Multidimensional (2D) Array Topics

1. Definition & Concept
2. Declaration & Initialization
3. Accessing Elements
4. Traversal (nested for / for-each loop)
5. Common Operations (sum, max, min, row/column sum, diagonal sum, transpose)
6. Passing 2D Array to Methods
7. Returning 2D Array from Methods
8. Anonymous 2D Arrays

Jagged (Ragged) Array Topics

1. Definition & Concept
2. Declaration & Initialization
3. Accessing Elements
4. Traversal (nested loops)
5. Common Operations (row sum, max of rows, overall max)
6. Passing Jagged Array to Methods
7. Returning Jagged Array from Methods
8. Anonymous Jagged Arrays

🧩 Anonymous Array (Java Notes)

🔹 **Definition:**

An **array without any name** is called an **Anonymous Array**.
It is **created and used immediately** — mostly for **one-time use**.

🔹 **Syntax:**

```
new dataType[] { elements };
```

🔹 **Example:**

```
System.out.println(sum(new int[] {10, 20, 30}));
```

👉 The array {10, 20, 30} has no name.
👉 It is directly passed to the method sum().

🔹 Example Program:

```
class AnonymousArray {
   static int sum(int[] arr) {
      int total = 0;
      for (int i : arr)
         total += i;
      return total;
   }

   public static void main(String[] args) {
      System.out.println("Sum = " + sum(new int[] {10, 20, 30, 40}));
   }
}
```

Output:

```
Sum = 100
```

🔹 Key Points:

- No name → cannot be reused
- Created using new keyword
- Used mostly in **method calls**
- **Size decided automatically** by elements
- Can be **1D or 2D**

🔹 2D Anonymous Array Example:

```
printMatrix(new int[][] {
   {1, 2, 3},
   {4, 5, 6}
});


static void printMatrix(int[][] mat) {
    for (int[] row : mat) {
        for (int val : row)
            System.out.print(val + " ");
        System.out.println();
    }
}
```

---

25. Advantages of Arrays

- Easy to access using index
- Memory-efficient (contiguous)
- Easy to traverse and manipulate
- Supports random access

---

26. Limitations of Arrays

- Fixed size (cannot grow/shrink)
- Difficult insertion/deletion
- Can store only similar data type
- No built-in methods for dynamic resizing

---

27. Real-life Examples of Arrays in Java

- Storing marks of students
- Keeping daily temperature records
- Managing stock prices
- Game leaderboards
- Storing employee IDs, salaries, etc.