1. Why String is immutable in Java?

👉 *Interview answer:*
"String is immutable mainly for security and performance reasons. Since String is used in class loading, database URLs, and file paths, immutability prevents unauthorized changes. It also allows String constant pool reuse, makes String thread-safe, and enables hashCode caching which is important when Strings are used as keys in HashMap."

---

2. Difference between `==` and `equals()`?

👉 *Interview answer:*
"`==` compares references, meaning it checks whether both variables point to the same object. `equals()` compares content. In String class, `equals()` is overridden to compare actual values."

---

3. How does HashMap work internally?

👉 *Interview answer:*
"HashMap works based on hashCode and equals methods. First, hashCode of the key is calculated and converted into a bucket index. If the bucket is empty, the entry is stored directly. If there is a collision, entries are stored as a linked list or a red-black tree from Java 8. While retrieving, equals is used to identify the correct key."

---

4. What if two keys have the same hashCode?

👉 *Interview answer:*
"If two keys have the same hashCode, they go into the same bucket. HashMap then uses equals() to check key equality. From Java 8, if collisions increase, the linked list is converted into a balanced red-black tree to improve performance."

---

5. Why override hashCode when overriding equals?

👉 *Interview answer:*
"Because HashMap uses hashCode to find the bucket. If two objects are equal but have different hashCodes, they may be stored in different buckets, breaking the HashMap contract and causing unexpected behavior."

---

6. ArrayList vs LinkedList?

👉 *Interview answer:*
"ArrayList uses a dynamic array, so access is fast but insertion and deletion are slower. LinkedList uses a doubly linked list, so insertion and deletion are faster but access is slower. In most cases, ArrayList is preferred."

---

7. HashMap vs ConcurrentHashMap?

👉 *Interview answer:*
“HashMap is not thread-safe. ConcurrentHashMap is thread-safe and uses segment-level locking, allowing multiple threads to read and write concurrently. Also, ConcurrentHashMap does not allow null keys or values.”

---

8. HashMap vs Hashtable?

👉 *Interview answer:*
“Hashtable is synchronized and thread-safe but slower. HashMap is not synchronized and performs better. Hashtable is legacy and does not allow null keys or values.”

---

9. What is Garbage Collection?

👉 *Interview answer:*
“Garbage Collection is an automatic process managed by the JVM to identify unused objects and free heap memory. Developers cannot explicitly destroy objects in Java.”

---

10. When is an object eligible for GC?

👉 *Interview answer:*
“An object becomes eligible for garbage collection when there are no active references pointing to it, such as when it is set to null, reassigned, or goes out of scope.”

---

11. Difference between throw and throws?

👉 *Interview answer:*
“`throw` is used to explicitly throw an exception object, while `throws` is used in the method signature to declare exceptions that may occur.”

---

12. Can finally block be skipped?

👉 *Interview answer:*
“Generally no, but it can be skipped if the JVM terminates abruptly, like when System.exit() is called or in case of a JVM crash.”

---

13. Checked vs Unchecked exception?

👉 *Interview answer:*
"Checked exceptions are verified at compile time and must be handled, like IOException. Unchecked exceptions occur at runtime, such as NullPointerException."

---

14. What is exception propagation?

👉 *Interview answer:*
"If an exception is not handled in the current method, it propagates to the calling method until it is caught or the program terminates."

---

15. Why multiple inheritance not supported in Java?

👉 *Interview answer:*
"Java does not support multiple inheritance with classes to avoid ambiguity known as the Diamond Problem. However, Java supports multiple inheritance through interfaces."

---

16. Abstract class vs Interface?

👉 *Interview answer:*
"Abstract classes can have instance variables and constructors. Interfaces support multiple inheritance. From Java 8 onwards, interfaces can have default and static methods."

---

17. Can we override static methods?

👉 *Interview answer:*
"No, static methods belong to the class, not the object. What we get is method hiding, not method overriding."

---

18. Why main method is static?

👉 *Interview answer:*
"The main method is static so that JVM can call it without creating an object of the class."

---

19. What is synchronization?

👉 *Interview answer:*
"Synchronization is used to control access to shared resources by multiple threads, ensuring data consistency."

---

20. sleep() vs wait()?

👉 *Interview answer:*
"sleep() pauses the current thread without releasing the lock, whereas wait() releases the lock and waits until notified."

---

21. What is deadlock?

👉 *Interview answer:*
"Deadlock occurs when two or more threads wait indefinitely for each other's resources, causing the program to freeze."

---

22. How to avoid deadlock?

👉 *Interview answer:*
"We can avoid deadlock by maintaining proper lock ordering, avoiding nested locks, and minimizing synchronized blocks."

---

23. What is volatile keyword?

👉 *Interview answer:*
"Volatile ensures visibility of variable changes across threads by preventing caching, but it does not guarantee atomicity."

---

24. Runnable vs Thread?

👉 *Interview answer:*
"Runnable is preferred because it supports multiple inheritance and provides better design flexibility than extending Thread."

---

25. What are Java 8 Streams?

👉 *Interview answer:*
"Streams are used for processing collections in a functional style. They support operations like filter, map, and reduce and do not modify the original data."

---

26. map() vs flatMap()?

👉 *Interview answer:*
"map() transforms each element individually, while flatMap() flattens nested structures into a single stream."

---

27. What is Functional Interface?

👉 *Interview answer:*
"A functional interface has exactly one abstract method and is used with lambda expressions, like Runnable or Comparator."

---

28. What is Optional?

👉 *Interview answer:*
"Optional is a container object introduced in Java 8 to avoid NullPointerException by explicitly handling absence of values."

---

29. What is serialization?

👉 *Interview answer:*
"Serialization is the process of converting an object into a byte stream so it can be stored or transmitted over a network."

---

30. Comparable vs Comparator?

👉 *Interview answer:*
"Comparable is used for natural ordering using compareTo, while Comparator is used for custom sorting using compare."