Java Interview Question and Answers

Topic – Java Array:

Most Important Array Interview Questions in Java

- **1. What is an Array in Java?**
- An *array* is a collection of elements (or data) of the same type that are referenced by a common name. For example, we can store a group of strings, a group of int values, or a group of float values in the array. But, we cannot store some int values or some float values in the array.
- **2. How can an array reference variable be declared in java?**
- Ans: An array can be declared by specifying its data type, name, and size. For example, int[ ] a;
- **3. What is an array variable?**
- Ans: An array variable is a reference variable that points to an array object. The memory that is allocated to an array variable actually stores a reference to an array object. It does not actually store an array itself.
- **4. Name the keyword that is used for allocating memory space to an array?**
- Ans: new keyword
- **5. How to create an array in java?**
- Ans: We can create an array object by using a new keyword and assign its reference to the variable. The general syntax for the creation of array object is as follows:

```
datatype[ ] arrayname = new datatype[size];
For example:
int[ ] myList = new int[10]; // It can store 10 elements of int type.
```

- **6. Which element is positioned at num[9] of an array num?**
- Ans: Tenth element.
- **7. If, array[ ] = {10, 20, 5, 30, 25};**
- **a) What is array.length?**
b) What is array[2]?
- Ans: (a) 5, (b) 5
- **8. How to declare a 2d array of size 2 * 3 to store only characters?**

- Ans: char[ ][ ] chrs = new char[2][3];
- **9. How to create an integer array of size 3 * 2 and initialize with values from 1 to 9?**
- Ans: int[ ][ ] x = {{2, 3}, {3, 5}, {5, 9}};
- **10. Where are elements of an array stored?**
- Ans: Elements of an array are stored in the contiguous memory locations of RAM during runtime by JVM. They are stored in heap memory.
- **11. What is the total size of an array chrs having 25 elements of char type?**
- Ans: 50 bytes
- **12. What is the total size of an array num[100] of int type?**
- Ans: 400 bytes.
- **13. What is sorting of array in Java?**
- Ans: The process of arranging elements of an array in a specified array is called sorting of array.
- **14. Can we specify array size as negative?**
- Ans: No, it is not possible to specify array size as negative. Still, if we try to specify the negative size, there will be no compile-time error, but at runtime, we will get an exception named NegativeArraySizeException.
- **15. In which of the following lines of code snippets, there will be compile-time error and why?**

```
int[ ][ ] x = new int[2][ ]; // Line 1
int[ ] y = new int[2]; // Line 2
int num = 2; // Line 3
x[1] = y; // Line 4
x[1] = num; // Line 5
```

- Ans: Line 1 will compile fine and it will create two dimensional array.
- Line 2 will also compile fine and it will create one dimensional array.
- There is no compilation problem in line 3. It declares and initializes an int variable.
- Line 4 also compiles fine since a two-dimensional array is an array of arrays. It assigns a one-dimensional array to one of the dimensions in the two-dimensional array.
- Line 5 will create a compilation error problem because we cannot assign an int variable to an int array variable.

- **16. Can we modify the size (or length) of the array once set it?** Or, can we insert or delete elements after creating an array?
- Ans: No. Once the size of an array is defined, we cannot modify it. We cannot add or delete elements after creation of an array. In other words, an array cannot expand or shrink at runtime.
- **17. What will happen when the following lines of code are compiled and executed?**

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

- Ans: The code above will compile fine but at runtime, it will throw an exception named NegativeArraySizeException because i is a negative number that cannot be used as an array index.
- **18. Will the below code compile and execute successfully?**

```
int i = 3;
int[ ] arr = new int[i];
arr[0] = 1;
arr[1] = 2;
arr[2] = 3;
arr[3] = 4;
```

- Ans: Yes, the lines of code will be compiled successfully but at runtime, there will be ArrayIndexOutOfBoundsException. This exception occurs when we attempt to use an index that is less than zero or greater than or equal to the length of an array.
- **19. Will the above lines of code compile and execute successfully? If yes, what will be the output?**

```
public class Test {
public static void main(String[] args)
{
  int i = 4;
  String[ ] arr = new String[i];
   arr[0] = "A";
   arr[1] = "B";
   arr[2] = "C";
  System.out.print(Arrays.toString(arr));
 }
}
```

- Ans: Yes, the code will be compiled successfully. The output is [A, B, C, null].

- **20. Which exception will be thrown at runtime when the below code will be compiled?**

```
public class Test {
public static void main(String[] args)
{
 Object[ ] arr = new String[3];
   arr[0] = "A";
   arr[1] = "B";
   arr[2] = 20;
   System.out.print(Arrays.toString(arr));
  }
}
```

- Ans: The code will compile fine but at runtime, we will get ArrayStoreException (runtime exception). This exception generally occurs when we attempt to store non-compatible elements in an array object.
- The type of the elements must be compatible with the type of array object. For example, we can store only string elements in an array of type String. If we will attempt to insert integer elements in an array of String, an ArrayStoreException will be thrown at runtime.

- **21. For the below code, what will be the total size taken by the whole array in the memory?**

```
double[ ] num = new double[10];
```

- Ans: Each element in the num array is a variable of double type that needs 8 bytes in the memory. So, the whole array will take space 80 bytes in the memory, plus 8 bytes for the num variable to store the reference of array in the memory.
- **22. What is the difference between int[ ] arr and int arr[ ]?**
- Ans: There is no difference between int[ ] arr and int arr[ ]. Both are the legal ways to declare an array in java. int[ ] arr is a standard declaration.
- **23. What is an anonymous array in java?**
- Ans: An array without reference is called an anonymous array. For example:

```
// Creating an anonymous array.
    System.out.print(new int[]{1, 2, 3}.length); // It will display 3.
```

24. What is the default value of elements of an array at the time of creation?

Ans: When we create an array, it is always initialized with the default values. The default values of an array of different types are as follows:

- If an array is created of byte, short, int, and long type, elements of an array are initialized with 0.0.
- If an array is created of float and double type, elements of an array are initialized with 0.
- If an array is created of Boolean type, the default value is false.
- If an array is created of an Object type, the default value is null.

25. Is there any error in the below code? If not, what is the output of the following program?

```
import java.util.Arrays;

public class Test {

public static void main(String[] args)

{

  String[ ] str = new String[5];

   str[0] = "A";

   str[1] = "20";

   str[3] = "B";

   str[4] = "C";

  System.out.println(Arrays.toString(str));

  }

}
```

Ans: No, there is no error in the above lines of code. The output will be [A, 20, null, B, C].

26. What is the difference between length() method and length in java?

Ans: The length() method is provided by String class that is used to return the number of characters in the string. For example:

```
String str = "Hello world";

System.out.println(str.length());
```

str.length(); will return 11 characters including space.

Whereas, length is an instance variable in arrays that will return the number of values or objects in array. For example:

```
String[ ] names = {"John", "Herry", "Deepak", "Ivaan", "Ricky"};

System.out.println(names.length);
```

names.length; will return 5 since the number of values in names array is 5.

---

27. What is the important difference between an Array and ArrayList in Java?

Ans: There are the following important differences between Arraya and *ArrayList in Java*. They are as follows:

Array:

1. Array is static in size. It has a fixed size.
2. We cannot change the size (or length) of the array after creating it.
3. An array can contain both primitive data types as well as reference types.
4. Array in java does not support generics.
5. Elements of an array can be iterated using for loop.
6. An Array can be multi-dimensional.

ArrayList:

1. ArrayList is dynamic in size.
2. The size of the ArrayList grows or shrinks automatically as we insert or remove elements.
3. ArrayList cannot contain primitive data types. It contains only objects. That is, we can store only reference types in ArrayList.
4. ArrayList in java supports generics.

5. In an ArrayList, we can use an Iterator object to traverse elements.
6. Arraylist is always of a single dimension.

---

28. Which one is the preferred collection for storing objects between an Array and ArrayList?
Ans: ArrayList is backed up by array internally. There are several usability advantages of using an ArrayList over an array in Java. They are:

a) An array has a fixed length at the time of creation. Once it is created we cannot modify its length.
ArrayList is dynamic in size. Once it reaches a threshold, it automatically allots a new array and copies the contents of old array to the new array.

b) Also, ArrayList supports Generics. But Array does not support Generics.

The array should be used in the place of an ArrayList if

- We know the size of an array in advance and do not need re-sizing the collection.
- We need to store the same type of elements.

---

29. Why does not an array support generics in Java?
Or, why cannot we create generic array in java?
Ans: Java does not allow the creation of a generic array because an array carries type information of its elements at runtime. This information is used at runtime to throw ArrayStoreException if data type of elements does not match with the defined type of Array.

In the case of *Generics*, the type information of a collection gets erased at runtime by Type Erasure. Due to which an array cannot use generics.

30. What are the types of arrays in java?
Ans: Generally, arrays are categorized into two forms that are:

- Single dimensional arrays (or 1D array called one-dimensional array)
- Multidimensional arrays (or 2D, 3D arrays)

31. What are the main advantages of arrays in java?
Ans: There are several advantages of arrays in Java. They are:

a) The main advantage of array structure is that it organizes data in such a way that it can be easily manipulated.
b) We can access randomly nth element in an array at a constant time.
c) Arrays generally use less space because no variable needs to support navigation.
d) They help to optimize the code, thereby, we can retrieve or sort the data efficiently.

32. What are the drawbacks/disadvantages/limitations of Arrays in Java?
Ans: There are many drawbacks of arrays in java that are as follows:

a) Since arrays in java are fixed in size, we can store only the fixed size of elements in the array.
b) Once the size of an array is created, it cannot be changed. That is, an array cannot expand or shrink at runtime.
c) We cannot store different kinds of elements in an array.

---

33. What will be the output of the following program?

```
public class Test {

public static void main(String[] args)

{

  int x = 10;

  int[ ] nums = new int[x];

    x = 30;

   System.out.println("x: " +x);

   System.out.println("Size of nums: " +nums.length);

  }

}
```

Ans: x: 30, Size of nums: 10

34. What will be the output of the following code?

```
public class Test {

public static void main(String[] args)

{

   int arr[ ] = {1, 2, 3, 4, 5, 6};

     for (int i = 1; i < arr.length; i++)

        arr[i] = arr[i - 1];

          for (int i = 0; i < arr.length; i++)

          System.out.print(arr[i] + " ");

  }

}
```

Ans: Output: 1 1 1 1 1 1

35. What is a jagged array in java?

Ans: A two-dimensional array having different rows is called jagged array. It is also called ragged array in java.

36. How many ways of copying an array into another array?

Ans: There are four ways available in java to copy an array into another array. They are:

- Using for loop
- Using Arrays.copyOf() method
- Using System.arraycopy() method
- Using clone() method

37. Will the below code compile fine? If yes, what will be the output of the following program?

```
public class Test {

public static void main(String[] args)
{

  int[ ] arr;

    arr = new int[1];

    arr = new int[2];

      arr[0] = 20;

      arr[1] = 30;

      arr[2] = 40;

  }

}
```

Ans: Yes, the code will compile fine but at runtime, we will get ArrayIndexOutOfBoundsException because once an array is created, its size cannot be resized.

38. Is there any error in the below code? If not, what will be the output of the following program?

```
public class Test {

public static void main(String[] args)

{

  int[ ] arr = new int[3];

  int[ ] arr2 = new int[5];

   arr2 = arr;

    System.out.println(arr.length);

    System.out.println(arr2.length);
  }

}
```

Ans: Output: 3 3

39. Is it possible to mark an array volatile?

Ans: Yes, we can mark an array *volatile in Java*. But we can make only the reference to array volatile, not the whole array.

40. How can we print elements of an Array in Java?

Ans: Arrays class provides two static methods toString() and deepToString() to print each element of an array.

41. How will we convert an array of String objects into a List?

Ans: Java Arrays class present in java.util package provides a method asList() that accepts an Array as input and returns a List as output. The sample code is as follows:

```
String[ ] myArray = {"George" , "Jack" , "Ryan"};

List<String> myList = Arrays.asList(myArray);

System.out.println(myList);

Output: [George, Jack, Ryan]
```

42. Which new methods were introduced in Java 8 for the processing of Arrays on multi-core machines?

Ans: Java 8 has enhanced the Arrays class by introducing some new methods that can run efficiently on multi-core machines. These methods start with keyword parallel.

For example, Arrays.parallelSetAll(), Arrays.parallelSort(), etc. These methods process Arrays in parallel to make Java code very fast on a multi-core machine.

---

43. Suppose there are two array objects of int type. one has 20 elements and another one has 10 elements. Is it possible to assign an array of 20 elements to an array of 10 elements?

Ans: Yes, it is possible to assign an array of 20 elements to an array of 10 elements provided they should be of the same type. Java compiler checks only the type of the array, not the size while assigning. The sample code is as follows:

```
int[ ] x = new int[10];

int[ ] y = new int[20];

x = y; // Here, array references x and y both are pointing to the same array
contents (i.e. array of 20 elements).
```

44. Identify and fix errors in the below code. What will be output after fixing them?

```
import java.util.Arrays;

public class Test {

public static void main(String[] args)

{

  int num[ ] = new int[6]{1, 2, 3, 4, 5, 6};

    System.out.println(Arrays.toString(num));

  }

}
```

Ans: We need not define the size of the array when we are storing the array contents. After removing the size of array, the output is [1, 2, 3, 4, 5, 6]

45. Can we check the equality of two arrays in java?

Or, how will you compare the two arrays in java?

Ans: Yes, we can compare the equality of two arrays in java. Arrays class provides two methods: Arrays.equals() method to compare one-dimensional array and Arrays.deepEquals() for comparing multidimensional arrays. The sample code is as:

```
String[ ] s1 = {"Ivaan", "John", "Helly", "Ricky"};
```

```
String[ ] s2 = {"Deep", "Shubh", "Herry", "Micky"};

String[ ] s3 = {"Ivaan", "John", "Helly", "Ricky"};

System.out.println(Arrays.equals(s1, s2)); // Output : false

System.out.println(Arrays.equals(s1, s3)); // Output : true
```

46. How will you sort an array of objects?

Ans: To sort an array of objects, use Arrays.sort() method. This method internally uses quick sort algorithm to sort an array of objects. The sample code is as follows:

```
String[] s1 = {"Ivaan", "John", "Helly", "Ricky"};

Arrays.sort(s1);

System.out.println(Arrays.toString(s1));

Output: [Helly, Ivaan, John, Ricky]
```

If you want to sort a list of objects, use Collections.sort() method. Collections internally use the Arrays sorting method. So, both of them have the same performance except that Collections take some time to convert list to an array.

---

47. Can we check an array contains a particular element or not?

Ans: Arrays class provides two methods: isExists() and contains() to check an array has an element or not. Both the methods return true if an array has elements, otherwise returns false. The sample code is as:

```
String[ ] s1 = {"Ivaan", "John", "Helly", "Ricky"};

List<String> nameList = new ArrayList<>(Arrays.asList(s1));

System.out.println(nameList.contains("John")); // true.
```

48. Can we retrieve the class name of an array?

Ans: Yes, we can retrieve the class name of an array using getClass() and getName() methods. The getClass() method of the Object class returns the runtime class of the object.

While the getName() method of the Class class returns the name of the class/array class. The sample code is as:

```
String[] s1 = {"Ivaan", "John", "Helly", "Ricky"};

System.out.println(s1.getClass().getName()); // Output: java.lang.String;
```

49. Which method of the Arrays class can be used to search/find a particular element in an array?

Ans: The Arrays.binarySearch() method can be used to search/find a particular element in an array. This method uses the binary search algorithm.

The array should be in natural ordering (either in ascending or descending order) before calling it. This is the simplest and most efficient approach to find an element in a sorted array.

50. What is the difference between linear search and binary search in java?

Ans: Binary search is much faster than linear search for a large array. Binary search algorithm is mostly used for large size array whereas, linear search is used for very small size array.

---

51. Which kind of operations can be performed on an array?

Ans: There are several operations that can be performed on array. They are:

- Searching
- Sorting
- Traversal
- Deletion
- Insertion

52. What are the simple approaches to find duplicate elements in an array?

Ans: There are mainly three simple approaches to find duplicate elements from an array.

a) **Brute Force Method**: In this approach, each element of an array is compared with elements of another array. If any two elements are found equivalent, we consider them as duplicates. The time complexity of this approach is O(n^2).
b) **Using HashSet**: In this approach, we use the HashSet class to find the duplicate elements in an array. To find the duplicate elements, first iterate over the array elements and then add them into HashSet by calling add() method of the HashSet class.
If the method returns false it means that the element is already present in the Set. It takes O(n) time to find the duplicate elements.

c) **Using HashMap**: We can also use HashMap to find duplicate elements in an array. HashMap uses key-value pair to store an element.
To find the duplicate array element, we store elements of the array as keys and their occurrence as values. If the value of any key is greater than 1, the key is considered a duplicate element. Its time and space complexity is O(n).

53. Which of the following are valid declarations?

```
a. String[ ] green = new String[ ] {"green"};

b. String[ ] red = new String[1] {"red"};

c. String bear [ ] = new String[ ] {};

d. String[ ] obj = new String[ ] {};
```

Ans: a, c, d, are valid declarations.

54. Which of the following are valid declarations of array?

```
a. String[ ][ ][ ] a = new String[ ][2][ ];

b. int[ ][ ][ ] b = new int[3][3][ ];

c. float[ ][ ][ ] c = new float[3][ ][5];

d. Object[ ][ ] obj = new Object[1][ ];
```

Ans: b and d are valid declarations of arrays because while creating the multidimensional arrays, we cannot specify an array dimension after an empty dimension. It generates compile-time error.

55. Identify and fix the error in the following lines of code.

```
int x;

int[ ] num = {2, 4, 6, x, 10};
```

Ans: There is an error in the first line of code because the variable x is not initialized.

Hope that this tutorial has covered almost all the important **array interview questions in java** with the best possible answers. I hope that you will have practiced array coding programming questions.
All the best!!!
