Java Array Interview Questions and Answers for Freshers

Author: Ramesh Fadatare
Interview Java Java Arrays

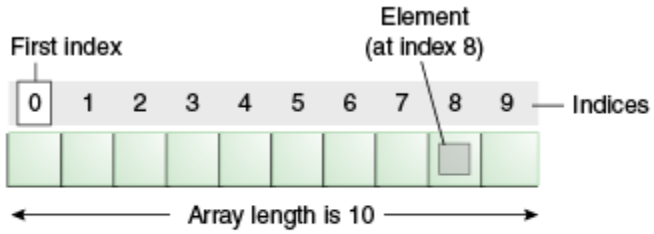
In this article, we will discuss a few frequently asked Java Array interview questions with answers for beginners.

YouTube Video

1. What is an Array?

An Array is a data structure that defines an index-based collection of a fixed number of homogeneous data elements. This means that all elements in the array have the same data type and Array a starts from index 0.

For example, This is an array of 10 elements. All the elements are integers and homogeneous.



An array of 10 elements

The size of an array is fixed and cannot be changed after the array has been created.

In Java, arrays are objects. Arrays can be of primitive data types or reference types.
The main use of Array is used to store multiple values in a single variable, instead of declaring separate variables for each value.

For example:

```
int[] array = { 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 };
```

2. What Is the Advantage and Disadvantage of an Array?

Advantage of an Array

-> **The main use of Array is used to store multiple values in a single variable, instead of declaring separate variables for each value.**

For example:

```
int[] array = { 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 };
```

-> **We can access any element randomly by using indexes provided by arrays.**

For example:

```
// initialize primitive one dimensional array
int[] anArray = new int[5];

anArray[0] = 10; // initialize first element
anArray[1] = 20; // initialize second element
anArray[2] = 30; // and so forth
anArray[3] = 40;
anArray[4] = 50;

// Each array element is accessed by its numerical index:
System.out.println("Element 1 at index 0: " + anArray[0]);
System.out.println("Element 2 at index 1: " + anArray[1]);
System.out.println("Element 3 at index 2: " + anArray[2]);
System.out.println("Element 4 at index 3: " + anArray[3]);
System.out.println("Element 5 at index 4: " + anArray[4]);
```

Output:

```
Element 1 at index 0: 10
Element 2 at index 1: 20
Element 3 at index 2: 30
Element 4 at index 3: 40
Element 5 at index 4: 50
```

-> **We can sort multiple elements of Array at the same time.**

The disadvantage of an Array

Size Limit: We can store the only fixed size of elements in the array. It doesn't grow its size at runtime.

Arrays are Strongly Typed: This means that all elements in the array have the same data type. We can not store different types of data in an Array.

3. What are the Types of Array in Java?

There are two types of array.

1. Single Dimensional Array
2. Multidimensional Array

Single Dimensional Array

A single-dimensional array of Java is a normal array where the array contains sequential elements (of the same type).

For example:

```
int[] array = { 100, 200, 300, 400, 500, 600, 700, 800, 900, 1000 };
```

The following program creates an array of integers, puts some values in the array, and prints each value to standard output.

```
public class ArrayBasics {
     public static void main(String[] args) {
                // declares an array of integers
        int[] anArray;

        // allocates memory for 10 integers
        anArray = new int[10];

        // initialize first element
        anArray[0] = 100;
        // initialize second element
        anArray[1] = 200;
        // and so forth
        anArray[2] = 300;
        anArray[3] = 400;
        anArray[4] = 500;
        anArray[5] = 600;
        anArray[6] = 700;
        anArray[7] = 800;
        anArray[8] = 900;
        anArray[9] = 1000;

        for (int i = 0; i < anArray.length; i++) {
                System.out.println("Element at index " + i + ": " +anArray[i]);
        }
    }
}
```

Output:

```
Element at index 0: 100
Element at index 1: 200
```

```
Element at index 2: 300
Element at index 3: 400
Element at index 4: 500
Element at index 5: 600
Element at index 6: 700
Element at index 7: 800
Element at index 8: 900
Element at index 9: 1000
```

Multidimensional Array

A multi-dimensional array in Java is an array of arrays. A two-dimensional array is an array of one-dimensional arrays and a three-dimensional array is an array of two-dimensional arrays.

Example for two-dimensional array:

```
class MultiDimArrayDemo {

    public static void main(String[] args){
        String[][] names = {
            {"Mr. ","Mrs. ","Ms. "},
            {"Smith","Jones"}};

        // Mr. Smith
        System.out.println(names[0][0] +
                           names[1][0]);
        // Ms. Jones
        System.out.println(names[0][2] +
                           names[1][1]);
    }
}
```

Output:

```
Mr. Smith
Ms. Jones
```

4. Can You Pass the Negative Number in Array Size?

No, you can not pass the negative number as Array size. If you pass a negative number in Array size then you will not get the compiler error. Instead, you will get the NegativeArraySizeException at run time.

5. When ArrayIndexOutOfBoundsException occurs?

ArrayOutOfBoundsException is thrown when an attempt is made to access the Array with an illegal index. For example, an illegal index means if the index is either negative or greater than or equal to the size of the Array.

For example 1: Below program tries to access the element at index 5 but the array index starts from index 0. The index 5 is equal to the size of an Array hence throwing *ArrayOutOfBoundsException*:

```
public class Main {
    public static void main(String[] args) {
        int[] array = { 100, 200, 300, 400, 500 };

        int element = array[5];

        System.out.println(element);
    }
}
```

Output:

```
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index 5 out of
bounds for length 5
        at com.java.sms.Main.main(Main.java:7)
```

For example 2: Access array with negative index to Array leads to throwing *ArrayOutOfBoundsException*:

```
public class Main {
    public static void main(String[] args) {
        int[] array = { 100, 200, 300, 400, 500 };

        System.out.println("Array length -> " + array.length);

        int element = array[-2];

        System.out.println(element);
    }
}
```

Output:

```
Array length -> 5
Exception in thread "main" java.lang.ArrayIndexOutOfBoundsException: Index -2 out of
bounds for length 5
        at com.java.sms.Main.main(Main.java:9)
```

6. Difference of Array and ArrayList

1. An Array is static in nature i.e. of fixed length. The size of an array is fixed and cannot be changed after the array has been created. ArrayList is dynamic in nature. If you add elements to an ArrayList, it will automatically increase its size.

2. An Array can contain both primitive and Object data types. ArrayList does not contain primitive data types (but contains primitive Wrapper classes). It only contains object entries.

3. Java provides *add()* method to insert an element into ArrayList and we can use the **assignment operator** to store elements into Array.

ArrayList.add() method:

```
        List<String> animals = new ArrayList<>();
        // Adding new elements to the ArrayList
        animals.add("Lion");
        animals.add("Tiger");
```

Assignment operator to store elements into Array:

```
// initialize primitive one dimensional array
int[] anArray = new int[5];

anArray[0] = 10; // initialize first element
anArray[1] = 20; // initialize second element
anArray[2] = 30; // and so forth
```

4. We can not use Generics along with Array whereas ArrayList allows you to use Generics to ensure type safety.

5. Length of the ArrayList is provided by the *size()* method while Each array object has the length variable which returns the length of the array.

Example 1: find the length of the ArrayList using the size() method:

```
import java.util.ArrayList;
import java.util.List;

public class Main {
    public static void main(String[] args) {
        // Creating an ArrayList of String using
        List<String> animals = new ArrayList<>();
        // Adding new elements to the ArrayList
        animals.add("Lion");
        animals.add("Tiger");
        animals.add("Cat");
        animals.add("Dog");
        System.out.println(animals.size());
    }
}
```

Output:

```
4
```

Example 2: find the length of an Array using the *length* variable:

```
public class Main {
    public static void main(String[] args) {
        int[] array = { 100, 200, 300, 400, 500 };
        System.out.println("Array length -> " + array.length);
    }
}
```

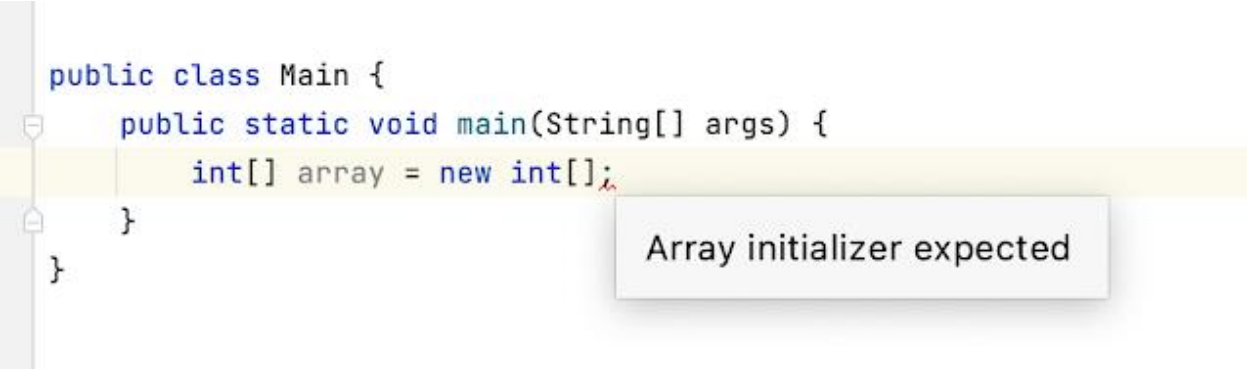
Output:

```
Array length -> 5
```

6. The performance of Array and ArrayList depends on the operation you are performing :
resize() operation: Automatic resize of ArrayList will slow down the performance as it will use a temporary array to copy elements from the old array to the new array.
add() or get() operation: adding an element or retrieving an element from the array or ArrayList object has almost the same performance, as for ArrayList object these operations run in constant time.

7. Can You Declare an Array Without Array Size?

No, you can not declare an Array without an Array size. You will get a compile-time error.



8. Where Does Array Store in JVM Memory?

As we know that Array is an object in java. So, Array is stored in **heap memory** in JVM.

9. What is ArrayStoreException? When this exception is thrown?

ArrayStoreException is a runtime exception. The array must contain the same data type elements.

This exception is thrown to indicate that an attempt has been made to store the wrong type of object in an array of objects. In other words, if you want to store the integer Object in an Array of String you will get *ArrayStoreException*.

The following code throws ArrayStoreException :

```
public class Main {
    public static void main(String[] args) {
        Object x[] = new Employee[3];
        x[0] = new String("javaguides");
    }
}
class Employee{

}
```

Output:

```
Exception in thread "main" java.lang.ArrayStoreException: java.lang.String
        at com.java.sms.Main.main(Main.java:6)
```

10. What is the Difference Between ArrayStoreException and ArrayOutOfBoundsException?

ArrayStoreException is thrown if you are trying to add an incompatible data type. For example, if you try to add an integer object to String Array, then ArrayStoreException is thrown.

ArrayOutOfBoundsException is thrown when an attempt is made to access the Array with an illegal index. For example, an illegal index means if the index is either negative or greater than or equal to the size of the Array.

11. What is an Anonymous Array in Java? Give an Example?

An array without any name (or reference) is called an Anonymous Array. They are useful for scenarios where we need one-time usage of Array.

For example:

```
public class Main {
    public static void main(String[] args)
```

```
    {
        // anonymous array
        sum(new int[]{ 1, 2, 3 });
    }
    public static void sum(int[] a)
    {
        int total = 0;

        // using for-each loop
        for (int i : a)
            total = total + i;

        System.out.println("The sum is:" + total);
    }
}
```

Output:

```
The sum is:6
```

12. Are arrays mutable or immutable in Java?

Arrays are mutable. The size of the array is fixed, but the elements can be changed.

```
int[] arr = {1, 2, 3};
arr[0] = 10; // This modifies the first element of the array from 1 to 10.
```

In the example above, we've changed the first element of the array, demonstrating the mutability of arrays in Java.

13. Java Array Programs Asked in Interviews

1. Java Program to Reverse an Array Without Using Another Array
2. Java Program to Find Duplicate Elements in an Array
3. Java Program to Find Largest Number in an Array
4. Java Program to Check the Equality of Two Arrays