Java Collections Framework – Interview Cheat Sheet

■ Basics

Q1. Collection vs Collections? -> Collection = Interface, Collections = Utility class.
Q2. Root interface? -> Collection (List, Set, Queue) & Map.
Q3. List vs Set vs Map? -> List: ordered+duplicates, Set: no duplicates, Map: key-value.

■ List

Q4. ArrayList vs LinkedList? -> AL: fast random access, LL: fast insert/delete.
Q5. Vector vs ArrayList? -> Vector: synchronized, ArrayList: not synchronized.
Q6. Initial capacity? -> 10, grows 50%.

■ Set

Q7. HashSet vs LinkedHashSet vs TreeSet? -> HS: no order, LHS: insertion order, TS: sorted.
Q8. Why no duplicates? -> Uses hashCode + equals.
Q9. Comparable vs Comparator? -> Comparable: natural order, Comparator: custom order.

■ Map

Q10. HashMap vs Hashtable? -> HM: not sync, allows nulls; HT: sync, no nulls.
Q11. HashMap vs LinkedHashMap vs TreeMap? -> HM: no order, LHM: insertion order, TM: sorted.
Q12. How HashMap works? -> hashCode -> bucket -> collision via list/tree.

■ Queue

Q13. Queue vs Deque? -> Queue: FIFO, Deque: double-ended.
Q14. PriorityQueue? -> Orders by natural/comparator.

■ Iterators

Q15. Iterator vs ListIterator? -> I: forward only, LI: forward+backward.
Q16. Fail-fast vs Fail-safe? -> FF: exception, FS: works on copy.

■ Concurrency

Q17. Thread-safe collection? -> synchronizedList or ConcurrentHashMap.
Q18. SynchronizedMap vs ConcurrentHashMap? -> SM: full lock, CHM: bucket lock.

■ Most Important

Q19. Why override hashCode+equals? -> Ensures uniqueness in HashMap/HashSet.
Q20. modCount? -> Tracks modifications, used in fail-fast.