

Java Practice Questions

● LEVEL 1 – Beginner (1–100)

◆ Basic Input, Output, and Data Types

1. Write a program to read an integer and print it.
2. Write a program to take a float input and print it.
3. Take two integers as input and print their sum.
4. Input a character and print its ASCII value.
5. Take a boolean input and display it.
6. Read an integer and print whether it's even or odd using `if-else`.
7. Read two numbers and print the larger one using `if-else`.
8. Take a float input and print it rounded to the nearest whole number.
9. Read an integer and print its square.
10. Read a number and check whether it's positive, negative, or zero.

◇ *while Loops*

11. Print numbers from 1 to 10 using a `while` loop.
12. Print even numbers from 1 to 20 using `while` loop.
13. Print numbers from 10 to 1 using `while`.
14. Print the first 10 multiples of 5.
15. Print the sum of numbers from 1 to N.
16. Read a number N and print the factorial of N using `while`.
17. Print all odd numbers between 1 to N.
18. Read a number and count how many digits it has.
19. Reverse the digits of a given number using `while`.
20. Check if a number is a palindrome.

◇ *Condition Checks and Logical Operators*

21. Read three numbers and find the largest.

22. Check if a number is divisible by both 5 and 11.
23. Read a character and check if it's a vowel.
24. Read a year and check if it's a leap year.
25. Read a number and check if it's in the range 10 to 100.
26. Check if a person is eligible to vote ($\text{age} \geq 18$).
27. Read two numbers and check if the first is divisible by the second.
28. Read marks of a student and assign grade (A/B/C/Fail).
29. Read temperature and check if it's Hot/Cold/Warm.
30. Read three angles and check if they can form a triangle.

◇ 1D Arrays

31. Read 5 integers into an array and print them.
32. Read 10 integers and print their sum.
33. Count how many numbers in the array are even.
34. Print only the odd elements from an array.
35. Find the maximum number in an array.
36. Read 5 float values into an array and print their average.
37. Read an array and print elements in reverse order.
38. Count how many positive and negative numbers are in the array.
39. Read 10 integers and print how many are divisible by 3.
40. Read 5 names into a String array and print them.

◇ 2D Arrays

41. Read a 2x2 integer matrix and print it.
42. Read and print a 3x3 matrix.
43. Input a 3x3 matrix and print the sum of all elements.
44. Count the number of even numbers in a 2D array.
45. Read a 2D array and print the main diagonal.
46. Print the row-wise sum of a 2D array.
47. Read a 2D array and find the maximum element.
48. Print column-wise elements of a matrix.
49. Print all elements on the secondary diagonal.
50. Print the border elements of a 2D array.

◇ **Methods**

51. Write a method that adds two numbers and returns the result.
52. Write a method to check if a number is even.
53. Write a method that returns the square of a number.
54. Write a method that takes a character and returns `true` if it is a vowel.
55. Write a method to find the maximum of three numbers.
56. Write a method to calculate area of a rectangle.
57. Write a method that takes an array and returns the sum.
58. Write a method to reverse a number.
59. Write a method that takes a year and returns if it is leap year.
60. Write a method to count digits in a number.

◇ **Logical, Assignment, Relational Operators**

61. Read two numbers and check if they are equal using `==`.
62. Read a number and use `!=` to check if it's not zero.
63. Use `>=` and `<=` to check age range.
64. Use logical AND (`&&`) to check if number is in range 10–20.
65. Use logical OR (`|`) to check if number is `< 10` or `> 100`.
66. Demonstrate all relational operators with two integers.
67. Read a number and increment it using `+=`.
68. Decrease a number by 2 using `-=` operator.
69. Multiply a value by 3 using `*=` operator.
70. Use `!` operator to toggle a boolean value.

◇ **Pattern Printing with `while`**

71. Print 5 stars in a row.
72. Print a right triangle of stars (5 rows).
73. Print a number pattern like:
1
1 2
1 2 3
74. Print a triangle using characters:
A
A B
A B C
75. Print a square of numbers (3x3).

- 76. Print a hollow square (4x4).
- 77. Print a pyramid of numbers using `while`.
- 78. Print a mirrored triangle.
- 79. Print a pattern with decreasing stars.
- 80. Print a pattern with alternate 0s and 1s.

◇ *Miscellaneous Practice*

- 81. Swap two numbers using a temporary variable.
- 82. Take a number and print its multiplication table.
- 83. Convert Celsius to Fahrenheit.
- 84. Convert inches to centimeters.
- 85. Take marks of 5 subjects and calculate percentage.
- 86. Calculate simple interest.
- 87. Calculate area and circumference of a circle.
- 88. Calculate power using a `while` loop (e.g., a^b).
- 89. Print sum of digits of a number.
- 90. Count number of vowels in a string.

◇ *Mini Challenges*

- 91. Print the Fibonacci series up to N terms.
- 92. Read a number and check if it's an Armstrong number.
- 93. Check if a character is uppercase, lowercase, or digit.
- 94. Reverse a string (without using built-in methods).
- 95. Check if a number is prime.
- 96. Find the second largest number in an array (no sorting).
- 97. Calculate GCD of two numbers using `while`.
- 98. Take an array and count frequency of a given number.
- 99. Find the average of numbers divisible by 5 in an array.

- 100. Check if an array is symmetric (i.e., reads same forward and backward).

● LEVEL 2 – Intermediate (1–100)

◆ Complex Conditionals and Logical Flow

1. Read three numbers and check if they form a Pythagorean triplet.
2. Check whether a character is alphabet, digit, or special symbol.
3. Read time in 24-hour format and convert to 12-hour format.
4. Check if a number lies within two user-entered boundaries.
5. Take three numbers and check if they can form a triangle.
6. Implement a basic calculator using `if-else` (add, sub, mul, div).
7. Take an age and check category: child/teen/adult/senior.
8. Check if a number is a "Harshad number" (divisible by sum of its digits).
9. Take three sides and identify triangle type (scalene, isosceles, equilateral).
10. Input electricity units and calculate bill using slab system.

◇ Nested *while* and Control Structures

11. Print a square pattern using nested `while` loop.
12. Print multiplication tables from 1 to N using nested loops.
13. Print Pascal's Triangle up to N rows using `while`.
14. Print the following pattern:
1
2 3
4 5 6
15. Print Floyd's triangle using `while`.
16. Take a number and print a pyramid with that many levels.
17. Print the alphabet pattern:
A
B B
C C C
18. Take input and print right-aligned triangle.
19. Print inverted triangle pattern using numbers.
20. Print a diamond pattern using stars.

◇ **1D Array with Logic**

21. Take 10 integers and count how many are prime.
22. Find the sum of numbers at even indices.
23. Replace all negative numbers in an array with 0.
24. Search for an element in the array.
25. Find the frequency of each element in an array.
26. Find the average of numbers greater than 50.
27. Check if the array has any duplicate elements.
28. Replace all even numbers in the array with their half.
29. Merge two arrays of size 5 into one.
30. Reverse an array without creating a new one.

◇ **2D Arrays – Logic and Computation**

31. Read a 3x3 matrix and count all prime elements.
32. Find the row with the maximum sum.
33. Transpose a 2x2 matrix.
34. Find the sum of both diagonals in a matrix.
35. Replace all even elements in a matrix with -1.
36. Count how many rows have all positive numbers.
37. Read two 2x2 matrices and print their element-wise sum.
38. Create an identity matrix of size 3x3.
39. Read a matrix and check if it's symmetric.
40. Print matrix in spiral order (3x3 only).

◇ **Methods – Decomposition and Return**

41. Write a method to return reverse of a number.
42. Create a method to check if a character is uppercase.
43. Write a method to convert lowercase to uppercase.
44. Write a method to return count of vowels in a string.
45. Method to check if number is palindrome.
46. Method to check if number is Armstrong.
47. Write a method to return sum of digits of a number.
48. Create a method that takes an array and returns the count of even numbers.
49. Method to check if a string is palindrome.
50. Method to print Fibonacci series up to n.

◇ **Logical + Assignment + Relational Operators**

- 51. Take two boolean inputs and print result of AND, OR, NOT.
- 52. Demonstrate short-circuiting in logical expressions.
- 53. Use compound assignment to multiply and add a value.
- 54. Compare characters using relational operators.
- 55. Use logical operators to validate a login (username/password).
- 56. Use relational and logical operators to check if student passed.
- 57. Demonstrate pre/post increment in an expression.
- 58. Chain relational operators using &&.
- 59. Chain logical operations to validate complex conditions.
- 60. Use != to identify non-zero values in array.

◇ **Pattern Practice – While-based**

- 61. Print number pyramid with spaces.
- 62. Print left-aligned increasing triangle of characters.

- 63. Print a pattern like:

```
5
4 4
3 3 3
```

- 64. Print star pattern in reverse order using while.
- 65. Print even number triangle (2, 4, 6...).
- 66. Print a pattern:

```
1
0 1
1 0 1
```

- 67. Print increasing odd numbers pattern.
- 68. Print triangle where rows start from user input.
- 69. Print pyramid with characters from user.
- 70. Print reverse alphabet triangle (Z to A).

◇ **Multi-Concept Problems**

- 71. Input N, read N integers into an array, return how many are > 100.
- 72. Read a sentence, count total vowels and consonants.

73. Read array, print all numbers that are divisible by 2 and 3.
74. Count how many rows in a matrix have sum > 100.
75. Take temperature for 7 days, find average.
76. Read a 2D matrix and return maximum element of each row.
77. Input 10 student marks, find how many passed (≥ 40).
78. Read 5 names, return the longest one.
79. Take 2D char array and count how many are vowels.
80. Input a date and print whether it's a valid calendar date.

◇ **Challenge Practice**

81. Input a number and convert it to binary manually using `while`.
82. Create a number guess game (user tries to guess random number).
83. Validate a 4-digit PIN using Scanner and logic.
84. Simulate a basic ATM (withdraw, balance, deposit).
85. Create a simple login system (username/password with retry limit).
86. Take a number and check if it's "strong number" (sum of factorial of digits = number).
87. Print nth Fibonacci number using method.
88. Create a calculator with menu (add, sub, mul, div) using methods.
89. Take a string and print each word on a new line.
90. Input 10 values, replace all duplicates with -1.

◇ **Method-based Problems with Arrays**

91. Method that accepts array and returns average of odd numbers.
92. Method that returns number of prime numbers in an array.
93. Method that accepts 2D matrix and returns sum of boundary elements.
94. Method that checks if 1D array is palindrome.
95. Method that replaces all negative elements with square of absolute value.
96. Method to find the longest word in an array of strings.
97. Method that accepts 2 strings and returns true if they are anagrams.
98. Method that checks if each row in 2D matrix is sorted.
99. Method that returns true if a number is present in matrix.
100. Method that returns how many times max element occurs in array.

● LEVEL 3 – Advanced (1–100)

◆ Multi-Layered Conditions

1. Read a date (dd, mm, yyyy) and check if it's a valid calendar date (leap year logic required).
2. Input a password and validate length, special characters, and digits using logical operators.
3. Take three numbers and determine if they can form an arithmetic or geometric sequence.
4. Implement a basic tax calculator based on income slabs.
5. Read a number and determine if it's a "perfect number" (sum of divisors = number).
6. Implement a basic BMI calculator with classification (Underweight, Normal, etc.).
7. Input two times (hours and minutes) and compute the time difference.
8. Read a 6-digit number and determine if it's a "lucky number" (sum of first 3 digits = sum of last 3).
9. Create a voting eligibility system with country, age, and registration status check.
10. Input a number and print all factors, then determine if it's deficient, abundant, or perfect.

◇ *Nested while, Loops & Multi-Checks*

11. Generate a pattern of Pascal's triangle up to N levels.
12. Take a number and print a diamond of digits using while.
13. Print number triangle that shifts left each row.
14. Create a hollow diamond pattern using stars.
15. Create a mirrored Floyd's triangle.
16. Create a triangle pattern using only odd numbers.
17. Print alphabet pyramid with left and right alignment.
18. Create a zig-zag pattern with alternating 0s and 1s.
19. Print pattern that alternates between numbers and letters row-wise.
20. Print an hourglass pattern using while.

◇ **Advanced 1D Array Manipulation**

21. Read 10 integers and shift all elements right by one.
22. Rotate an array left by k positions using a method.
23. Count number of palindromes in an integer array.
24. Replace each element with sum of next two elements.
25. Remove duplicates from an array without using extra arrays.
26. Check if a sequence is arithmetic or geometric.
27. Find length of the longest contiguous increasing subarray.
28. Count frequency of all digits (0–9) in an array of integers.
29. Create a method that reverses only even elements in an array.
30. Calculate difference between max and min values in an array.

◇ **Advanced 2D Array Logic**

31. Input a 3x3 matrix and check if it is a magic square.
32. Check if a 3x3 matrix is a Latin square (each number appears only once per row and column).
33. Print boundary of matrix in clockwise order.
34. Calculate and return the saddle point of a matrix (smallest in row, largest in column).
35. Mirror a matrix horizontally and print.
36. Count how many prime numbers are present in each row.
37. Create a method that swaps first and last row of a matrix.
38. Replace all diagonal elements with zero.
39. Count number of unique elements in a matrix.
40. Print the matrix in snake pattern (zig-zag row-wise).

◇ **Method Design – Higher Utility**

41. Method to check if an array is “almost sorted” (no more than 2 out-of-place elements).
42. Method that accepts matrix and returns true if all rows are sorted.
43. Method that reverses digits of all even numbers in an array.
44. Method that checks if all elements in a row are prime.
45. Method that accepts 1D array and returns a new array with squares of original elements.
46. Method to calculate standard deviation of an integer array.

- 47. Method that takes character array and returns count of uppercase, lowercase, digits.
- 48. Method to generate Fibonacci sequence and return it in an array.
- 49. Method that accepts two matrices and returns element-wise max.
- 50. Method to return the first non-repeating element in an array.

◇ ***Data Type Challenges***

- 51. Accept float values, truncate them to 2 decimal places manually.
- 52. Take a double, convert to string, reverse it using logic (not StringBuilder).
- 53. Accept char input, return next alphabet character.
- 54. Input boolean values and perform XOR logic.
- 55. Read float and int arrays and compute their element-wise product.
- 56. Check which data type can hold the user-entered number (byte, short, int, long).
- 57. Perform operations mixing int, float, char, and print types using instanceof.
- 58. Accept a long number and count how many digits are even.
- 59. Simulate a character counter with type filtering (ignore symbols).
- 60. Accept a float array and print all values above average.

◇ ***Advanced Loop + Condition Challenges***

- 61. Input a number and return the sum of digits until a single digit remains.
- 62. Take a number and print all combinations of its digits.
- 63. Print all numbers from 1 to 100 that are divisible by the sum of their digits.
- 64. Generate and print all 2-digit numbers where both digits are prime.
- 65. Read a string and print frequency of each character (manually).
- 66. Count the number of prime digits in a number.
- 67. Print all armstrong numbers between 100–999.
- 68. Check if the digits of a number are in increasing order.
- 69. Generate a pattern where each row doubles the previous value.
- 70. Print a triangle of binary values with alternating rows.

◇ ***Real-World Logic Scenarios***

- 71. Basic shopping cart: read product prices into array, total bill with tax.
- 72. Simple bank transaction history using array.

73. Quiz grading system (MCQs), compare answers with key, count score.
74. Evaluate student marks from 3 subjects and apply scholarship rules.
75. Time sheet: input hours/day for a week, calculate total and overtime.
76. Simulate parking lot entry and exit (5 slots) using 1D array.
77. Read multiple email IDs and count how many are Gmail addresses.
78. Count how many words in an array start with a capital letter.
79. Given sales array, return number of days sales exceeded average.
80. Track rainfall data over 7 days, find peak rainfall day.

◇ **Challenging Patterns**

81. Print diamond pattern of numbers centered and spaced.
82. Generate pattern of odd numbers in descending triangle.
83. Alternating star and number triangle.
84. Binary triangle with mirror pattern.
85. Fibonacci triangle with increasing rows.
86. Pascal's triangle (mod 2) using while.
87. Pyramid with decreasing characters (E, D, C...).
88. Numeric pyramid where each row is square of row number.
89. Full star diamond using only while.
90. Hollow inverted triangle with numbers.

◇ **Final 10 Challenge Problems**

91. Create a grade calculator system for multiple students using 2D array.
92. Accept user records (name, age, score) and return names of top scorers.
93. Read a sentence and return the word with most vowels.
94. Create a password strength checker using if-else and logical operators.

95. Build a pattern of numbers like:

```
1
2 1
3 2 1
```

96. Rotate a matrix 90 degrees clockwise using only logic (no libs).
97. Create a method to validate if a 2D matrix is a magic square.
98. From a 1D array, extract subarray with the maximum even sum.
99. Build a student record system with input validation using methods.
100. Create a number system converter (binary <-> decimal) using loops.