Java Arrays – Interview Questions and Answers

---

1. What is an array in Java?

An array in Java is a data structure that allows you to store multiple values of the same data type under a single variable name. Arrays are stored in contiguous memory locations and can be accessed using an index starting from 0.

---

2. What are the different ways to declare an array in Java?

There are two main ways:

```java
int[] arr1 = new int[5];    // Preferred Java style
int arr2[] = new int[5];    // C-style (also allowed)
```

---

3. How are arrays initialized?

Arrays can be initialized in two ways:

```java
int[] arr = new int[3];
arr[0] = 1; arr[1] = 2; arr[2] = 3;
```

Or with values directly:

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

---

4. What is the default value of array elements?

Depends on data type:

* `int`: 0
* `float`: 0.0f
* `boolean`: false
* `Object`: null

---

5. What is a two-dimensional array?

A 2D array is an array of arrays. Example:

```java
int[][] matrix = {{1, 2}, {3, 4}};
```

This creates a 2x2 matrix.

---

6. How can you find the length of an array?

Using `.length` property:

```java
int[] arr = {1, 2, 3};
System.out.println(arr.length);  // Outputs 3
```

---

7. How do you loop through an array?

```java
for (int i = 0; i < arr.length; i++) {
    System.out.println(arr[i]);
}
```

Or use for-each:

```java
for (int num : arr) {
    System.out.println(num);
}
```

---

8. What is an ArrayIndexOutOfBoundsException?

Occurs when accessing an index that doesn't exist. Example:

```java
int[] a = new int[3];
a[5] = 10;  // Throws exception
```

---

9. Can we store objects in arrays?

Yes, arrays can store objects:

```java
Student[] students = new Student[3];
students[0] = new Student();
```

---

10. Difference between Array and ArrayList?

* Arrays are fixed-size; ArrayLists are dynamic
* Arrays can store primitives; ArrayLists only objects
* Arrays use `length`; ArrayLists use `size()`

---

11. How to sort an array?

```java
import java.util.Arrays;
int[] arr = {5, 3, 1};
Arrays.sort(arr);
```

---

12. How to copy an array?

```java
int[] copy = Arrays.copyOf(original, original.length);
```

---

13. What is a jagged array?

A 2D array with different column sizes:

```java
int[][] jagged = new int[2][];
jagged[0] = new int[3];
jagged[1] = new int[1];
```

---

14. What is a multidimensional array?

An array with more than 2 dimensions:

```java
int[][][] arr = new int[2][3][4];
```

---

15. What are the disadvantages of arrays?

* Fixed size
* Insertion and deletion are difficult
* Not dynamic

---

16. Can arrays be resized?

No. But you can create a new larger array and copy contents.

---

17. How do you reverse an array?

Loop from both ends and swap:

```java
int i = 0, j = arr.length - 1;
while (i < j) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
    i++; j--;
}
```

---

18. What is the time complexity to access, insert, and delete in arrays?

* Access: O(1)
* Insert/Delete: O(n) (shifting required)

---

19. How are arrays stored in memory?

Arrays are stored in contiguous blocks of memory. For objects, array stores references.

---

20. Can arrays be passed to methods?

Yes. Arrays are passed by reference.

```java
void printArray(int[] arr) {
   for (int x : arr) System.out.println(x);
}
```

15+ Important Java Programs on Arrays

1. Print elements of an array

```java
int[] arr = {1, 2, 3, 4, 5};
for (int num : arr) {
    System.out.println(num);
}
```

2. Find sum of elements in array

```java
int sum = 0;
for (int num : arr) {
    sum += num;
}
System.out.println("Sum = " + sum);
```

3. Find maximum element

```java
int max = arr[0];
for (int num : arr) {
    if (num > max) max = num;
}
System.out.println("Max = " + max);
```

4. Find minimum element

```java
int min = arr[0];
for (int num : arr) {
    if (num < min) min = num;
}
System.out.println("Min = " + min);
```

5. Search for an element

```java
int target = 3;
boolean found = false;
for (int num : arr) {
    if (num == target) found = true;
}
System.out.println(found ? "Found" : "Not Found");
```

6. Reverse an array

```java
for (int i = 0, j = arr.length - 1; i < j; i++, j--) {
    int temp = arr[i];
    arr[i] = arr[j];
    arr[j] = temp;
}
```

7. Remove duplicate elements

```java
Set<Integer> set = new LinkedHashSet<>();
for (int num : arr) {
    set.add(num);
}
System.out.println(set);
```

8. Copy array

```java
int[] newArr = Arrays.copyOf(arr, arr.length);
```

9. Sort array in ascending order

```java
Arrays.sort(arr);
```

10. Sort array in descending order

```java
Integer[] arr = {5, 3, 1};
Arrays.sort(arr, Collections.reverseOrder());
```

11. Merge two arrays

```java
int[] merged = new int[arr1.length + arr2.length];
System.arraycopy(arr1, 0, merged, 0, arr1.length);
System.arraycopy(arr2, 0, merged, arr1.length, arr2.length);
```

12. Count even and odd numbers

```java
int even = 0, odd = 0;
for (int num : arr) {
    if (num % 2 == 0) even++;
    else odd++;
}
System.out.println("Even: " + even + ", Odd: " + odd);
```

13. Left rotate array by one

```java
int first = arr[0];
for (int i = 0; i < arr.length - 1; i++) {
    arr[i] = arr[i + 1];
}
arr[arr.length - 1] = first;
```

14. Right rotate array by one

```java
int last = arr[arr.length - 1];
for (int i = arr.length - 1; i > 0; i--) {
    arr[i] = arr[i - 1];
}
arr[0] = last;
```

15. Check if array is sorted

```java
boolean sorted = true;
for (int i = 0; i < arr.length - 1; i++) {
    if (arr[i] > arr[i + 1]) {
        sorted = false;
        break;
```

```
    }
}
System.out.println(sorted ? "Sorted" : "Not Sorted");
```

16. Count frequency of each element

```java
Map<Integer, Integer> map = new HashMap<>();
for (int num : arr) {
    map.put(num, map.getOrDefault(num, 0) + 1);
}
System.out.println(map);
```

17. Find second largest element

```java
int first = Integer.MIN_VALUE, second = Integer.MIN_VALUE;
for (int num : arr) {
    if (num > first) {
        second = first;
        first = num;
    } else if (num > second && num != first) {
        second = num;
    }
}
System.out.println("Second Largest: " + second);
```

Let me know if you want MCQs, diagrams, or real-world examples added next.