

## **LAB 1: Introduction to Java & Basic Programs**

1. Write a Java program to display your name, faculty, and university.
2. Write a program to check whether a number is even or odd.
3. Write a program to compute the sum of the first  $n$  natural numbers.
4. Write a program to find the largest of three numbers.
5. Write a program to check whether a given year is a leap year.

## **LAB 2: Classes and Objects**

1. Create a class Student with attributes roll, name, and marks; display the details.
2. Create a class Rectangle with methods to calculate area and perimeter.
3. Write a program to create a class BankAccount and perform deposit/withdraw operations.
4. Create a class Time and add two time objects.
5. Create a class Complex to add and subtract complex numbers.

## **LAB 3: Constructors & Method Overloading**

1. Create a class with a default and parameterized constructor.
2. Demonstrate method overloading with different parameter lists.
3. Create a Box class with multiple constructors (volume calculation).
4. Implement constructor overloading for a Book class.
5. Create a Distance class with overloaded methods to convert meters ↔ kilometers.

## **LAB 4: Inheritance**

1. Implement single inheritance (Person → Student).
2. Implement multilevel inheritance (Vehicle → Car → ElectricCar).
3. Demonstrate hierarchical inheritance using at least three child classes.
4. Write a program to show constructor calling order in inheritance.
5. Create a program where a derived class overrides a method from the base class.

## **LAB 5: Polymorphism**

1. Demonstrate runtime polymorphism using method overriding.
2. Create a program using dynamic method dispatch.
3. Show compile-time polymorphism using overloaded methods.
4. Write a program where a parent reference refers to different child objects.
5. Implement a shape class hierarchy (Circle, Rectangle, Triangle) with an overridden area() method.

## **LAB 6: Abstraction & Interfaces**

1. Create an abstract class Shape with abstract method area().
2. Implement an interface Playable and provide classes like Football, Cricket, etc.
3. Create an interface Vehicle with multiple implementations.
4. Show multiple inheritance using interfaces.
5. Write a program where an abstract class contains both abstract and concrete methods.

## **LAB 7: Arrays & Strings**

1. Write a program to find the largest element in an array.
2. Implement matrix addition.
3. Implement matrix multiplication.
4. Perform common string operations (substring, length, replace, compare).
5. Write a program to check whether a string is a palindrome.

## **LAB 8: Exception Handling**

1. Write a program to handle ArithmeticException.
2. Demonstrate multiple catch blocks.
3. Use try, catch, finally to handle exceptions.
4. Create a user-defined exception InvalidAgeException.
5. Write a program that throws and catches a custom exception.

## **LAB 9: File Handling**

1. Write a program to read and write text files.
2. Count the number of words in a file.
3. Copy content from one file to another.
4. Append text to an existing file.
5. Serialize and deserialize an object.

## **LAB 10: Java GUI (AWT/Swing)**

1. Create a simple window using Swing.
2. Design a login form with text fields and buttons.
3. Handle button click events using ActionListener.
4. Create a calculator GUI with basic operations.
5. Build a form that stores user input into a file.

## **LAB 11: Java and Database**

1. Create a database connection using JDBC .
2. Perform CRUD Operations