Java Interview Questions and Answers

1. What is ClassLoader?

ClassLoader is a part of the Java Runtime Environment that dynamically loads classes into the JVM at runtime.

2. How to create a custom ClassLoader?

Extend the **ClassLoader** class and override the **findClass(String name)** method to define your own class loading logic.

3. What is Delegation Model?

The **Delegation Model** in Java ClassLoader means each ClassLoader delegates the class loading request to its parent before attempting to load the class itself.

4. What is Serializable and transient keyword?

Serializable allows an object to be converted into a byte stream. **transient** is used to skip specific fields during serialization.

5. Why/where to use Serializable and transient (real-life)?

In real-life, Serializable is used for saving state (e.g., games, sessions). **transient** is used to avoid saving sensitive data like passwords.

6. What is an Immutable Class?

A class is **immutable** if its objects cannot be changed once created. Example: **String** class in Java.

7. What is try-with-resources?

It is a try block that automatically closes resources like streams or readers when the try block exits - even if exceptions occur.

8. How many ways to create a thread?

Java Interview Questions and Answers

1. Extend **Thread** class
2. Implement **Runnable** interface
3. Use **ExecutorService**

9. What is Callable and its return type?

Callable<T> is a functional interface that returns a result (generic) and can throw a checked exception.

10. What is ExecutorService?

ExecutorService is a framework that provides thread pool management and task execution methods.

11. How to create thread pool using ExecutorService?

Use **Executors.newFixedThreadPool(int n)** or other factory methods.

12. Simple definition of ExecutorService:

It manages multiple threads efficiently using pools, avoiding manual thread creation.

13. Methods in ExecutorService:

submit(), invokeAll(), invokeAny(), shutdown(), shutdownNow()

14. What is map and flatMap in Stream?

map() transforms elements; **flatMap()** flattens nested structures then transforms.

15. What is Lambda Function?

A **lambda** is a short block of code that takes input and returns output. Syntax: `(a, b) -> a + b`

16. Functional Interfaces (Consumer, Supplier, etc.)

Yes, **Consumer, Supplier, Predicate, Function, BiFunction** are all functional interfaces.

Java Interview Questions and Answers

17. Hashtable vs HashMap

Hashtable is synchronized, slower. **HashMap** is not synchronized, faster.

18. HashMap vs LinkedHashMap vs TreeMap

HashMap: no order

LinkedHashMap: maintains insertion order

TreeMap: sorted by keys

19. ConcurrentHashMap vs Hashtable

Both are thread-safe. **ConcurrentHashMap** allows concurrent reads & partial locks - faster than Hashtable.

20. Map vs ConcurrentHashMap

Map is an interface. **ConcurrentHashMap** is a thread-safe implementation.

21. TreeSet vs TreeMap

TreeSet stores sorted unique elements. **TreeMap** stores key-value pairs sorted by keys.

22. PriorityQueue in Java

A queue that retrieves elements based on priority, **not FIFO**.

23. Enumeration vs Iterator

Enumeration is legacy (read-only). **Iterator** supports remove and modern iteration.

24. Fail-Fast vs Fail-Safe

Fail-Fast throws error immediately on modification. **Fail-Safe** works on a copy and does not throw errors.

Java Interview Questions and Answers

25. Comparable vs Comparator

Comparable uses compareTo() for natural order. **Comparator** uses compare() for custom sorting.