■ Top 50 Java Coding Questions for SDET Interviews

Developed by Ajay

GeeksforGeeks: https://www.geeksforgeeks.org/user/user_vvmb/
LinkedIn: https://www.linkedin.com/in/ajay-varma-profile

This guide contains 50 frequently asked Java coding questions for SDET/QA interviews. Each question includes a brief explanation, an example input/output and a GeeksforGeeks reference link (Java specific where available). Use this for rapid revision and interview practice.

Table of Contents

String Programs (1-10)

1. Reverse a String

Example: Input: "Geeks" → Output: "skeeG"
Explanation: Reverse characters of the string. Useful in text-processing and simple algorithmic checks.
GeeksforGeeks: https://www.geeksforgeeks.org/java/reverse-a-string-in-java/

2. Check if a String is Palindrome

Example: Input: "madam" → true; Input: "hello" → false
Explanation: Compare characters from both ends moving inward or reverse and compare. Frequently asked to test logic and edge-cases.
GeeksforGeeks: https://www.geeksforgeeks.org/java/java-program-to-check-whether-a-string-is-a-palindrome/

3. Count Vowels and Consonants in a String

Example: Input: "Hello World!" → Vowels:3, Consonants:7
Explanation: Traverse and classify characters; useful for validating text data.
GeeksforGeeks: https://www.geeksforgeeks.org/java/java-program-to-count-the-total-number-of-vowels-and-consonants-in-a-string/

4. Find Duplicate Characters in a String

Example: Input: "programming" → r, g, m
Explanation: Use frequency map to identify characters with count > 1; helps in uniqueness checks.
GeeksforGeeks: https://www.geeksforgeeks.org/dsa/print-all-the-duplicates-in-the-input-string/

5. Remove Duplicate Characters from a String

Example: Input: "banana" → "ban"
Explanation: Track seen characters (boolean array or LinkedHashSet) and build result preserving first occurrences.
GeeksforGeeks: https://www.geeksforgeeks.org/remove-all-duplicates-from-the-input-string/

6. Count Number of Words in a String

Example: Input: "This is a test." → 4 or 5 depending on separators
Explanation: Split on whitespace or parse to count transitions from space to non-space; useful for log parsing.
GeeksforGeeks: https://www.geeksforgeeks.org/count-words-in-a-given-string/

7. Find First Non-Repeated Character

Example: Input: "swiss" → 'w'
Explanation: Use LinkedHashMap to preserve order while counting, then pick first with count 1.
GeeksforGeeks: https://www.geeksforgeeks.org/given-a-string-find-its-first-non-repeating-character/

8. Check if Two Strings are Anagrams

Example: Input: "listen","silent" → true
Explanation: Compare sorted forms or frequency counts; tests handling of permutations.
GeeksforGeeks: https://www.geeksforgeeks.org/check-whether-two-strings-are-anagram-of-each-other/

9. Reverse Each Word in a Sentence

Example: Input: "Hello World" → "olleH dlroW"

Explanation: Split sentence by spaces, reverse each token and rejoin; useful for text transformations.
GeeksforGeeks: https://www.geeksforgeeks.org/reverse-individual-words/

10. Check if String Contains Only Digits

Example: Input: "12345" → true; "12a45" → false
Explanation: Use Character.isDigit or regex ^\d+$; used in validating numeric text fields.
GeeksforGeeks: https://www.geeksforgeeks.org/check-if-given-string-contains-only-digits/

Number Programs (11-20)

11. Check if a Number is Prime

Example: Input: 29 → true; 15 → false
Explanation: Test divisibility up to sqrt(n); common math/logic interview question.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-check-prime-number/

12. Check if a Number is Armstrong

Example: Input: 153 → true (1^3+5^3+3^3=153)
Explanation: Sum of digits raised to power of number length equals original; tests loops and power logic.
GeeksforGeeks: https://www.geeksforgeeks.org/armstrong-numbers-between-two-intervals/

13. Check if a Number is Palindrome

Example: Input: 121 → true; 123 → false
Explanation: Reverse digits and compare with original; tests numeric manipulations.
GeeksforGeeks: https://www.geeksforgeeks.org/check-if-a-number-is-palindrome/

14. Find Factorial of a Number

Example: Input: 5 → 120
Explanation: Multiply numbers from 1..n; watch for overflow in Java (use long/BigInteger).
GeeksforGeeks: https://www.geeksforgeeks.org/factorial-of-a-number-using-recursion-in-java/

15. Generate Fibonacci Series

Example: Input: n=6 → 0,1,1,2,3,5
Explanation: Iteratively add previous two numbers; used in sequence generation tests.
GeeksforGeeks: https://www.geeksforgeeks.org/program-for-fibonacci-numbers/

16. Sum of Digits of a Number

Example: Input: 123 → 6
Explanation: Extract digits via %10 and accumulate; common input-validation check.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-find-sum-of-digits-in-an-integer/

17. Reverse a Number

Example: Input: 1234 → 4321
Explanation: Build reversed number using modulo and multiply-add; watch negative numbers.
GeeksforGeeks: https://www.geeksforgeeks.org/reverse-digits-number-java/

18. GCD (HCF) of Two Numbers

Example: Input: 12,18 → 6
Explanation: Use Euclidean algorithm (modulo loop); basic number theory useful in logic tests.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-find-gcd-of-two-numbers/

19. LCM of Two Numbers

Example: Input: 12,18 → 36
Explanation: Compute via LCM = (a*b)/GCD; beware overflow.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-find-lcm-of-two-numbers/

20. Check if a Number is Perfect

Example: Input: 28 → true

Explanation: Sum proper divisors and compare to number; tests loops and correctness.
GeeksforGeeks: https://www.geeksforgeeks.org/perfect-number/

Array Programs (21-30)

21. Find Largest and Smallest in an Array

Example: Input: [3,5,1,9] → max=9, min=1
Explanation: Single pass tracking of min/max; often used in test assertions.
GeeksforGeeks: https://www.geeksforgeeks.org/find-largest-element-in-an-array/

22. Find Second Largest Element

Example: Input: [10,5,8,20] → 10
Explanation: Track first and second largest in one pass; watch duplicates.
GeeksforGeeks: https://www.geeksforgeeks.org/find-second-largest-element-array/

23. Sum of All Elements in Array

Example: Input: [1,2,3] → 6
Explanation: Simple loop sum; include tests for empty arrays.
GeeksforGeeks: https://www.geeksforgeeks.org/find-sum-of-elements-in-array-in-java/

24. Reverse Array

Example: Input: [1,2,3,4] → [4,3,2,1]
Explanation: Swap ends inward or create new array; verify in-place behavior.
GeeksforGeeks: https://www.geeksforgeeks.org/reverse-an-array-in-java/

25. Sort Array Without Built-in Methods

Example: Input: [3,1,2] → [1,2,3]
Explanation: Implement bubble/selection/insertion sort to demonstrate fundamentals.
GeeksforGeeks: https://www.geeksforgeeks.org/sort-array-without-using-sort-method/

26. Remove Duplicates from Array

Example: Input: [1,2,2,3] → [1,2,3]
Explanation: Use HashSet or sort+unique; preserve order if needed (LinkedHashSet).
GeeksforGeeks: https://www.geeksforgeeks.org/remove-duplicates-sorted-array/

27. Find Missing Number in Array (1..n)

Example: Input: [1,2,4,5] → 3
Explanation: Use sum formula or XOR trick; common easy interview question.
GeeksforGeeks: https://www.geeksforgeeks.org/find-the-missing-number/

28. Count Frequency of Each Element

Example: Input: [1,2,2,3] → 1:1,2:2,3:1
Explanation: Use HashMap to tally frequencies; used in validation of outputs.
GeeksforGeeks: https://www.geeksforgeeks.org/count-frequencies-elements-array/

29. Check if Two Arrays are Equal

Example: Input: [1,2,3], [3,2,1] → true if same elements
Explanation: Sort and compare or use frequency maps; consider order-sensitivity.
GeeksforGeeks: https://www.geeksforgeeks.org/check-if-two-arrays-are-equal-or-not/

30. Merge Two Arrays

Example: Input: [1,3], [2,4] → [1,3,2,4] or merged sorted [1,2,3,4]
Explanation: Concatenate or merge sorted arrays; useful for data aggregation.
GeeksforGeeks: https://www.geeksforgeeks.org/merge-two-sorted-arrays/

Pattern Programs (31-35)

31. Print Right-Angle Triangle Pattern

*Example: Input: n=4 → *,**,***,**** *
Explanation: Nested loops to print increasing stars; tests control flow.
GeeksforGeeks: https://www.geeksforgeeks.org/triangle-pattern-in-java/

32. Print Pyramid Pattern of Numbers

Example: Input: n=3 → centered number pyramid
Explanation: Use nested loops and spacing for centered patterns.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-print-pyramid-pattern/

33. Print Diamond Pattern

Example: Input: n=3 → diamond of stars
Explanation: Combine upper and lower pyramids; tests loop boundaries.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-print-diamond-shape-pattern/

34. Print Pascal's Triangle

Example: Input: n=4 → rows of Pascal's triangle
Explanation: Compute binomial coefficients iteratively or using previous row.
GeeksforGeeks: https://www.geeksforgeeks.org/pascal-triangle/

35. Print Floyd's Triangle

Example: Input: n=4 → sequential numbers in rows
Explanation: Incremental counter in nested loops; simple control flow exercise.
GeeksforGeeks: https://www.geeksforgeeks.org/programs-printing-pyramid-patterns-java/

General Logic Programs (36-45)

36. Swap Two Numbers Without Third Variable

Example: Input: a=5,b=3 → a=3,b=5
Explanation: Use arithmetic (a=a+b; b=a-b; a=a-b) or XOR trick to swap values.
GeeksforGeeks: https://www.geeksforgeeks.org/swap-two-numbers-without-using-temporary-variable/

37. Check Leap Year

Example: Input: 2020 → true; 1900 → false; 2000 → true
Explanation: Leap if divisible by 4 and not by 100 unless divisible by 400.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-check-leap-year/

38. Find Largest of Three Numbers

Example: Input: 4,9,2 → 9
Explanation: Use nested if-else or Math.max for clarity; consider equals.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-find-largest-of-three-numbers/

39. Convert Binary to Decimal

Example: Input: 1011 → 11
Explanation: Parse bits left-to-right multiplying by 2 or use Integer.parseInt with base 2.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-convert-binary-to-decimal/

40. Convert Decimal to Binary

Example: Input: 13 → 1101
Explanation: Divide by 2 collect remainders or use Integer.toBinaryString; test leading zeros.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-convert-decimal-to-binary/

41. Count Number of Digits in Integer

Example: Input: 12345 → 5
Explanation: Loop dividing by 10 or use log10 for positive numbers.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-count-the-digits-of-a-number/

42. Find Even and Odd Numbers in Range

Example: Input: 1..5 → Even:2,4 Odd:1,3,5
Explanation: Simple modulus checks inside loop; test inclusive/exclusive ranges.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-print-even-numbers-from-1-to-n/

43. Sum of First N Natural Numbers

Example: Input: n=5 → 15
Explanation: Use formula n*(n+1)/2 or loop; beware overflow.
GeeksforGeeks: https://www.geeksforgeeks.org/sum-of-first-n-natural-numbers/

44. Power of a Number Without Math.pow

Example: Input: base=2,exp=3 → 8
Explanation: Multiply base exp times; consider fast exponentiation for efficiency.
GeeksforGeeks: https://www.geeksforgeeks.org/java-program-to-calculate-power-of-a-number/

45. Check Strong Number

Example: Input: 145 → true
Explanation: Sum factorials of digits and compare; tests loops and factorial logic.
GeeksforGeeks: https://www.geeksforgeeks.org/strong-numbers/

Special Number Programs (46-50)

46. Check Happy Number

Example: Input: 19 → true
Explanation: Replace number by sum of squares of digits; detect cycles via set.
GeeksforGeeks: https://www.geeksforgeeks.org/happy-number/

47. Check Neon Number

Example: Input: 9 → true
Explanation: Square number and sum digits, compare to original.
GeeksforGeeks:
https://www.geeksforgeeks.org/java-program-to-check-whether-a-number-is-neon-number-or-not/

48. Check Automorphic Number

Example: Input: 25 → true
Explanation: Check if square ends with original number using modulo with power of 10.
GeeksforGeeks: https://www.geeksforgeeks.org/automorphic-number/

49. Check Harshad Number

Example: Input: 18 → true
Explanation: Sum digits and test divisibility; straightforward digit operations.
GeeksforGeeks: https://www.geeksforgeeks.org/harshad-number/

50. Find Nth Prime Number

Example: Input: n=6 → 13
Explanation: Iterate and count primes until nth found; simple primality checks.
GeeksforGeeks: https://www.geeksforgeeks.org/program-to-find-the-nth-prime-number/