



# INTERVIEW QUESTIONS

## ARRAY

**Softech Solutions Inc.**

[www.softechsolutionsgroup.com](http://www.softechsolutionsgroup.com)

[saney.alam@softechsolutionsgroup.com](mailto:saney.alam@softechsolutionsgroup.com)

# Interview Questions

Q1: If you are given a choice to use either ArrayList or LinkedList, Which one would you use and Why?

Ans. ArrayList is implemented in memory as arrays and hence allows fast retrieval through indices but are costly if new elements are to be inserted in between other elements. LinkedList allows for constant-time insertions or removals using iterators, but only sequential access of elements.

1. Retrieval - If Elements are to be retrieved sequentially only, Linked List is preferred.
2. Insertion - If new Elements are to be inserted in between other elements, Array List is preferred.
3. Search - Binary Search and other optimized way of searching is not possible on Linked List.
4. Sorting - Initial sorting could be pain but lateral addition of elements in a sorted list is good with linked list.
5. Adding Elements - If sufficiently large elements need to be added very frequently, Linked List is preferable as elements don't need consecutive memory location.

Q2: What is the difference between ArrayList and LinkedList ?

Ans. Underlying data structure for ArrayList is Array whereas LinkedList is the linked list and hence have following differences -

1. ArrayList needs continuous memory locations and hence need to be moved to a bigger space if new elements are to be added to a filled array which is not required for LinkedList.
2. Removal and Insertion at specific place in ArrayList requires moving all elements and hence leads to  $O(n)$  insertions and removal whereas its constant  $O(1)$  for LinkedList.
3. Random access using index in ArrayList is faster than LinkedList which requires traversing the complete list through references.

4. Though Linear Search takes Similar Time for both, Binary Search using LinkedList requires creating new Model called Binary Search Tree which is slower but offers constant time insertion and deletion.

5. For a set of integers you want to sort using quicksort, it's probably faster to use an array; for a set of large structures you want to sort using selection sort, a linked list will be faster.

Q3: Difference between Vector and ArrayList?

Ans. Vectors are synchronized whereas Array lists are not.

Q4: What are the disadvantages of using arrays ?

Ans. Arrays are of fixed size and have to reserve memory prior to use. Hence if we don't know size in advance arrays are not recommended to use.

Arrays can store only homogeneous elements.

Arrays store its values in contentious memory location. Not suitable if the content is too large and needs to be distributed in memory.

There is no underlying data structure for arrays and no ready-made method support for arrays, for every requirement we need to code explicitly.

Q5: What are the Disadvantages of using Collection Classes over Arrays ?

Ans. Collections can only hold objects; it can't hold primitive data types.

Collections have performance overheads as they deal with objects and offer dynamic memory expansion. This dynamic expansion could be a bigger overhead if the collection class needs consecutive memory location like Vectors.

Collections don't allow modification while traversal as it may lead to `ConcurrentModificationException`.

Q6. Will this code give error if I try to add two heterogeneous elements in the ArrayList? And Why?

```
List list1 = new ArrayList<>>();  
list1.add(5);  
list1.add("5");
```

Ans. If we don't declare the list to be of specific type, it treats it as list of objects. `int 1` is auto boxed to Integer and `"1"` is String and hence both are objects.

Q7: What is the advantage of using arrays over variables?

Ans. Arrays provide a structure wherein multiple values can be accessed using single reference and index. This helps in iterating over the values using loops.

Q8: Advantage of Collection classes over Arrays?

Ans. Collections are re-sizable in nature. We can increase or decrease the size as per requirement.

Collections can hold both homogeneous and heterogeneous data's.

Every collection follows some standard data structures.

Collection provides many useful built in methods for traversing, sorting and search.

Q9: Difference between ArrayList and LinkedList ?

Ans. LinkedList and ArrayList are two different implementations of the List interface. LinkedList implements it with a doubly-linked list. ArrayList implements it with a dynamically resizing array.

Q10: What is the difference between `int[] x;` and `int x[];` ?

Ans. No Difference. Both are the acceptable ways to declare an array.

Q11: What is CopyOnWriteArrayList ?

Ans. Its a type of ArrayList in which all Write operations , i.e add and set are performed by creating a new copy. This array never changes during the lifetime of the iterator, so it never throws ConcurrentModificationException.

Q12: What are the advantages and disadvantages of CopyOnWriteArrayList ?

Ans. This collections class has been implemented in such a manner that it can never throw ConcurrentModificationException. As it performs update and write operations by creating a new copy of ArrayList, It's slower compared to ArrayList.

Q13: What is ArrayIndexOutOfBoundsException ?

Ans. Exception thrown by the application is we try to access an element using an index which is not within the range of array i.e lower than 0 or greater than the size of the array.

Q14: What will the following code result?

```
List> intList = new ArrayList<>();
```

```
intList.add(Optional.empty());
```

```
intList.add(Optional.of(2));
```

```
intList.add(Optional.of(3));
```

```
intList.add(Optional.of(4));
```

```
System.out.println(intList.get(null));
```

Ans. Compile time error at last line as the get method expects argument of type native int.

Q15: What will the following code result ? Will it compile ?

```
List > intList = new ArrayList > ();  
intList.add(Optional.empty());  
intList.add(Optional.of(2));  
intList.add(Optional.of(3));  
intList.add(Optional.of(4));  
  
System.out.println(intList.get((Integer)null));
```

Ans. Yes but the last line will throw NullPointerException upon execution.

Q16: Can we add more elements to an array list that has been marked as final?

Ans. Yes, the array list can hold more elements. Final only puts the restriction that the array list reference cannot hold any other array list.

Q17: Does an ArrayList allow elements of different types ? If not, why the following code works

```
List list1 = new ArrayList<>();  
list1.add(1);  
list1.add("1");
```

Ans. With Java 7 or Later. If you don't declare the list to be of specific type , it treats it as list of objects. int 1 is auto boxed to Integer and "1" is String and hence both are objects.

Q18: Is array an object in Java ? How can you prove that ?

Ans. Yes.

There are 2 ways this can be confirmed

1. Accessing object class methods using array reference.
2. Checking if the array is an instance of Object class  
if (arrayRef instanceof Object).

Q19: How can we get an array out of ArrayList?

Ans. We can get an array out of ArrayList by using toArray() method of an array list.  
String a[] = arrayList.toArray();

Q20: Why iterators of an array list are fail fast?

Ans. Because it's access isn't synchronized and hence access / modification by multiple threads may lead to inconsistent state.

Q21: What is meant by "Vector is synchronized and ArrayList isn't " ?

Ans. It means that only 1 thread can access have access to Vector at a time and no parallel access is allowed whereas Array List allows parallel access by multiple threads.

Q22: What problem we could have with ArrayList which aren't possible with Vectors ?

Ans. ArrayLists aren't synchronized and hence doesn't allow synchronized access. As multiple threads can access an arraylist in parallel, it may result in an inconsistent state.

Q23: What is the quickest way to find count of duplicate elements in an arraylist, without using iteration or loops?

Ans. We can copy the elements to a Set and then find the difference of count between ArrayList and Set. As Set doesn't allow duplicates, they will be removed in the set.

Q24: Why do we pass an array of strings to main method ?

Ans. Array of strings in the main method are the list of arguments or parameters which are sent to the application / program.

Q25: What does the following initialization mean?

```
ArrayList<LinkedList> traversalPaths = new ArrayList<LinkedList>();
```

What could be the use of such a collection?

Ans. Initialize an ArrayList that will hold LinkedLists i.e every element of the arraylist will be a linked list.

Such collection could be used in algorithms that require first random access and then sequential traversal. For example - Storing traversal paths for a graph wherein we can start from any vertex. Implementing dictionary with each arraylist element holding starting with character and then linked list holding duplicate words.

Q26: How can we convert String into Char array and vice versa ?

Ans. There is a method `toCharArray()` within String class that can be used to convert string to char array.

```
string.toCharArray();
```

String class has an argument constructor that takes a char array and create a string

```
new String(charArray);
```

Q27: How can we convert a character or a character array into a String?

Ans. String has an argument constructor that takes char array as argument and creates a string.

There is no constructor available with String that takes in a character and creates a String. We can use `StringBuilder` which has a char argument constructor.