**1. Conceptual Questions**

1. **What is the difference between ArrayList and LinkedList?**
2. **Why does HashSet not allow duplicate values?**
3. **How does HashMap work internally?**
4. **What is the difference between HashMap, LinkedHashMap, and TreeMap?**
5. **Why are HashMap keys immutable?**
6. **Why is Hashtable considered legacy?**
7. **How does ConcurrentHashMap prevent concurrent modification?**
8. **What is the difference between Collections.synchronizedMap() and ConcurrentHashMap?**
9. **Why does ConcurrentModificationException occur, and how do you avoid it?**
10. **What is the difference between fail-fast and fail-safe iterators?**
11. **Why is TreeSet faster for range queries than HashSet?**
12. **How does WeakHashMap work, and when is it useful?**
13. **What is the difference between IdentityHashMap and HashMap?**
14. **How does CopyOnWriteArrayList work, and why is it useful?**
15. **How does PriorityQueue maintain order internally?**

**2. Coding Questions**

**Easy**

1. **Remove duplicates from a List<Integer> while maintaining order.**
2. **Find the first non-repeating character in a String using Map.**
3. **Sort a List<String> based on length using Comparator.**
4. **Implement a custom LinkedList class with add(), remove(), and get() methods.**

**Medium**

1. **Find the most frequent element in an ArrayList<Integer>.**
2. **Design a LRU (Least Recently Used) cache using LinkedHashMap.**
3. **Merge two sorted ArrayLists into a single sorted list without duplicates.**
4. **Find the top 3 repeated words in a paragraph using Map.**
5. **Reverse a LinkedList without using extra space.**
6. **Check if two HashMaps are equal (same keys and values).**

**Hard**

1. **Implement a thread-safe CustomHashMap without using ConcurrentHashMap.**
2. **Write a custom ImmutableList that does not allow modifications.**
3. **Implement a BlockingQueue using synchronized and wait/notify.**
4. **Find the longest subsequence of unique numbers in a list.**
5. **Design an in-memory key-value store with TTL (Time-to-Live) support using ConcurrentHashMap.**

**3. Deep-Dive Questions on Internals**

**ArrayList vs. LinkedList**

* How does ArrayList dynamically grow?
* What is the time complexity of inserting at the beginning of ArrayList vs. LinkedList?
* Why is ArrayList preferred for read-heavy operations?

**HashMap Internal Working**

* How does HashMap handle collisions?
* What happens if two objects have the same hashCode() but different equals()?
* What changes were introduced in HashMap in Java 8?

**Concurrency & Fail-Fast**

* What happens if multiple threads modify a HashMap concurrently?
* How does ConcurrentHashMap divide data internally?
* How does CopyOnWriteArrayList ensure thread safety?
* Why does ConcurrentModificationException occur?

**4. Real-World Scenario Questions**

1. **How would you implement an in-memory database using Map?**
2. **Which collection would you use for an autocomplete system?**
3. **How would you efficiently store and retrieve a million unique users?**
4. **How would you count word frequency in a large file using Map?**
5. **Which collection would you use to implement a stock price ticker that keeps track of the latest 100 prices?**