1. Default Values of Array for Different Data Types

In Java, arrays are initialized with default values based on their data types:

- **Numeric types (int, float, double, byte, long, short)** → `0`
- **char** → `'\u0000'` (null character)
- **boolean** → `false`
- **Object references** → `null`

Example:

java
CopyEdit

```
int[] arr = new int[5]; // {0, 0, 0, 0, 0}
boolean[] boolArr = new boolean[3]; // {false, false, false}
String[] strArr = new String[2]; // {null, null}
```

---

2. Can You Pass a Negative Number in Array Size?

No, Java does not allow negative numbers for array size. If attempted, it throws a **NegativeArraySizeException**.

Example:

java
CopyEdit

```
int[] arr = new int[-5]; // Throws NegativeArraySizeException
```

---

3. Where is an Array Stored in JVM Memory?

Arrays in Java are stored in the **Heap Memory**, as they are objects in Java. The reference to the array is stored in the **Stack Memory** if declared inside a method.

---

4. Disadvantages of Arrays

1. **Fixed Size** – Once an array is declared, its size cannot be changed.
2. **Memory Waste** – If the array size is overestimated, unused space is wasted.

3. **No Built-in Methods** – Unlike Collections, arrays do not have built-in methods like `add()` or `remove()`.
4. **Slow Insertions/Deletions** – Shifting elements is required when inserting or deleting elements in the middle.

---

5. What is an Anonymous Array in Java?

An **anonymous array** is an array that is created **without a reference variable** and is used immediately.

Example:

java
CopyEdit

```
new int[]{10, 20, 30, 40};  // Anonymous array

// Passing anonymous array to a method
printArray(new int[]{1, 2, 3, 4});

static void printArray(int arr[]) {
    for (int num : arr) {
        System.out.print(num + " ");
    }
}
```

---

6. Different Ways to Traverse an Array in Java

Using for loop:
java
CopyEdit

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

1.

Using for-each loop:
java

CopyEdit

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

2.

Using while loop:
java
CopyEdit

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

3.

Using Streams (Java 8+):
java
CopyEdit

```
Arrays.stream(arr).forEach(System.out::println);
```

4.

Using Iterator (for collections, not arrays directly):
java
CopyEdit

```
List<Integer> list = Arrays.asList(1, 2, 3, 4);
Iterator<Integer> iterator = list.iterator();
while (iterator.hasNext()) {
    System.out.println(iterator.next());
}
```

5.

---

7. Difference Between `length` and `length()` Method

- `length` – Property used for **arrays** to get the size.
- `length()` – Method used for **Strings** to get the number of characters.

Example:

java
CopyEdit

```
int[] arr = {1, 2, 3, 4, 5};
System.out.println(arr.length); // Output: 5

String str = "Hello";
System.out.println(str.length()); // Output: 5
```