **sports** class representing the score in sports. The sports and test class should be inherited by a **result** class having the functionality to add the scores and display the final result for a student. Demonstrate the concept of Virtual base class on Hybrid inheritance.
7. Write a C++ program illustrating how the constructors are implemented and the order in which they are called when the classes are inherited. Use three classes named **alpha**, **beta** and **gamma** such that alpha and beta are base classes and gamma is a derived class inheriting alpha & beta.
8. Write a C++ program to define classes **Shapes**, **Circle**, **Square**, **Ellipse** and **Rectangle** with member functions to get the values for finding corresponding areas and print the same. Utilize the concept of **Abstract Class** and **Runtime polymorphism** to solve the problem.