**1. What is array traversal?**

**Answer:**  
📖 **Definition:** Array traversal means visiting each element of an array **one by one** to either **read**, **print**, or **modify** its value.  
It is essential because arrays store multiple elements, and we usually need to process all of them.

✅ **Example:**

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

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

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

}

**Key Point:** Traversal is done using loops like for, for-each, or while.

**2. Why do we use loops for arrays?**

**Answer:**  
We use loops because:

1. They prevent **repetitive code** (no need to write arr[0], arr[1] manually).
2. They work with **arrays of any size** using .length.
3. They make programs **scalable and maintainable**.

✅ **Example Without Loop (Bad):**

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

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

✅ **With Loop (Good):**

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

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

}

**3. Write the syntax of a classic for loop to print an array.**

**Answer:**  
📜 **Syntax:**

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

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

}

* int i = 0 → loop starts at first index
* i < array.length → runs until last element
* i++ → moves to next index

**4. Difference between for and for-each loop in arrays.**

**Answer:**

| **For Loop** | **For-Each Loop** |
| --- | --- |
| Uses index (arr[i]) | Directly gets value |
| Can traverse forward or reverse | Traverses forward only |
| Good for modifying specific index | Cannot modify using loop variable |

✅ **Example for loop:**

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

✅ **Example for-each:**

for (int num : arr) { }

**5. Example of printing an array using for-each.**

**Answer:**

int[] numbers = {5, 10, 15};

for (int value : numbers) {

System.out.println(value);

}

**Explanation:**

* value takes each element of the array automatically.
* No index is needed.

**6. Why is .length used in loops with arrays?**

**Answer:**

* .length gives **number of elements** in the array.
* It makes code **dynamic** (works even if array size changes).
* Prevents ArrayIndexOutOfBoundsException.

✅ **Example:**

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

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

}

**7. Write a loop to print an array in reverse.**

**Answer:**

for (int i = arr.length - 1; i >= 0; i--) {

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

}

**Explanation:**

* Start from last index (arr.length - 1).
* Decrease i until 0.

**8. Can we modify array elements inside a for-each loop?**

**Answer:**  
No, because the loop variable is a **copy** of the element, not the original array slot.  
To modify, use a **classic for loop**:

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

arr[i] += 10;

}

**9. Program to print all even numbers in an array.**

**Answer:**

for (int num : arr) {

if (num % 2 == 0) {

System.out.println(num);

}

}

**Explanation:**

* % 2 == 0 checks divisibility by 2.
* Prints only even numbers.

**10. What happens if we use i <= arr.length in a loop?**

**Answer:**  
It will cause **ArrayIndexOutOfBoundsException** because the last valid index is arr.length - 1.  
Example:  
If arr.length = 5, valid indices are 0 to 4. i = 5 is invalid.

**11. Which loop should you use if you want the index of each element?**

**Answer:**  
Use a **classic for loop**, because it has a counter variable (i) that represents the index.

**12. Program to calculate the sum of all elements.**

**Answer:**

int sum = 0;

for (int num : arr) {

sum += num;

}

System.out.println("Sum: " + sum);

**Explanation:**

* sum accumulates values as loop runs.

**13. What is the output?**

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

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

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

}

**Answer:**  
1 2 3

**14. Can we use while loop to traverse an array?**

**Answer:**  
Yes.  
Example:

int i = 0;

while (i < arr.length) {

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

i++;

}

**15. Find the largest element using loops.**

**Answer:**

int max = arr[0];

for (int num : arr) {

if (num > max) {

max = num;

}

}

System.out.println("Max: " + max);

**16. Which loop is shorter and cleaner: for or for-each?**

**Answer:**

* for-each is **shorter and cleaner** for reading values.
* Classic for is better when index is needed.

**17. Can you break out of a loop while traversing?**

**Answer:**  
Yes, using break:

for (int num : arr) {

if (num == 50) break;

}

**18. Count positive elements in an array.**

**Answer:**

int count = 0;

for (int num : arr) {

if (num > 0) count++;

}

**19. Difference between .length and .length()?**

**Answer:**

* Arrays: .length → property (no brackets)
* Strings: .length() → method (with brackets)

**20. Can a for-each loop traverse a 2D array?**

**Answer:**  
Yes, using nested for-each:

for (int[] row : matrix) {

for (int value : row) {

System.out.print(value + " ");

}

}