Core Java Hands-On Practice Problems

Practice Problems:

1. Write a program to check if a number is prime. (Basics, Loops)
2. Reverse an integer without using String. (Basics, Loops, Operators)
3. Find the factorial of a number (recursion + loop). (Recursion, Loops)
4. Print the Fibonacci series up to n. (Recursion, Loops)
5. Find the largest and smallest element in an array. (Arrays, Loops)
6. Check if a string is a palindrome. (Strings)
7. Sort an array without using Arrays.sort(). (Arrays, Loops)
8. Rotate an array to the right by k steps. (Arrays, Loops)
9. Count the number of vowels and consonants in a string. (Strings, Loops)
10. Remove duplicate characters from a string. (Strings, Collections)
11. Find the most frequent character in a string. (Strings, Collections)
12. Create a BankAccount class with deposit, withdraw, checkBalance methods. (Classes, Objects, Constructors, Encapsulation)
13. Create a Car class that counts how many objects are created. (Static variables, OOP)
14. Create a Student class with private fields and implement getters and setters. (Encapsulation, OOP)
15. Create Animal → Dog → Puppy to show inheritance. (Inheritance)
16. Create Shape class with area() overridden in Circle and Rectangle. (Polymorphism, Method Overriding)
17. Demonstrate method overloading with add(). (Polymorphism, Overloading)
18. Demonstrate method overriding using Parent/Child classes. (Polymorphism, Overriding)
19. Show Dynamic Method Dispatch with Parent ref → Child obj. (Polymorphism, DMD)
20. Create abstract Employee class with calculateSalary(). Extend into FullTimeEmployee and PartTimeEmployee. (Abstraction)
21. Create Playable interface with play(). Implement in MusicPlayer and VideoPlayer. (Interfaces)
22. Create a Functional Interface Calculator and implement using Lambda. (Functional Interface, Lambda)
23. Throw custom exception if balance < withdraw amount. (Custom Exception)
24. Handle ArrayIndexOutOfBoundsException with try-catch. (Exception Handling)
25. Demonstrate finally block always executing. (Exception Handling)
26. Read input and throw NumberFormatException if invalid. (Exception Handling)
27. Store student names in an ArrayList and print them. (ArrayList)
28. Store unique numbers in a HashSet. (HashSet)
29. Use HashMap to count word frequency in a sentence. (HashMap)
30. Sort employees by salary using Comparable. (Comparable)
31. Sort employees by name using Comparator. (Comparator)
32. Create two threads: one prints even numbers, another prints odd numbers. (Threads)
33. Demonstrate Thread.sleep() by printing numbers with delay. (Threads)
34. Create class extending Thread and another implementing Runnable. (Threads, Runnable)
35. Simulate race condition where 2 threads withdraw from same account. (Threads, Synchronization)
36. Filter even numbers, square them, and sort using streams. (Streams)
37. Find longest string from a list using streams. (Streams)
38. Convert names to uppercase using map(). (Streams)
39. Find sum of odd numbers using reduce(). (Streams)
40. Count frequency of words in a list using streams. (Streams)
41. ATM Simulation (deposit, withdraw, check balance, custom exception). (OOP, Exception Handling)
42. Student Management System using ArrayList. (OOP, Collections)

43. Library Management System using HashMap. (Collections, OOP)
44. Online Shopping Cart using ArrayList. (Collections, OOP)
45. Multithreaded Counter with race condition and fix using synchronized. (Threads, Synchronization)