🔹 1. ArrayIndexOutOfBoundsException

- Happens when you try to access an index that does not exist.
- **Cause:** Index < 0 OR Index ≥ array.length

```
class Demo1 {
    public static void main(String[] args) {
        int[] arr = {10, 20, 30};
        System.out.println(arr[2]);  // ✅ valid
        System.out.println(arr[3]);  // ❌
ArrayIndexOutOfBoundsException
    }
}
```

🔹 2. NegativeArraySizeException

- Happens when you try to create an array with a negative size.

```
class Demo2 {
    public static void main(String[] args) {
        int[] arr = new int[-5];  // ❌ NegativeArraySizeException
    }
}
```

🔹 3. NullPointerException

- Happens when the array reference is `null` and you try to use it.

```
class Demo3 {
    public static void main(String[] args) {
        int[] arr = null;
        System.out.println(arr.length);  // ❌ NullPointerException
    }
}
```

🔹 4. ArrayStoreException

- Happens when you try to store **wrong type of object** into an array of objects.

```
class Demo4 {
    public static void main(String[] args) {
        Object[] arr = new String[3];
        arr[0] = "Hello";  // ✅ valid
        arr[1] = 100;      // ❌ ArrayStoreException (int is not
String)
    }
}
```

🔹 5. ClassCastException (in array casting context)

- If you try to cast an array incorrectly.

```
class Demo5 {
    public static void main(String[] args) {
        Object obj = new int[5];
        String[] s = (String[]) obj; // ❌ ClassCastException
    }
}
```

✅ Quick Summary (Array Exceptions in Java)

Exception Name	Cause
ArrayIndexOutOfBoundsException	Accessing invalid index
NegativeArraySizeException	Creating array with negative size
NullPointerException	Using a null array reference
ArrayStoreException	Storing wrong type in an array

ClassCastException	Wrong casting of array types