**✅ 1. Single-Dimensional Arrays in Java**

A **single-dimensional array** is a list of elements of the same type, stored in contiguous memory.

**Syntax:**

java

CopyEdit

int[] arr = new int[5]; // declaration + memory allocation

arr[0] = 10; // initialization

**Or directly initialize:**

java

CopyEdit

int[] arr = {10, 20, 30, 40, 50};

**Loop through an array:**

java

CopyEdit

for (int i = 0; i < arr.length; i++) {

System.out.println(arr[i]);

}

**✅ 2. Multi-Dimensional Arrays in Java**

A **multi-dimensional array** is an array of arrays. The most common is a **2D array**.

**Syntax for 2D Array:**

java

CopyEdit

int[][] matrix = new int[3][3]; // 3 rows, 3 columns

**Initialization:**

java

CopyEdit

int[][] matrix = {

{1, 2, 3},

{4, 5, 6},

{7, 8, 9}

};

**Access Elements:**

java

CopyEdit

System.out.println(matrix[0][1]); // Output: 2

**Traverse a 2D array:**

java

CopyEdit

for (int i = 0; i < matrix.length; i++) {

for (int j = 0; j < matrix[i].length; j++) {

System.out.print(matrix[i][j] + " ");

}

System.out.println();

}

**✅ 3. Nested Loops in Java**

Nested loops are **loops inside other loops**, useful when dealing with multi-dimensional data.

**a) Nested for loop:**

java

CopyEdit

for (int i = 1; i <= 3; i++) {

for (int j = 1; j <= 2; j++) {

System.out.println("i = " + i + ", j = " + j);

}

}

**b) Nested while loop:**

java

CopyEdit

int i = 1;

while (i <= 3) {

int j = 1;

while (j <= 2) {

System.out.println("i = " + i + ", j = " + j);

j++;

}

i++;

}

**c) Nested do-while loop:**

java

CopyEdit

int i = 1;

do {

int j = 1;

do {

System.out.println("i = " + i + ", j = " + j);

j++;

} while (j <= 2);

i++;

} while (i <= 3);

**📌 Use Cases for Nested Loops and 2D Arrays**

* Matrix operations (addition, multiplication)
* Tabular data processing
* Games like Tic-Tac-Toe
* Printing patterns (stars, numbers)

**✨ Bonus: Pattern Printing Example Using Nested Loops**

java

CopyEdit

// Right-angled triangle pattern

for (int i = 1; i <= 5; i++) {

for (int j = 1; j <= i; j++) {

System.out.print("\* ");

}

System.out.println();

}