Lab IA – 01

1	Design a class hierarchy for a Library Management System with base class `Item` and derived classes `Book`, `Journal`, and `DVD`. Include virtual functions for displaying details and demonstrate polymorphism.
2	Implement a class `FileHandler` that opens a file in the constructor and closes it in the destructor. Demonstrate RAII by writing and reading data from the file.
3	Create a class `Student` where a static data member keeps track of the highest marks scored among all students. Use static member functions to update and display it.
4	Create two classes `Number` and `Roman`. Store decimal value in `Number` and Roman numeral string in `Roman`. Write a friend function `convert(Number, Roman&)` to perform Decimal-to-Roman conversion.
5	Define a class `Matrix` and overload operators `+`, `-`, and `*` to perform matrix addition, subtraction, and multiplication. Demonstrate with user input.
6	Create a class `Shape` with pure virtual function `area()`. Derive classes `Circle`, `Rectangle`, and `Triangle` implementing their own area calculations. Demonstrate runtime polymorphism.
7	Design a class `Matrix` with dynamic memory allocation for 2D arrays. Implement constructor for initialization, destructor for cleanup, and copy constructor for deep copy.
8	Design a class `Logger` with a static counter to track number of log messages created. Provide a static function to retrieve total count.
9	Write a program with two classes `Polynomial` and `Equation`. Use a friend function to evaluate the polynomial expression from private coefficients stored in `Polynomial`.
10	Create a class `BigInteger` to represent very large numbers using strings. Overload `+` and `*` operators to perform arithmetic on them.
11	Design a class `BankAccount` with deposit, withdraw, and balance inquiry methods. Derive classes `SavingsAccount` and `CurrentAccount` with specific rules (e.g., minimum balance, overdraft). Use virtual functions.
12	Write a program to create a class `NetworkConnection` that connects to a mock server in its constructor and disconnects in its destructor. Demonstrate object lifecycle.
13	Write a program with class `Database` where a static member maintains the number of active connections. Ensure connections are opened/closed via methods and track the total.
14	Design two classes `Student` and `Result`. Use a friend function to calculate grade of a student based on marks stored privately in `Student` and criteria in `Result`.
15	Write a program to define a class `Polynomial` and overload `+` and `*` operators to add and multiply two polynomials.