Make sure you prepare answers to the following 20 questions before going for an interview 👇  
  
1. Monolith vs Microservices differences and when to choose which.  
2. How to design a microservice from scratch.  
3. API Gateway pattern and its advantages.  
4. Inter-service communication: REST vs Messaging.  
5. Circuit Breaker pattern and its implementation using Resilience4j.  
6. Load balancing in microservices using Spring Cloud Load Balancer.  
7. How Spring Cloud Config helps in centralized configuration management.  
8. Service discovery using Eureka or Consul.  
9. Feign Client vs WebClient: Which one to use and why.  
10. Event-driven architecture and Kafka integration.  
11. Database per service vs Shared Database: Pros and cons.  
12. Saga Pattern for distributed transactions in microservices.  
13. JWT-based authentication and OAuth2 in microservices.  
14. How to handle security in an API Gateway.  
15. Observability: Logging, tracing, and monitoring best practices.  
16. Role of Prometheus and Grafana in microservices monitoring.  
17. Kubernetes deployment strategies for microservices.  
18. Blue-Green and Canary deployments in microservices.  
19. When to use WebFlux for reactive microservices.  
20. CQRS and Event Sourcing: When and why to use them.

🚀 ava Streams Question in an Interview! 🚀  
 Tech Mahindra interview, asked Java Streams question! Exciting moment! 🎯

💡 The Problem Statement:  
 Given a list [2,3,2,3,5,2,20,0], find the duplicate values efficiently using Java Streams.

🔥Solution Using Streams API:  
import java.util.\*;  
import java.util.stream.Collectors;  
  
public class FindDuplicates {  
 public static void main(String[] args) {  
 List<Integer> numbers = Arrays.asList(2, 3, 2, 3, 5, 2, 20, 0);  
  
 Set<Integer> unique = new HashSet<>();  
 Set<Integer> duplicates = [**numbers.stream**](http://numbers.stream/)()  
 .filter(n -> !unique.add(n)) // Returns false if already present, meaning it's a duplicate  
 .collect(Collectors.toSet());  
  
 System.out.println("Duplicate values: " + duplicates);  
 }  
}  
🔹 Output:  
Duplicate values: [2, 3]

💡 Why Use Java Streams?  
✅ Concise & Readable – No need for manual loops  
 ✅ Efficient (O(n)) – Uses Set to track duplicates  
 ✅ Functional Approach – Cleaner & maintainable code

1. How to avoid deadlock in java ?  
2. Difference between synchronized and lock ?  
3. What is a volatile keyword in java ?  
4. What is the purpose of ExecutorService in java ?  
5. Difference between Callable and Runnable ?  
6. What is a thread pool in java ?  
7. What is CyclicBarrier in java ?  
8. What is CountDownLatch class in java ?  
9. What Is the Difference Between a Process and a Thread?  
10. Tell me some multithreading options you know ?

1. What is an interface ?  
   2. What are Default methods in interface ?  
   3. What are Static methods in interface ?.  
   4. What are Sealed interfaces (Java 15+) ?  
   5. What is functional interface in java ?  
   6. Why should I go for a static method in interface ?  
   7. What is marker interface in java ?  
   8. What is an adapter class ?  
   9. If two interfaces contain a method with the same signature but different return types, then how can we implement both interfaces simultaneously?  
   10. What modifiers are applicable to the interface ?  
     
     
   Core Java Questions  
    1. How does a HashMap handle collisions, and what’s the difference between HashMap and ConcurrentHashMap?  
    2. Explain Garbage Collection what’s the deal with G1 GC vs. CMS?  
    3. What’s the difference between fail-fast and fail-safe iterators?  
    4. How would you implement an immutable class with mutable fields?  
    5. What are soft references, weak references, and phantom references in Java, and why do they matter?  
     
    Multithreading and Concurrency  
    1. What’s a race condition, and how would you prevent it?  
    2. How does the Fork/Join framework work, and when would you use it?  
    3. What are CountDownLatch and CyclicBarrier, and how do they differ?  
    4. Implement a producer-consumer pattern using Java’s BlockingQueue.  
    5. How would you debug a deadlock in a multithreaded application?  
     
    Spring Boot Questions  
    1. What’s the role of Spring Boot Actuator, and why is it useful?  
    2. How does Spring Boot’s auto-configuration work under the hood?  
    3. Explain AOP (Aspect-Oriented Programming) with a real-world example.  
    4. How do you secure a REST API with Spring Security and JWT?  
    5. What’s the difference between @Component, @Service, @Repository, and @Controller?  
     
    API and Microservices  
    1. How would you implement rate limiting in a microservices architecture?  
    2. What’s the difference between asynchronous APIs and synchronous APIs?  
    3. Explain the role of Eureka and Zuul in Spring Cloud.  
    4. How do you ensure backward compatibility with API versioning?  
    5. Design a retry mechanism for a failing API call in a distributed system.  
     
   Advanced Topics  
    1. How would you debug a memory leak in a production Java application?  
    2. Explain CAP theorem and its relevance to distributed systems.  
    3. How does event-driven architecture work with Kafka?  
    4. Redis vs. Memcached: Which one do you pick for caching, and why?  
    5. How do you monitor and troubleshoot issues in a microservices architecture?  
     
     
   Key Topics to Master  
   Java Fundamentals: HashMap internals, GC (G1 vs CMS), fail-fast vs fail-safe iterators.  
   Multithreading: Thread synchronization, race conditions, Fork/Join, producer-consumer.  
   Spring Boot & Microservices: Auto-config, JWT Security, Hystrix, Eureka, API versioning.  
   System Design: Rate limiting, caching (Redis vs Memcached), Kafka, scalable APIs.  
   Advanced Topics: CAP theorem, memory leaks, event-driven architecture, troubleshooting microservices

Java interview questions and answers   
-------------------------------------------------  
1. Difference between StringBuffer and StringBuilder?  
  
StringBuffer  
- mutable class  
 - synchronized  
- thread safe  
- used in multi thread environments  
- methods used append(), insert(), reverse()  
  
StringBuilder   
- mutable class  
- unsynchronized  
- not thread safe  
- used in single thread environments  
- methods used append(), insert(), reverse()  
------------------------------------------------------  
2. Difference between Collections and Collection in Java?  
  
Collection:  
- is an interface   
-it represents the group of objects  
-it is parent interface for List,Set and Queue interface  
-Methods used are add(),remove(),size() etc.  
  
Collections   
- is a utility class   
- we can import from java.util package  
-performs operation on Collection objects.  
- provides static methods for operations like sorting, searching, reversing etc  
------------------------------------------------------  
3. What is LinkedHashMap?  
  
-It is the class that implements Map interface.   
-stores elements in key value format  
- duplicate values are allowed  
- null keys are allowed   
-Only one null key is allowed, and if you add another, it overwrites the value of the existing null key.  
-If you try to insert a new entry with a key that already exists, the new value will overwrite the value associated with the existing key.  
- used when insertion order needs to be maintained  
  
  
1. Explain about Event Driven Architecture(EDA) in microservices  
2. Explain about Message Queues  
3. Explain about API Gateways ?  
4. How does API Gateway contribute to scalability, security, and fault tolerance ?  
5. How do you handle inter-service communication in microservices ?  
6. What are different communication patterns in Microservices ?  
7. How do you replace a legacy application with new microservices (or) How to modernize a legacy application to microservices ?  
8. Challenges in Inter-Service Communication in Microservices ?  
9. How do you handle security in microservices ?  
10. What are the strategies for service discovery in a microservices architecture ?

\*\* JAVA 8 Features Interview Questions \*\*  
1. Can I use the synchronized or final keyword on default methods in Java 8 ?  
2. Can a functional interface extend another interface ?  
3. Why were default methods introduced in Java 8?  
4. What happens when a class implements multiple interfaces that have the same default method?  
5. How Default Methods in java 8 Solve the Diamond Problem ?  
6. What Is the Difference Between a Normal and Functional Interface in Java?  
7. What is the Bi-Function interface?  
8. What is MetaSpace? How does it differ from PermGen?  
9. Difference between Collection API vs Stream API?  
10. What does the term “method reference” mean in the context of Java 8?

Core Java Questions  
1. HashMap collisions & ConcurrentHashMap – Talked about hashing, bucket storage, and linked list vs. tree-based structure. Mentioned ConcurrentHashMap uses segment-based locking for better concurrency.  
2. Garbage Collection (G1 GC vs CMS) – Explained G1 GC divides memory into regions and CMS helps reduce stop-the-world pauses.  
3. Fail-fast vs fail-safe iterators – Fail-fast throws ConcurrentModificationException, fail-safe iterates over a copy of the collection.  
4. Immutable class with mutable fields – Used private final variables, no setters, and defensive copies in the constructor.  
5. Soft, Weak, and Phantom references – Soft is GC-aware but survives if memory isn’t full, Weak gets collected immediately if unreferenced, Phantom is for cleanup before GC happens.  
  
Multithreading and Concurrency  
  
1. Race condition & prevention – Used synchronized blocks, locks, and Atomic variables.  
2. Fork/Join framework – Explained how it splits tasks into smaller ones for parallel execution.  
3. CountDownLatch vs CyclicBarrier – Latch is one-time, Barrier is reusable for multiple cycles.  
4. Producer-consumer using BlockingQueue – Used LinkedBlockingQueue to handle inter-thread communication.  
5. Debugging deadlocks – Suggested jstack to analyze blocked threads and timeouts in locks to avoid deadlocks.  
  
Spring Boot & Microservices  
  
1. Spring Boot Actuator – Used for monitoring app health, metrics, and performance logs.  
2. Auto-configuration – Explained @EnableAutoConfiguration and how Spring Boot scans dependencies.  
3. AOP with real-world example – Used @Aspect for logging execution time in services.  
4. Spring Security & JWT – JWT for stateless authentication, configured filters for validation.  
5. @Component vs @Service vs @Repository vs @Controller – Basic role differences in Spring dependency injection.  
  
API & System Design  
  
1. Rate limiting in microservices – Mentioned Redis with token bucket algorithm to throttle requests.  
2. Async vs Sync APIs – Async is message-driven (RabbitMQ, Kafka), Sync is direct HTTP request-response.  
3. Eureka & Zuul in Spring Cloud – Eureka for service discovery, Zuul for API Gateway and routing.  
4. API versioning – Talked about path-based (/v1/users) and header-based versioning.  
5. Retry mechanism for failing API calls – Used Exponential Backoff + Circuit Breaker (Resilience4J)  
  
Did I Answer Everything Correctly?  
I walked out of the interview feeling confident, but let’s be honest no one answers everything perfectly. There were moments where I over-explained, times when I could’ve structured my response better, and a couple of places  
  
  
\*\* Java Spring Interview Questions PART IV \*\*  
1. 1. Difference between Lazy loading and Eager loading in Spring ?  
2. Tell me about Eager Loading and Lazy Loading in terms of Spring Beans  
3. What is the default scope of @component, @Repository or @service ?  
4. When ApplicationContext is initialized, are beans declared as PROTOTYPE loads eagerly or not ? Why ?  
5. Difference between Prototype Scope and Request Scope in spring bean   
6. What is Dependency Injection and how do you achieve it ?  
7. Are spring beans thread safe ?  
8. Tell me some bean lifecycle methods   
9. Difference between BeanFactory, Applicationcontext, webapplicationcontext in spring ?  
10. What are different ways to configure a class as Spring Bean?

\*\* MICROSERVICES interview questions in JAVA Part-V \*\*  
1. Explain about Event Driven Architecture(EDA) in microservices  
2. Explain about Message Queues  
3. Explain about API Gateways ?  
4. How does API Gateway contribute to scalability, security, and fault tolerance ?  
5. How do you handle inter-service communication in microservices ?  
6. What are different communication patterns in Microservices ?  
7. How do you replace a legacy application with new microservices (or) How to modernize a legacy application to microservices ?  
8. Challenges in Inter-Service Communication in Microservices ?  
9. How do you handle security in microservices ?  
10. What are the strategies for service discovery in a microservices architecture ?

Core Java:  
-----------  
How is the volatile keyword used in Java?  
  
How would you develop a custom HashMap? Which data structure would you   
  
use, and how does the get() method work internally?  
  
What is the garbage collector, and when is it invoked?  
  
How do you make a Singleton thread-safe?  
  
Java 8:  
--------  
Write a program using the Stream API to sum up elements of a list.  
  
Write a program to group words in a list based on their first letter using the Stream API.  
  
What's the difference between anonymous inner classes and lambda expressions?  
  
How do Consumer and BiConsumer differ?  
  
What are the key differences between streams and parallel streams?  
  
Spring Boot & REST API:  
-------------------------  
How do you develop a custom exception in Spring Boot?  
  
What is the API lifecycle, and how do you manage it?  
  
How would you develop a global exception handler in Spring Boot?  
  
Spring Data JPA:  
-------------------  
If one method (method1) with @Transactional calls another (method2) also annotated with @Transactional, and method2 fails, does rollback occur for method1?  
  
What is the default Spring propagation level? What are the different propagation levels?  
  
How do you configure a service to connect with multiple databases in Spring Boot?  
  
Microservices:  
---------------  
How do microservices communicate, and how do you handle inter-service failures?  
  
What are the disadvantages of microservices?  
  
Kafka:  
-------  
How do you ensure message delivery in Kafka?  
  
Can you explain consumer groups in Kafka?  
  
Database:  
-----------  
What is a database index, and how does it improve query performance?  
  
General Questions:  
--------------------  
How would you improve the performance of an API?  
  
How do you decide which database to use (e.g., Oracle, MySQL, MongoDB)?  
  
How are application properties configured in your app?  
  
What are the key differences between Kafka and RabbitMQ?

\*\* Java interview questions about FINAL \*\*  
1. What is a FINAL in java ?  
2. Difference between FINAL and FINALIZE in java?  
3. Is every method present in FINAL class is final ?  
4. Is every variable present in FINAL class is final ?  
5. Can a constructor be declared as FINAL ? Why ?  
6. What is the difference between a final variable and a static final variable?  
7. What happens if we declare all of class constructors as private.  
8. Can we create an instance of final class in another class?  
9. Can we mark a block final in Java?  
10. Why Integer class has been defined final in Java?

\*\* MODIFIERS in Java interview questions \*\*  
1. What is the only modifier applicable to local variables ?  
2. Tell me about access modifiers in java  
3. Tell me about non-access modifiers in java  
4. Difference between FINAL and PRIVATE ?  
5. Can I declare a class as private in java ?  
6. What are transient and volatile modifiers? Explain their use cases.  
7. What is the significance of protected access modifier in Java? Can a protected member be accessed outside the package ?  
8. How do access-modifiers affect the ability to override or overload a method ?  
9. Can you use the super keyword with a private method or variable ?  
10. What does the default modifier do in Java interfaces ?  
  
\*\* MICROSERVICES interview questions in JAVA Part-II \*\*  
1. How can you verify that the application registers with the proper instance of the Eureka Server?  
2. Is Spring Boot and Spring Cloud the same ?  
3. Tell me how you implemented Spring Security in a microservice architecture ?  
4. What are components of microservice ?  
5. What is the SAGA Pattern ?  
6. How do microservices communicate with each other ?  
7. Where do you configure API gateway for your spring boot ?  
8. What is Service Discovery in Microservices?  
9. Tell me about service registration in microservices ?  
10. How is Service lookup done ?

\*\* Java == and equals() interview questions \*\*  
  
Let's say :  
  
String a1 = "apple"; String a2 = "apple";  
int i = 5; int j = 5;  
  
Any one can write a java program to find above two strings or two integers are equal or not.  
  
Here the next tricky things for the interview.  
i) Some people get confused using equals() and ==   
ii) What if String a1 = "Apple" and String a2 = "apple"  
iii) For int i = 5 and j = 5, Java allows you to use == for comparison. What happens if I use equals() in this case ?  
iv) Given Integer i = 5 and Integer j = 5, should you use == or .equals() for comparison ?  
v) Why is there a difference between comparing primitive int and Integer objects?