

Java Arrays Complete Cheat Sheet

1. What is an Array?

An array is a collection of elements of the same data type stored in contiguous memory locations. Index starts from 0.

2. Declaration & Initialization

Declaration: `int[] arr;` | Creation: `arr = new int[5];` | Initialization: `int[] arr = {1, 2, 3};`

3. Default Values

`int/byte/short/long` → 0 | `float/double` → 0.0 | `char` → `'\u0000'` | `boolean` → `false` | `Object` → `null`

4. Traversing Arrays

`for (int i=0; i<arr.length; i++)` | `for (int x : arr)` | `Arrays.stream(arr).forEach(System.out::println);`

5. Common Operations

Sum: `sum += arr[i];` | Max: `if(arr[i]>max) max = arr[i];` | Reverse: swap elements using two-pointer method.

6. Arrays Class Methods

`Arrays.toString(arr)`, `Arrays.sort(arr)`, `Arrays.equals(a,b)`, `Arrays.copyOf(a,n)`, `Arrays.fill(a,val)`, `Arrays.binarySearch(a,key)`

7. Multi-Dimensional Arrays

`int[][] matrix = {{1,2},{3,4}};` Access: `matrix[i][j];` Traverse using nested loops.

8. Jagged Arrays

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

9. Array Copying

`System.arraycopy(src,0,dest,0,len);` | `Arrays.copyOf(arr, len);`

10. Arrays vs ArrayList

Array: Fixed size, supports primitives | ArrayList: Dynamic, Objects only (`java.util.ArrayList`)

11. Important DSA Concepts

Reverse/Rotate array, Kadane's Algorithm, Searching & Sorting, Prefix Sum, Frequency Array

12. Common Interview Problems

Find 2nd largest, Move zeros, Rotate array, Merge sorted arrays, Max subarray sum

13. Best Practices

Check bounds, Use for-each if index not needed, Use Arrays.copyOf, Avoid hardcoded lengths

Data Type	Default Value
int, byte, short, long	0
float, double	0.0
char	\u0000
boolean	false
Object references	null