Java Intermediate Programming Questions

This file contains a curated list of 50 progressively challenging programming questions designed to test your knowledge of Java fundamentals and help you build problem-solving logic.

☆ Level 1: Core Understanding & Syntax Practice (Q1–Q15)

Focus: Data types, operators, flow control, type casting, variables

1. Write a program to swap two numbers without using a third variable.
2. Check if a number is even or odd using ternary operator.
3. Write a program to print the ASCII value of a character.
4. Calculate the area of a circle, rectangle, and triangle using user input.
5. Convert temperature from Celsius to Fahrenheit.
6. Check if a year is a leap year or not.
7. Find the largest of three numbers using nested `if`.
8. Check if a number is prime.
9. Print all prime numbers between 1 and 100.
10. Find factorial of a number using both loop and recursion.
11. Reverse a number.
12. Check if a number is a palindrome.
13. Find the sum of digits of a number.
14. Print Fibonacci series up to N terms.
15. Check if a number is an Armstrong number.

☆ Level 2: Strings, Loops, and Logic Building (Q16–Q30)

Focus: Strings, StringBuffer, loop nesting, conditional logic

16. Count vowels, consonants, digits, and white spaces in a string.

17. Check if a string is a palindrome.
18. Reverse a string using `StringBuffer` and manually using loops.
19. Remove all white spaces from a string.
20. Check if two strings are anagrams.
21. Count the occurrence of each character in a string.
22. Print the frequency of each word in a sentence.
23. Find the longest word in a sentence.
24. Replace all vowels in a string with `*`.
25. Toggle the case of each character in a string.
26. Sort characters in a string in ascending order.
27. Find duplicate characters in a string.
28. Compress a string (e.g., aabbbcc → a2b3c2).
29. Check if a string contains only digits.
30. Write a method that returns true if two strings are rotations of each other.

☆ Level 3: Object-Oriented Concepts (Q31–Q40)

Focus: Classes, objects, inheritance, encapsulation, polymorphism

31. Create a class `Student` with fields name, roll, and marks. Write methods to accept and display details.
32. Implement a bank account system with deposit and withdraw methods using encapsulation.
33. Create a class hierarchy: `Shape` → `Circle`, `Rectangle`, `Triangle`. Use inheritance and override area methods.
34. Create a class `Person`. Derive `Employee` from it. Add salary and department. Use constructors.
35. Demonstrate constructor overloading with a `Box` class (length, width, height).
36. Demonstrate method overloading with a `Calculator` class.
37. Use `super` to invoke the parent class constructor.
38. Create an abstract class `Vehicle` with abstract method `move()`. Implement in `Car` and `Bike`.
39. Demonstrate the use of `final` keyword with a class, method, and variable.
40. Write a program to show dynamic method dispatch (runtime polymorphism).

☆ Level 4: Advanced Logic & Modifiers (Q41–50)

Focus: Modifiers, access control, static blocks, complex logic

41. Demonstrate access specifiers (private, protected, default, public) using a package.
42. Create a class with a `static` block and explain its execution order.
43. Write a static method to find the GCD of two numbers.
44. Use the `Scanner` class to take input for a 2D array and print it.
45. Create a utility class with all methods static to handle string utilities (reverse, vowel count, etc.).
46. Create a mini student grading system with marks as input and grade as output.
47. Simulate a login system with max 3 attempts using loops and conditions.
48. Create a simple `Book` management class with methods to add, search, and list books.
49. Create a program to detect duplicate elements in an array of integers.
50. Implement a mini calculator with switch-case that performs basic operations: +, -, *, /, %, and exit.

> ✉ **Tip:** Start with 2 programs per day. Use meaningful variable names and document your logic with comments.

Some Questions

- Count of Matches in Tournament.
- Trapping Rain Water.
- Pascal's Triangle.
- Climbing Stairs.
- Find the Missing Number
- Find the Single Number
- Reverse Array with O(1) space.