**🔹 1. What is an array in Java? Explain with an example.**

**Answer:**  
An array in Java is a data structure that stores multiple values of the same type in a single variable. It helps in reducing the number of variables needed and allows accessing elements using an index.

**Example:**

int[] marks = {85, 90, 78, 88, 76};

System.out.println(marks[2]); // Outputs 78

Here, marks is an integer array storing 5 values. Indexing starts from 0.

**🔹 2. How do you declare an array in Java? List different ways.**

**Answer:**  
There are multiple ways to declare and initialize an array in Java:

1. Declaration only:

int[] arr;

1. Declaration + size:

arr = new int[5];

1. Declaration + initialization:

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

1. Using new with values:

int[] arr = new int[]{5, 10, 15};

**🔹 3. What is zero-based indexing in arrays?**

**Answer:**  
Zero-based indexing means the first element of an array is at index 0, not 1.  
So, if arr = {10, 20, 30}, then:

* arr[0] = 10
* arr[1] = 20
* arr[2] = 30

Trying to access arr[3] will result in an ArrayIndexOutOfBoundsException.

**🔹 4. How do you access the last element of an array safely?**

**Answer:**  
To access the last element:

arr[arr.length - 1];

This ensures you never go out of bounds, even if array size changes.

**🔹 5. What is the default value stored in an array after memory allocation?**

**Answer:**  
When an array is created using new, default values are assigned:

* int[] → 0
* double[] → 0.0
* boolean[] → false
* String[] → null

Example:

int[] nums = new int[3]; // {0, 0, 0}

**🔹 6. Can we change the size of an array once declared?**

**Answer:**  
No. Arrays in Java are **fixed in size**. Once declared (e.g., new int[5]), the size cannot be changed.  
To use dynamic resizing, use ArrayList from the Collections framework.

**🔹 7. How do you modify an element in an array?**

**Answer:**  
Use the index to assign a new value:

arr[2] = 100;

This changes the 3rd element in the array. Indexing starts from 0.

**🔹 8. What happens if you access an invalid array index?**

**Answer:**  
Java throws a **runtime exception**:

Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException

This happens when you access an index less than 0 or greater than or equal to arr.length.

**🔹 9. How do you print all elements of an array without using loops (manually)?**

**Answer:**

int[] arr = {1, 2, 3};

System.out.println(arr[0]);

System.out.println(arr[1]);

System.out.println(arr[2]);

**🔹 10. What is the use of arr.length in Java?**

**Answer:**  
arr.length returns the size of the array.  
It is **not a method**, it’s a **property**.  
Used to:

* Loop through the array
* Find the last index: arr[arr.length - 1]

**🔹 11. Explain the concept of an array with real-life examples.**

**Answer:**  
An array is like a **row of mailboxes** or **train compartments** where each box stores one item, and every box has a number (index).  
Example: Marks of 5 students → int[] marks = new int[5];

**🔹 12. Can you store different data types in a single array?**

**Answer:**  
No. Arrays are **homogeneous** — they store values of the same type only.  
If you need mixed types, use Object[], but that’s not recommended for beginners.

**🔹 13. What is the difference between declaring and initializing an array?**

**Answer:**

* Declaration: int[] arr; (Just tells the type)
* Initialization: arr = new int[5]; (Allocates memory)

You can also combine both:

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

**🔹 14. How do you store names of 5 students using arrays?**

**Answer:**

String[] names = {"Ravi", "Meena", "Arjun", "Sara", "Ali"};

System.out.println(names[2]); // Prints "Arjun"

**🔹 15. What is the output of the following code?**

int[] arr = {1, 2, 3};

System.out.println(arr[3]);

**Answer:**  
This will throw:

ArrayIndexOutOfBoundsException: Index 3 out of bounds for length 3

Because valid indices are 0, 1, 2 only.

**🔹 16. Can an array be null? If yes, how?**

**Answer:**  
Yes. You can declare an array without initialization:

int[] arr = null;

Trying to access arr[0] will throw a **NullPointerException**.

**🔹 17. What are multidimensional arrays?**

**Answer:**  
These are arrays within arrays. Most common: **2D arrays** (like matrices):

int[][] matrix = {

{1, 2},

{3, 4}

};

System.out.println(matrix[1][0]); // 3

**🔹 18. How do you check the number of elements in an array?**

**Answer:**  
Use arr.length.  
If array = {5, 10, 15}, arr.length = 3

**🔹 19. What is the difference between length and length()?**

**Answer:**

* For arrays: arr.length → property (no ())
* For strings: str.length() → method

**🔹 20. Can we use a loop without knowing the array size?**

**Answer:**  
Yes, always use arr.length in loops:

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

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

}

Avoid hardcoding values like i < 5.