100 Java Programs to Improve Logic & Problem-Solving

■ Part 1 — Beginner (Loops, Conditions, Numbers)

Print "Hello World"
Hint: Use System.out.println

Add two numbers
Hint: Take two numbers and add them

Check even or odd
Hint: Use modulo operator (%)

Find largest of two numbers
Hint: Use if-else condition

Find largest of three numbers
Hint: Use nested if-else

Check positive, negative, or zero
Hint: if-else conditions

Swap two numbers with temp
Hint: Use a temporary variable

Swap two numbers without temp
Hint: Use arithmetic operations

Sum of first N natural numbers
Hint: Use loop or formula n*(n+1)/2

Factorial of a number
Hint: Use loop or recursion

■ Part 2 — Intermediate (Arrays & Strings)

Find largest element in an array
Hint: Iterate and keep max

Find smallest element in an array
Hint: Iterate and keep min

Calculate sum of array elements
Hint: Loop through array and sum

Find average of array elements
Hint: Sum / array length

Count even and odd numbers in array
Hint: Use modulo inside loop

Sort array in ascending order
Hint: Use nested loops or Arrays.sort()

Sort array in descending order
Hint: Sort then reverse or custom compare

Search an element in array (linear search)
Hint: Loop and check equality

Implement binary search
Hint: Array must be sorted; divide & conquer

Reverse elements of an array
Hint: Swap start & end elements

■ Part 3 — Strings & Functions

Count vowels and consonants
Hint: Loop through characters and check vowels

Reverse a string
Hint: Loop from end or use StringBuilder.reverse()

Check palindrome string
Hint: Compare reversed string with original

Convert string to uppercase
Hint: Use toUpperCase()

Convert string to lowercase
Hint: Use toLowerCase()

Count words in a string
Hint: Split by space and count

Remove spaces from string
Hint: Use replaceAll or manual check

Remove special characters
Hint: Use regex or Character.isLetterOrDigit()

Check if two strings are anagrams
Hint: Sort strings and compare

Find duplicate characters in a string
Hint: Use HashMap or nested loops

■■ Part 4 — Recursion & OOP Basics

Factorial using recursion
Hint: n! = n * factorial(n-1)

Fibonacci using recursion
Hint: fib(n) = fib(n-1) + fib(n-2)

Power of a number using recursion
Hint: pow(x,n) = x * pow(x,n-1)

Sum of digits using recursion
Hint: sum = last digit + sum(remaining)

GCD using recursion
Hint: Use Euclidean algorithm recursively

Palindrome check recursive
Hint: Compare first and last characters

Reverse number using recursion
Hint: Take last digit and recur

Count digits using recursion
Hint: Divide number by 10 recursively

Print numbers 1 to N recursively
Hint: Base case N=0

Sum of array elements using recursion
Hint: sum(arr, n) = arr[n-1] + sum(arr, n-1)

■ Part 5 — Object-Oriented Logic

Create Student class and display info
Hint: Use fields and method to print details

BankAccount with encapsulation
Hint: Private variables with getters/setters

Employee class with inheritance
Hint: Use extends keyword

Method overloading example
Hint: Same method name, different parameters

Method overriding example
Hint: Child class redefines parent method

Constructor overloading
Hint: Multiple constructors with different args

Implement interface example
Hint: Use implements keyword

Abstract class example
Hint: Use abstract methods and concrete methods

Simple Calculator using OOP
Hint: Methods for add, subtract, multiply, divide

Library management mini-class structure
Hint: Classes for Book, Member, Library

■ Part 6 — Patterns & Printing

Solid rectangle
Hint: Nested loops

Hollow rectangle
Hint: Print stars on border

Right triangle pattern
Hint: Outer and inner loops

Inverted triangle
Hint: Outer loop decreasing

Pyramid
Hint: Spaces + stars

Inverted pyramid
Hint: Spaces + decreasing stars

Pascal’s triangle
Hint: Use combination formula or previous row

Floyd’s triangle
Hint: Increment numbers row-wise

Diamond pattern
Hint: Combine pyramid and inverted pyramid

Number pattern (1, 22, 333...)
Hint: Print row index repeatedly

■ Part 7 — Advanced Logic & DSA Basics

Bubble sort
Hint: Compare adjacent and swap

Selection sort
Hint: Find min/max and swap

Insertion sort
Hint: Shift elements to insert

Merge sort
Hint: Divide and merge recursively

Quick sort
Hint: Partition and sort recursively

Binary search (recursive)
Hint: Compare mid element

Balanced parentheses using Stack
Hint: Push for '(' and pop for ')'

Find missing number in array
Hint: Sum formula or XOR trick

Find pair with given sum in array
Hint: Use HashSet or two-pointer

Check Armstrong number
Hint: Sum of digits^count == number

■ *Tip: Start with beginner problems daily, think logically, and try to solve hints before checking solutions.*