Interview Questions

JAVA-111: Session 9

Answering interview questions is crucial in your journey of applied learning. Review them to ensure your understanding of important topics covered in the previous session and to prepare yourself for upcoming challenges. Remember that it's important to answer these questions on your own before viewing the solution. The solutions are hyperlinked to community posts on our platform.

**Note: The questions below have been sourced from previous interviews**

Questions

1. [What is the difference between Array and ArrayList?](https://crio.do/learn/crio-community/topic/what-is-the-difference-between-array-and-arraylist/169161)
2. [How do you sort an ArrayList in descending order?](https://crio.do/learn/crio-community/topic/how-do-you-sort-an-arraylist-in-descending-order/188124)
3. What is the difference between ArrayList and LinkedList?
4. [How does HashMap work internally and how does it handle collisions?](https://crio.do/learn/crio-community/topic/how-does-hashmap-work-internally-and-how-does-it-handle-collisions/252739)
5. [What is the Java Collection Framework?](https://crio.do/learn/crio-community/topic/what-is-the-java-collection-framework/252773)
6. [What are the differences between HashSet and TreeSet?](https://crio.do/learn/crio-community/topic/what-are-the-differences-between-hashset-and-treeset/252780)
7. [What are the main differences between HashMap and ConcurrentHashMap in Java? In what scenarios would you use one over the other?](https://crio.do/learn/crio-community/topic/what-are-the-main-differences-between-hashmap-and-concurrenthashmap-in-java-in-what-scenarios-would-you-use-one-over-the-other/252781)
8. [What are the differences between a HashMap and a TreeMap in Java?](https://crio.do/learn/crio-community/topic/what-are-the-differences-between-a-hashmap-and-a-treemap-in-java/252816)
9. [What is a Java Iterator and how is it used?](https://crio.do/learn/crio-community/topic/what-is-a-java-iterator-and-how-is-it-used/252817)
10. [How does HashMap work internally?](https://crio.do/learn/crio-community/topic/how-does-hashmap-work-internally/252832)
11. [What is a List in Java and what are its types?](https://crio.do/learn/crio-community/topic/what-is-a-list-in-java-and-what-are-its-types/252833)
12. [What is a HashSet and how is it different from a List?](https://crio.do/learn/crio-community/topic/what-is-a-hashset-and-how-is-it-different-from-a-list/252883)
13. [In what scenarios would you prefer using an ArrayList over a LinkedList and vice versa?](https://crio.do/learn/crio-community/topic/in-what-scenarios-would-you-prefer-using-an-arraylist-over-a-linkedlist-and-vice-versa/252889)
14. [How do you convert an ArrayList to a LinkedList and vice versa?](https://crio.do/learn/crio-community/topic/how-do-you-convert-an-arraylist-to-a-linkedlist-and-vice-versa/252890)
15. [Explain the concept of wrapper classes in Java.](https://crio.do/learn/crio-community/topic/explain-the-concept-of-wrapper-classes-in-java/252914)
16. [What is autoboxing and unboxing in Java?](https://crio.do/learn/crio-community/topic/what-is-autoboxing-and-unboxing-in-java/252915)
17. [How do you iterate over elements in an ArrayList? Provide a simple code example.](https://crio.do/learn/crio-community/topic/how-do-you-iterate-over-elements-in-an-arraylist-provide-a-simple-code-example/252943)

**1. What is the difference between Array and ArrayList?**

| **Feature** | **Array** | **ArrayList** |
| --- | --- | --- |
| Size | Fixed | Dynamic |
| Type | Can store primitives & objects | Only stores objects |
| Performance | Faster (less overhead) | Slightly slower (dynamic resizing) |
| Flexibility | Less flexible | More flexible (resizable, many methods) |

**2. How do you sort an ArrayList in descending order?**

java

CopyEdit

import java.util.\*;

ArrayList<Integer> list = new ArrayList<>(Arrays.asList(5, 2, 9, 1));

list.sort(Collections.reverseOrder()); // Descending sort

System.out.println(list); // Output: [9, 5, 2, 1]

**3. What is the difference between ArrayList and LinkedList?**

| **Feature** | **ArrayList** | **LinkedList** |
| --- | --- | --- |
| Storage | Dynamic array | Doubly linked list |
| Access Time | Fast (index-based) | Slower (traverse from head/tail) |
| Insert/Delete | Slower (shifting required) | Faster (just re-link nodes) |
| Best Use Case | Frequent access by index | Frequent add/remove operations |

**4. How does HashMap work internally and how does it handle collisions?**

* Uses an **array of buckets** (hash table).
* Each key's hashCode() is used to find the bucket index.
* If two keys have the same hash, a **collision** occurs.
* Collisions are handled using:
  + **Linked list** (Java 7 and earlier)
  + **Balanced tree (Red-Black Tree)** (Java 8+) if bucket size exceeds threshold.

**5. What is the Java Collection Framework?**

* A **unified architecture** to store and manipulate groups of objects.
* Includes:
  + Interfaces (List, Set, Map, Queue)
  + Implementations (ArrayList, HashSet, HashMap, etc.)
  + Algorithms (sorting, searching)
  + Utilities (Collections class)

**6. What are the differences between HashSet and TreeSet?**

| **Feature** | **HashSet** | **TreeSet** |
| --- | --- | --- |
| Order | No guarantee | Sorted (natural or custom) |
| Performance | Faster | Slower (uses Red-Black Tree) |
| Null | Allows one null | Doesn’t allow null in comparator-based TreeSet |

**7. Differences between HashMap and ConcurrentHashMap?**

| **Feature** | **HashMap** | **ConcurrentHashMap** |
| --- | --- | --- |
| Thread-safe? | No | Yes |
| Performance (multi-threading) | Poor | Excellent |
| Null keys/values | 1 null key, many null values | No null keys/values |
| Use Case | Single-threaded | Multi-threaded |

**8. Differences between HashMap and TreeMap?**

| **Feature** | **HashMap** | **TreeMap** |
| --- | --- | --- |
| Order | Unordered | Sorted (natural or custom) |
| Performance | Faster (O(1)) | Slower (O(log n)) |
| Null Key | Allows one null key | Doesn’t allow null key |

**9. What is a Java Iterator and how is it used?**

* An Iterator allows you to **loop through collections**.
* Use hasNext() and next().

java

CopyEdit

Iterator<String> it = list.iterator();

while (it.hasNext()) {

System.out.println(it.next());

}

**10. How does HashMap work internally?**

(*See Q4 above; same answer*)

**11. What is a List in Java and what are its types?**

* List is an interface that represents **ordered collections**.
* Types:
  + ArrayList
  + LinkedList
  + Vector
  + Stack

**12. What is a HashSet and how is it different from a List?**

| **Feature** | **HashSet** | **List** |
| --- | --- | --- |
| Duplicates | Not allowed | Allowed |
| Order | Unordered | Ordered |
| Access | No index access | Index-based access |

**13. When to prefer ArrayList vs LinkedList?**

* **Use ArrayList**:
  + If you do lots of **index-based access**.
* **Use LinkedList**:
  + If you do lots of **insertions/deletions** in the middle.

**14. Convert ArrayList to LinkedList and vice versa**

java

CopyEdit

ArrayList<String> al = new ArrayList<>();

LinkedList<String> ll = new LinkedList<>(al); // AL → LL

LinkedList<String> ll2 = new LinkedList<>();

ArrayList<String> al2 = new ArrayList<>(ll2); // LL → AL

**15. What are Wrapper Classes in Java?**

* Wrapper classes are object versions of primitives.
* Example:
  + int → Integer
  + double → Double
  + char → Character
* Needed for collections (since collections only store objects).

**16. What is Autoboxing and Unboxing?**

* **Autoboxing**: Convert primitive → wrapper class  
  int x = 5; Integer y = x;
* **Unboxing**: Convert wrapper → primitive  
  Integer y = 5; int x = y;

**17. How do you iterate over elements in an ArrayList?**

**🔁 Method 1: For loop**

java

CopyEdit

for (int i = 0; i < list.size(); i++) {

System.out.println(list.get(i));

}

**🔁 Method 2: Enhanced for loop**

java

CopyEdit

for (String item : list) {

System.out.println(item);

}

**🔁 Method 3: Using Iterator**

java

CopyEdit

Iterator<String> it = list.iterator();

while (it.hasNext()) {

System.out.println(it.next());

}