**List of Practical**

**Minimum ten practical’s have to be performed and 1 virtual lab experiment**

1. Write a Java program to demonstrate the use of classes and object concepts with a real-world example (e.g., a simple banking system with account details and operations like deposit and withdrawal).

2. Write a Java program to illustrate the concept of compile-time polymorphism (method overloading) by creating a calculator that can add integers, doubles, and arrays.

3. Write a Java program to illustrate the concept of dynamic polymorphism (method overriding) using a real-world example, such as a vehicle system where different types of vehicles have their own implementation of a start method.

4. Write a Java program to demonstrate the working of the final modifier on classes, methods, and variables by creating a program that ensures a constant value cannot be changed and a class cannot be extended.

5. Write a Java program to illustrate the need for inheritance with a real-time example, such as a company employee hierarchy where different types of employees (full-time, part-time) inherit common attributes from a base Employee class.

6. Write a Java program to demonstrate the working of both abstract and final modifiers, showcasing the restriction on class extension and the need for abstract methods in a base class (e.g., an abstract class Shape with abstract method draw).

7. Write a Java program to demonstrate the purpose of interfaces, by implementing a payment gateway system where different payment methods (credit card, PayPal) use a common Payment interface.

8. Write a Java program to demonstrate the purpose of user-defined packages, organizing a program into multiple packages for better modularity, such as creating a package for mathematical operations and another for user interaction.

9. Write a Java program to demonstrate the notion of multiple catch blocks, by handling different types of exceptions such as ArrayIndexOutOfBoundsException, NullPointerException, and ArithmeticException.

10. Write a Java program to implement a thread-based multitasking system, where multiple threads perform tasks like printing numbers, calculating sums, and sorting an array concurrently.

11. Write a Java program to demonstrate the working of the List interface and its classes, by creating a task manager application where tasks are added, removed, and displayed using ArrayList or LinkedList.

12. Write a Java program to illustrate the working of the Set interface and its classes, by developing a program that manages a collection of unique student IDs, demonstrating HashSet and TreeSet.

13. Write a Java program to illustrate the working of the Map interface and its classes, by implementing an employee database where employee IDs are mapped to their respective names, using HashMap and TreeMap.