

Compiled List of Java Interview question by Umesh Kaushik

Connect Here for all Doubts or Guidance : [Click to Connect](#)

Java Core Concepts:

1. **Platform Independence (JVM, JRE):**
 - How is Java Platform Independent (drilled down to basics)?
 - JVM, JRE differences?
2. **Class Loading:**
 - What is a ClassLoader? How is it used while running the program in Java?
 - ClassNotFoundException vs NoClassDefFoundError difference?
 - Why ClassNotFoundException is found at runtime only?
3. **String Pool & String Handling:**
 - What is the String Pool? Where is it present?
 - What is the `intern()` function?
 - String literals difference between `==` and `equals()`?
 - String objects difference between `==` and `equals()`?
4. **Object-Oriented Programming:**
 - What is Encapsulation?
 - What is a Serializable interface?
 - Can a Java class be static/final/private?
 - What is Object-Oriented Programming (OOP)?
 - Difference between OOP and scripting language?
5. **Java Exceptions:**
 - What is the super class of all exceptions?
 - What is the difference between error and exception?
 - Checked vs unchecked exceptions?
 - What is try-with-resources block?
 - What is Exception propagation in Java?
6. **Java Language Features:**
 - What is AutoBoxing?
 - What are Java 8 features?
 - What is a lambda function, and how is it useful?
 - How does indexing work internally?
 - Can a constructor be present in an interface? Why not?
 - Performance difference between traditional `for` loop and enhanced `for-each` loop?

Concurrency and Multithreading:

7. **Threads and Synchronization:**
 - Write code to work with 5 threads printing data one by one.
 - Difference between `volatile` and `atomic`?

Design Patterns:

8. Design Patterns:

- What is the Builder Pattern? Write code for it.
- Write code for Singleton Pattern.
- What is the concept of the Factory Pattern? (Mention dissatisfaction in previous interview)
- Why do we need design patterns?
- How do design patterns help simplify programming?
- What design patterns are you familiar with?
- How would you invent a new design pattern?

Spring & Spring Boot:

9. Spring Core:

- What is Spring Boot? Why is it popular?
- What are the Spring annotations (e.g., @Qualifier, @Value, @Bean, @Component)?
- What is the difference between @Controller and @RestController?
- What is the difference between @Mock and @InjectMock?
- What is the difference between @RestController and @Controller?

10. Spring Security:

- How does OAuth work?
- Implementing role-based access using Spring Security.

11. Spring Data JPA:

- What are the three interfaces of Spring Data JPA?

12. Error Handling:

- How to handle exceptions in the controller and return proper status code?

13. Transaction Management:

- What is transaction propagation in Spring Boot?

14. Miscellaneous:

- What is the application.properties file?
- How does Spring/Hibernate automatically generate SQL queries?
- What is MVC? Differences between Controller, Service, and Repository?
- How to do pagination in Spring Boot with SQL database?
- How to create unique constraints in a database entity class?

Java Collections:

15. Collections API:

- How does HashMap work?

- HashMap vs HashTable?
- HashMap vs ConcurrentHashMap?
- Difference between List, Set, Map? When to choose what?
- What is LinkedList?
- What is the Comparable interface?
- How is hashCode generated for an object?
- How many collection implementations do you know?
- What is the Iterable interface?

Java Streams:

16. Streams API:

- What is the Streams API?
- How many ways are there to create streams in Java?
- Difference between Streams and Collections?
- What are the two types of operations in the Stream API?
- Say methods that you used from Streams?
- Difference between Map and FlatMap?
- What is the `collect()` method in streams?
- Find First vs Find Any in Streams?
- How to handle exceptions in streams?
- How to convert a Stream into a Collection?
- Difference between sequential and parallel streams? Any effect?

Java Functional Programming:

17. Functional Interfaces:

- What is a functional interface?
- Can a functional interface contain default and static methods?
- Example of a functional interface on paper?

Microservices and Distributed Systems:

18. Microservices vs Monolithic Architecture:

- Differences between Microservices and Monolithic services?

19. Distributed Systems:

- What is the CAP theorem? Which trade-off is important for distributed systems?

Other Java Topics:

20. Singleton/Factory Pattern:

- Write pseudo code for Singleton.
- Factory Pattern - refine code.

21. Immutability:

- How to make a class immutable?

22. Miscellaneous:

- PUT vs POST vs PATCH?
- REST API URL naming convention?
- Maven scopes (compile, runtime, test, etc.)?
- What versions of Java are you familiar with?
- Differences between Java versions?

Employee-Related Coding Problems:

23. Employee Data Processing:

- Sort employee list by highest salary first and collect into a new list.
- Find employees with salaries more than a given number.
- Find average age of employees whose salary is more than X.
- Top 3 highest-paid employees.
- Find the total number of employees from X city.
- Find the sum of salary from a given list of employees.
- Extract address list from the employee list and save it into another list.
- Group employees by their city?

Troubleshooting and Debugging:

24. Troubleshooting:

- What would be your troubleshooting steps when a backend service is not accessible after deployment?