CodInClub

Powered By BridgeLabz

DSOOPS - FA 3 REFERENCE QUESTIONS

Compiling and Interpreting Java Program, `public static void main(String[] args)`, Command Line Arguments

Easy

1. Write a Java program that prints “Hello, World!” when executed.
2. Write a program that prints the number of command-line arguments passed to it.
3. Write a program that prints the first command-line argument, or “No args” if none are provided.
4. Write a program that prints all command-line arguments, each on a new line.
5. Write a program that checks if exactly one argument is passed and prints “OK” if true, else “Invalid args”.

Medium

1. Write a program that sums all numeric command-line arguments (ignore non-numeric ones) and prints the total.
2. Write a program that finds and prints the longest string among all command-line arguments.
3. Write a program that reverses the order of command-line arguments and prints them space-separated.
4. Write a program that checks if the last argument is “--debug” and prints “Debug mode ON” if true.
5. Write a program that accepts two numbers as args and prints their product. If not exactly two args, print “Usage java Program <num1> <num2>”.

Hard

1. Write a CLI calculator Accept 3 args — number, operator (+,-,*,/), number — and print the result. Handle invalid input gracefully.
2. Write a program that accepts any number of file paths as args and prints “Exists” or “Not Found” for each (use `java.io.File`).
3. Write a program that groups command-line arguments by their length and prints them in ascending order of length.
4. Write a program that treats every odd-positioned argument as a key and even-positioned as value, then prints them as key=value pairs.
5. Write a program that accepts a flag like “--sort” as the first argument, and if present, sorts and prints the remaining args alphabetically.

Identifiers, Keywords, Java Data Types & Operators

Easy

1. Write a program that declares two int variables, assigns values, adds them, and prints the result.
2. Write a program that swaps two integer variables using a temporary variable.
3. Write a program that calculates and prints the area of a rectangle (length * width).
4. Write a program that checks if a number is even or odd using the modulus operator.
5. Write a program that increments a variable using `++` and prints its value before and after.

Medium

1. Write a program that takes an integer and prints whether it’s positive, negative, or zero using the ternary operator only.
2. Write a program that calculates `(a + b)²` without using `Math.pow()`, using only arithmetic operators.

3. Write a program that takes 3 numbers and prints the middle one (not max, not min) using only relational and logical operators.
4. Write a program that uses compound assignment operators (`+=`, `*=`, etc.) to modify a variable 5 times and print the final value.
5. Write a program that uses bitwise operators to toggle the 3rd bit of an integer and print the result.

Hard

1. Write a program that evaluates a string like `"5+3*2"` by parsing and applying correct operator precedence (no `eval` or scripting).
2. Write a program that simulates “operator overloading” by creating methods `add()`, `multiply()` etc. for a custom Number class.
3. Write a program that converts a decimal number to binary using only bitwise operators and no `Integer.toBinaryString()`.
4. Write a program that takes two integers and returns the maximum without using `if`, `switch`, `Math.max`, or ternary operator.
5. Write a program that implements a 4-bit adder using only bitwise operations (AND, OR, XOR, shifts) — simulate full binary addition.

Control Statements — Decision, Loops, Jump (break, continue, return)

Easy

1. Write a program that prints numbers 1 to 10 using a for loop.
2. Write a program that prints even numbers from 1 to 20 using a while loop.
3. Write a program that prints “Pass” if score >= 50, else “Fail”, using if-else.
4. Write a program that skips printing number 5 in a 1-10 loop using `continue`.

5. Write a program that breaks out of a loop when it encounters the number 7.

Medium

1. Write a program that prints the first 10 Fibonacci numbers using a loop.
2. Write a nested loop program to print multiplication tables from 1 to 5.
3. Write a program that finds the first prime number greater than 100 and prints it (use break when found).
4. Write a program that reads numbers from user until 0 is entered, then prints the sum (use while + break).
5. Write a program that prints numbers 1-100 but replaces multiples of 3 with “Fizz”, multiples of 5 with “Buzz”, and multiples of both with “FizzBuzz”.

Hard

1. Write a program with nested loops that finds and prints all Pythagorean triplets ($a^2 + b^2 = c^2$) where a,b,c <= 50. Use labeled break to exit early if needed.
2. Write a program that simulates a retry mechanism Ask user for password (hardcode “secret”). Allow 3 attempts. Use `continue` for invalid, `break` on success.
3. Write a method that takes an array and returns the index of the first negative number. Use `return` inside loop to exit early.
4. Write a program that prints a hollow diamond pattern using nested loops and `continue` to skip inner content.
5. Write a program that generates a random number 1-100, then lets user guess it. Loop until correct. Use `break` on correct guess, `continue` to skip invalid inputs.

Pattern Problems

Easy

1. Write a program to print a right triangle of 5 rows using `*`

```
*
**
***
****
*****
```

2. Write a program to print numbers 1 to 5 in a single row `1 2 3 4 5`
3. Write a program to print a square of 4 rows and 4 columns of `*`.
4. Write a program to print numbers from 5 to 1 in reverse order, each on new line.
5. Write a program to print a single row of 8 `` characters.

Medium

1. Write a program to print a pyramid of 5 rows

```
    *
   ***
  *****
 *******
*********
```

2. Write a program to print Floyd's Triangle with 4 rows

```
1
2 3
4 5 6
7 8 9 10
```

3. Write a program to print a hollow square of 5x5 stars (only border).
4. Write a program to print an inverted right triangle of 5 rows

*****

****

***

**

*

5. Write a program to print a diamond pattern (5 rows up, 4 rows down).

Hard

1. Write a program to print the following number pattern

1

21

321

4321

54321

2. Write a program to print Pascal's Triangle up to 6 rows using combinations (nCr) computed via loops.

3. Write a program to print a butterfly pattern with 4 rows

* *

** **

*** ***

****

*** ***

** **

* *

4. Write a program to print the following

A
BB
CCC
DDDD
EEEEE

5. Write a program to print a spiral matrix pattern for n=4

1 2 3 4
12 13 14 5
11 16 15 6

---

5 Linear Arrays — Representation, Traversal, Insertion, Deletion + Applications (DB, Caching, Matrix)

Easy

1. Write a program to declare an array of 5 integers, initialize it, and print all elements.
2. Write a program to find the sum of all elements in an integer array.
3. Write a program to find the maximum element in an array.
4. Write a program to reverse the elements of an array and print the reversed array.
5. Write a program to check if a given number exists in an array. Print “Found” or “Not Found”.

Medium

1. Write a program to insert a number at a given index in an array (shift elements right).

2. Write a program to delete an element at a given index in an array (shift elements left).
3. Write a program to remove all duplicates from a sorted array and print the unique elements.
4. Write a program to merge two sorted arrays into a third sorted array.
5. Write a program to rotate an array to the right by 1 position.

Hard

1. Write a program to simulate a fixed-size LRU cache using arrays (track access order, evict least recently used on overflow).
2. Write a program to multiply two 3x3 matrices stored as 2D arrays and print the result matrix.
3. Write a program to find the longest contiguous subarray with sum = 0.
4. Write a program to simulate a simple “database table” using 2D array — insert rows, delete by index, print all.
5. Write a program to implement “circular buffer” using array — support add(), remove(), and wrap-around indexing.

Recursion using Java and its Applications

Easy

1. Write a recursive function to calculate the factorial of a number.
2. Write a recursive function to print numbers from 1 to n.
3. Write a recursive function to calculate the sum of first n natural numbers.
4. Write a recursive function to find the nth Fibonacci number.
5. Write a recursive function to check if a string is a palindrome.

Medium

1. Write a recursive function to reverse a string.
2. Write a recursive function to calculate the power of a number (x^n).
3. Write a recursive function to find the GCD of two numbers using Euclidean algorithm.
4. Write a recursive function to count the number of digits in an integer.
5. Write a recursive function to check if an array is sorted in ascending order.

Hard

1. Write a recursive function to generate all permutations of a given string.
2. Write a recursive solution for Tower of Hanoi with n disks.
3. Write a recursive function to solve the N-Queens problem (place N queens on NxN board without attacking).
4. Write a recursive function to find all subsets of a given array (subset generation).
5. Write a recursive function to solve the “Rat in a Maze” problem — find a path from top-left to bottom-right in a 2D grid (1=wall, 0=path).