

# Java Interview Questions

Below is the compilation of all the different types of Java, Spring Boot, Database and related questions that were asked in companies like JP Morgan Chase, BlackRock, Morgan Stanley, Deloitte, UBS, Goldman Sachs, BNP Paribas and some other companies interviews that I personally gave.

## Note

Below are some important feedback/improvements suggested to me in my interview so I am highlighting that so that you are not making those mistakes.

1. Maintain proper body language.
2. Whenever you speak, look into the camera and speak. Don't speak looking up or down, etc.
3. Wear formals whenever appearing for an interview: offline or online. Wear formals.
4. Don't rush or speak fast. Explain clearly what you know by taking appropriate pauses.
5. Clearly listen what is being asked.
6. Whenever trying to present or solve something. Always speak out loud i.e. Speak while you are performing tasks so that interviewer is aware and he/she is on same page.

Sr.No	Content	Page No
1	Core Java Concepts	3
2	String Concepts	3
3	Object Oriented Programming	3-5
4	Collection API in Java & Map Interface	5-6
5	Stream API/Collections/DSA Hands-on Questions	6-7
6	Exception Handling	8
7	Multithreading and Concurrency in Java	8-9
8	Design Patterns	10
9	JDBC/Hibernate(JPA)/ Spring Data JPA	10
10	Spring/Spring MVC/Spring Boot/REST API/Asynchronous Communication	10-11
11	Spring Security	12
12	Microservices	12-13
13	Situational Java Questions/System Design/Cloud	13
14	Database Questions	14
15	Frontend/Angular Questions	15

## Core Java Concepts

1. What is the difference between JDK and JRE?
2. Why is Java a platform independent language?
3. Explain Java Memory Model? What is the difference between Stack and Heap memory?
4. Explain Garbage Collection and how it works in Java? What part of memory - Stack or Heap - is cleaned in the garbage collection process? What are various Garbage Collection algorithms in Java and which one is currently being used in recent stable version?
5. What is the difference between final, finally, and finalize?
6. What is the difference between a private and a protected modifier?
7. What is constructor overloading in Java?
8. What is the use of super keyword in Java?
9. What is the difference between static methods, static variables, and static classes in Java?
10. What exactly is System.out.println in Java? Explain each keyword of the statement.
11. What are varargs i.e. int a... in java? Where is it used?

## String Concepts

1. Why are strings immutable in Java and how many objects will be created in String temp = "A" + "B" + "C"; explain your answer in detail.
2. How to find duplicate strings in a list of strings.
3. Difference between String vs String Builder vs String Buffer
4. Difference between Comparable and Comparator.
5. Write a program to find if a String is a palindrome String.
6. Write a program to find a list of substrings of a given String and sort the individual substrings.
7. Find all the words from a dictionary that are anagrams of each other.
8. Run Length Encoded string
9. Finding the most repeating or common word in a string
10. What is String Interning/Intern() method/ String Pool concept?

## Object-Oriented Programming(OOPs)

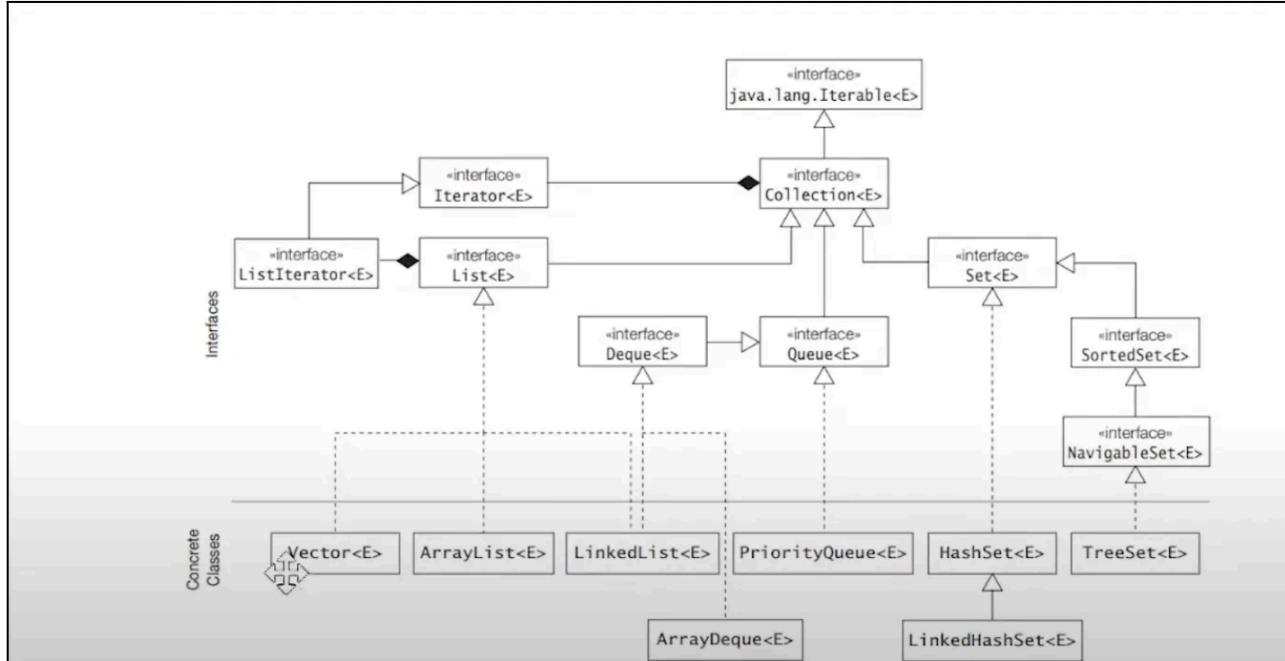
1. What are the Object Oriented Features supported by Java?
2. What is the difference between Method Overloading(Compile Time Polymorphism) and Method Overriding (Dynamic method dispatch or Runtime Polymorphism)?
3. Why does the method overloading doesn't create confusion between the

- same function names?
4. What are the different access specifiers used in Java?
  5. What are Enums? What all can you do with an Enum?
  6. What are marker interfaces?
  7. Shallow Copy vs Deep Copy
  8. What is Pure Component?
  9. What are Functional Interfaces? functional interfaces extended by other interfaces or not?
  10. What are Method References?
  11. What are Lambda Expressions?
  12. What is an Optional class? What is its importance?
  13. What is the use of static method in Interface?
  14. What is the difference between an abstract class and an interface?
  15. What is the difference between composition and inheritance? IS-A vs HAS-A relationship?
  16. What is the purpose of an abstract class?
  17. What are the differences between constructor and method of a class in Java?
  18. What is the diamond problem in Java and how is it solved? Multiple Inheritance
  19. What is the difference between local and instance variables in Java?
  20. What is a Marker interface in Java?
  21. Static method and is it thread safe?
  22. Cohesion vs Loose Coupling vs Tight Coupling?
  23. How can you make a final class or an immutable class? How do you make sure the list present inside the immutable class does not get changed whenever it is accessed from some other class and the new list modifies the reference to the list that is present in immutable class. (Ans: make sure when you return the list from the method in immutable class, you always return the new object of the list and pass the actual list as argument. This will make sure the new reference is created and old one is no longer used)
  24. What are Generics? What does ? means in generics? What is reifiable in Generics? Does data gets retained in Generic during runtime or not?
  25. How do you create an Inner Class? How to create object of inner class from the main() method? What is anonymous class and how do you create it?
  26. Serializable vs Externalizable interface? Serialization and Deserialization how to write the code to serialize an object to Bytestream or JSON etc and then deserialize it to matching object type again.
  27. Using how many stacks you can implement queue?
  28. How do you create a functional interface that concatenates two strings if given as argument and adds if two integers given as arguments.

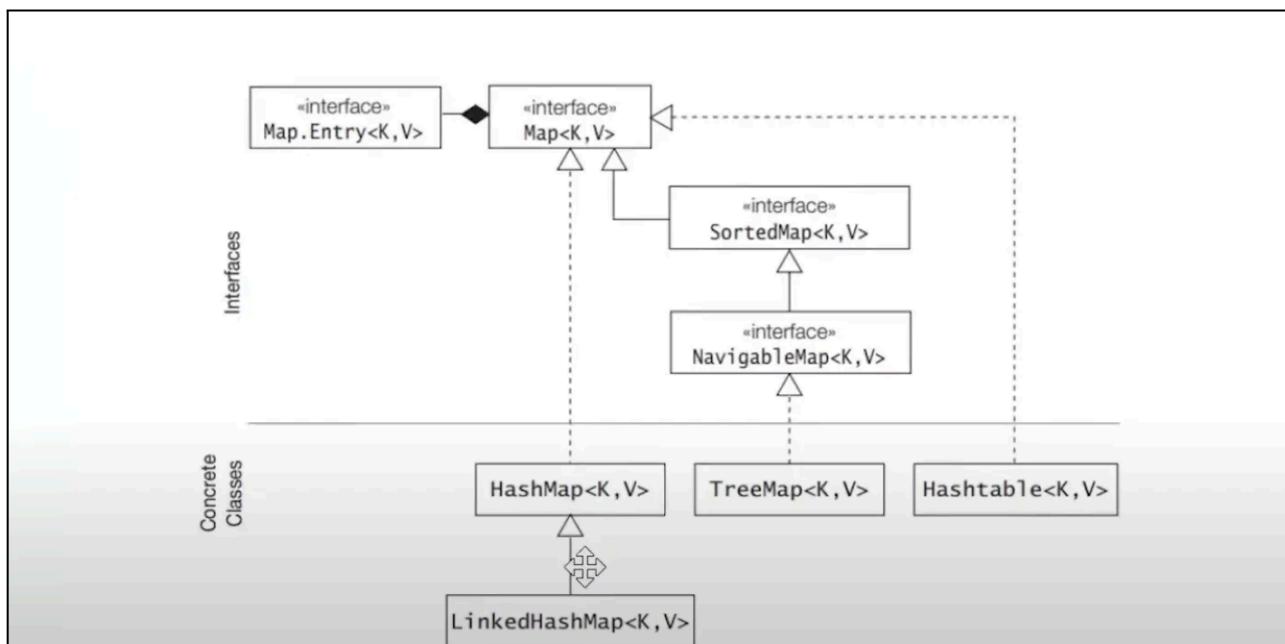
## Ans. Function<T>

29. How to initialize a list with 50 integers
30. Arrays.asList
31. What is type in Java
32. When we pass an object to a function, what exactly gets passed?  
Reference, object itself, reference copy etc

## Collection API in Java & Map Interface



Pic Credits: Riddhi Dutta @Youtube



Pic Credits: Riddhi Dutta @Youtube

1. Explain each interface and class present in the Collection interface. Starting from the Collection interface tell me the timecomplexity of each for insertion/deletion, accessing elements, insertion order maintained or not.
2. Internal working of HashSet? How are null key and values are handled in HashSet?
3. Why do we need to override equals and hashCode methods in Java? Equals - HashCode Contract in Java?
4. Hash Map Internal Working.
  - a. the Data structure it uses,
  - b. hash calculation and index calculation
  - c. capacity of buckets, load factor
  - d. hash collision? How doubly linkedlist is used to resolve hashcollision? what is used when all the buckets of doublylinkedlist is filled?
  - e. time complexities of insert, delete and search operations.
  - f. What is the Role of equals() and hashCode() methods in the Object class?
  - g. What will be the behavior if we override the hashCode() method to always return?
  - h. How are null key and values are handles in HashMap?
5. What is CopyOnWriteArrayList?
6. What is ConcurrentHashMap?
7. HashMap vs HashTable vs ConcurrentHashMap
8. Map vs FlatMap?
9. Synchronized collections vs Concurrent collections
10. Fail-safe vs Fail-fast concept in Java? What are Fail-fast and Fail Safe iterator in the collections framework?
11. Comparator in Java and how do you create one? What is natural ordering? Can it be used for multiple fields sorting?(No)
12. Comparable in Java and how do you create one? Advantages of Comparable over Comparator? What is total-ordering? Which one would you use in real time project for sorting or ordering? Can it be used for multiple fields sorting?(Yes)
13. What is the difference between ArrayList and LinkedList?
14. What is the difference between a HashMap and a TreeMap?
15. What is the difference between a HashSet and a TreeSet?
16. What is the difference between an Iterator and a ListIterator?
17. What is the difference between an ArrayList and a LinkedList?
18. How to sort an ArrayList? Ascending order/Descending Order using Custom Class fields.
19. How to sort a Map? Ascending order/Descending Order using Custom Class fields.
20. How do you pass the Comparable to the PriorityQueue or TreeSet?
21. Can we insert null in treeset?

## Stream API/Collections/DSA Hands-on Questions

**Note:**

1. Make sure you do lots of practice as it will be asked to run the code in an IDE. Be thorough with all the methods and ways you go about solving something especially in Java.
2. Always they will ask you to write the code in Java 8 or later way. So, always ensure you are using Stream API whenever necessary, and using all sorts of Functional Interfaces with the correct use of Lambda Expressions.
  
1. How to perform group by filtering on a list of Employees with employee having same city and gender. How to use comparators. How to sort based on two fields (equals method). How to get distinct employees i.e. no record should exist with same Age and City  
 Ex: Input: Employees list  
 Shivam, 23, Delhi, Male  
 Maaya, 24, Delhi, Female  
 Aradhya, 25, Delhi, Female  
 Kiosaki, 31, Mumbai, Male  
 Arjun, 25, Mumbai, Male  
 Output:  
 Delhi, Male, count=1  
 Delhi, Female, count=2  
 Mumbai, Male, count=2
2. Stream API code to get second highest salary for each dept. Given is  
`Map<String, List<Employee>> deptList; i.e. Map<DeptName, List<Employee>>`  
`d1={E1,E2,E3,E4,E5} d2={E2,E5}`
3. Stream API (multiply all the elements of a list)
4. Stream API (how to do exception handling in stream API)
5. DO MORE CODING ON STREAM API/ GROUP BY / AGGREGATION / REDUCE / STORING THE RESULTS IN A GIVEN COLLECTION
6. Write a code to rotate the array at a given index
7. How will you find a single duplicate number from a large array in minimum time?
8. How will you sort a salary of an Employee?(Show in IDE)
  - a. Once Sort Remove the 4th Highest Salary of an Employee?(Show in IDE)
9. Find the minimum element in the stack. An optimized solution is required.
10. Bucket, Counting sort, Radix Sort for O(n) time complexity for smaller list or arrays i.e. size 100
11. How will you implement a Stack using a Queue
12. How will you implement a Queue using a stack
13. How to sort data that can't fit into a main memory
14. Reverse a singly linked list

15. Check if a linked list is palindrome or not
16. Min-heap
17. Hashtable
18. Find the Top X occurring words in a very big file using Min-Heap and Hashtable.
19. How will you find the common ancestor of two given nodes A and B in a binary Tree?
20. How does the reverse key index help in faster data reading and what is the use case for this? The complexity of the B tree index?
21. How to find whether Binary Tree is BST or not?

## Exception Handling

1. What is an exception? What is an error? How are both different? How do you handle each? Compilation Error vs Runtime Exception.
2. How does an exception propagate throughout the Java code?
3. What is the difference between checked and unchecked exceptions? throw vs throws keyword?
4. What is the use of try-catch block in Java?
5. What is the use of try-with resources?
6. What is a multi-catch statement in Java?
7. What is the difference between throw and throws?
8. What is the use of the finally block? Does it runs everytime? When does the finally block does not run?
9. What's the base class of all exception classes?
10. Write a code in Java to create custom exceptions and use it in a try catch block?

## Multithreading and Concurrency in Java

1. Program vs Process
2. Inter Process Communication
3. Multiprocessing real life example
4. Multiprocessin vs Multithreading
5. What is Multithreading and where have you used the multithreading or written code for the same in your project?
6. Proc
7. What is volatile keyword in terms of thread?
8. Need for synchronization in java?
9. What is thread safety and How will you achieve thread safety in a Java Program?
10. What is a thread and what are the different stages in its lifecycle?
11. How do you create a thread in Java? What are the different ways of creating a thread? Which one is the best way to do it?
12. What is the difference between process and thread?What are the different types of thread priorities available in Java?
13. What is context switching in Java?
14. What is the difference between user threads and Daemon threads?
15. What is synchronization?
16. What is a deadlock?
17. Class level lock vs object level lock in Java?
18. What is the difference between synchronized and volatile in Java?
19. What is the purpose of the sleep() method in Java?
20. What is the use of the wait() and notify() methods?

21. What is the difference between wait() and sleep() in Java?
22. join() vs yield() method ?
23. What is the mechanism for inter-thread communication?
24. Threading: Yield vs Join vs Sleep vs Wait vs Notify
25. What is the difference between notify() and notifyAll() in Java?
26. Java Concurrency, Concurrency utils (Atomic Package), java.util.concurrent package.
  - AtomicInteger, AtomicLong, and AtomicBoolean
  - AtomicReference and Atomic ReferenceArray
  - Compare-and-Swap Operations
27. Concurrency Utilities:
  - Callable and Future, CompletableFuture
  - Scheduled ExecutorService
  - CountDownLatch, CyclicBarrier, Phaser, and Exchanger
28. Concurrent Collections:
  - ConcurrentHashMap
  - ConcurrentLinkedQueue and ConcurrentLinkedDeque
  - CopyOnWriteArrayList
  - BlockingQueue Interface
  - ArrayBlockingQueue
  - LinkedBlockingQueue
  - PriorityBlockingQueue
29. What is Thread Pool? Executor Service? Write a custom thread pool without executor.
30. Executor Framework vs ForkJoin Framework?
31. ThreadLocal Variable?
32. There is a very big text file containing words. How would you read & process it to print the below output.
  - a. Print the top ten ranked distinct words.
  - b. Print the occurrence of each alphabet in this file considering that the file can not fit into the main memory of the computer. [Solution discussed here](#)
33. Runnable vs Callable vs Future Interface
34. @Scheduled annotation: quartz process -> cron job
35. Common Concurrency Issues and Solutions
  - Deadlocks
  - Starvation
  - Livelocks
  - Race Conditions
  - Strategies for Avoiding Concurrency Issues
36. Locks and Semaphores
  - ReentrantLock
  - ReadWriteLock
  - StampedLock
  - Semaphores
  - Lock and Condition Interface

## Design Patterns

1. What are Design Patterns? Different Types of Design Patterns?
2. Write a code for Singleton Class? Write all the variations of a Singleton class from lazy loading to thread safe singleton to Bill Pugh Singleton.
3. How do you make a singleton class thread safe?
4. What is Builder Design Pattern?
5. What is a Decorator Design Pattern?
6. Concurrency Design Patterns

## JDBC/Hibernate(JPA)/ Spring Data JPA

1. What is JDBC? How to connect a database to a springboot application using JDBC?
2. What is an ORM?
3. Difference between JPA and Hibernate
4. JDBC vs Spring Data JPA. Performance comparison. Which one is better?
5. What are the main classes of Hibernate and some important classes of JPA.
6. How to configure and use Spring Data JPA with a springboot application. Remember each and every config property name as it will be asked and its relevance, why its required. Like Dialect, DDL-Auto, etc
7. JPA inheritance strategies
8. Entity Relationship with examples - OneToOne, OneToMany, ManyToMany). how to define relationship like many to one one to many in entity class
9. Transaction Management, Handling concurrent updates in the database, etc. Spring Transaction and its associate annotations. How do you handle rollback in Spring?
10. Difference between lazy and eager loading in Hibernate. How to prevent LazyLoadException?
11. hibernate: sesion.get vs session.load
12. What is hibernate session factory

## Spring/Spring MVC/Spring Boot/REST API/Asynchronous Communication

1. What is Java Enterprise Edition (Java EE)?
2. What is the difference between a Servlet and a JSP?
3. What is the purpose of the Java Persistence API (JPA)?
4. What is the difference between stateful and stateless session beans?
5. Spring vs Spring Boot. What is the need of Spring Boot?
6. Spring IOC Container. Spring Application Context.
7. SpringBootApplication annotation and its underlying annotations and their uses
8. Scope of Spring Beans: Singleton, Prototype, Session, WebSocket, Request. Where do we write Scope=Prototype?
9. Lifecycle methods of Spring Bean.
10. Qualifier vs Primary annotation in Spring. Are they both necessary or any one of them would help? Why they are used? How do you handle the situation when you have multiple beans with same names? How does Spring comes to know which one to be injected?

11. How to deploy Spring Boot Application WAR on Tomcat Server
12. Dependency Injection and its benefits. What are types of Dependency Injection in Spring? How do you resolve the Cyclic Dependency issue i.e. Dependency A is injected in Dependency B and Dependency B is injected in Dependency A.
13. How do you configure and use externalized configuration properties in Springboot app. Provide example with application.properties or .yaml file
14. Spring Profile. How do you setup multiple profiles like qa uat, dev, prod in springboot. Provide example for the same.
15. Explain the concept of autoconfiguration in springboot
16. Spring Annotations and Name any 10 annotations and explain them?
  - a. Component
  - b. Service
  - c. SpringBootApplication
  - d. EnableAutoConfiguration
  - e. ComponentScan
  - f. Configuration
  - g. Controller
  - h. RestController
  - i. Repository
  - j. Entity
  - k. RequestMapping
  - l. RequestBody
  - m. ResponseBody
  - n. RequestParam
  - o. PathVariable
17. REST vs SOAP
18. How do you create a REST API? Which protocol it uses? What are request and response formats used?
19. How do you create a SOAP API? Which protocol it uses? What are request and response formats used?
20. What is an Actuator in Spring Boot and why it is used? How to create custom endpoint in SpringBoot Actuator? What are some important endpoints?
21. Spring AOP. What is the need for it? What problem it solves? Aspect Oriented Programming (AOP) what are Cross Cutting Concerns?
22. Spring Batch. What is the need for it? What problem it solves?
23. What is Junit and how do you write test coverage. What is TDD?
  - a. What is Unit Testing
  - b. What is Integration Testing
  - c. What is Regression Testing
24. How do you write unit tests for springboot application. Provide an example to test the service layer.
25. How do you handle exceptions in Springboot application?
26. What is Swagger and how do we use Swagger API for API Documentation?
27. Security Issues in older Spring version before your current version of spring that you are working on. What are the improvements in the newer version of

## Spring boot 3?

### Spring Security

1. What is Spring Security?
2. Simple AUTH vs JWT(JSON Web Token) vs OAuth. Pros and cons of using each.
3. Security filters
4. Pre-authorization
5. Post-authorization
- 6.

### Microservices

1. What are Microservices and when should someone think about Microservices in terms of migration? What problems does it solves? What are the disadvantages of Microservices?
2. CAP theorem in case of Microservices
3. How do microservices communicate with each other?
4. Point to Point Communication vs Publish Subscribe Communication
5. Apache Kafka.
  - How it offers High Throughput
  - What are Partitions
  - What are Consumer
  - How many partitions will be consumed by different individual consumers in a consumer group?
  - How many partitions will be consumed by different consumer groups?
  - How can you make a Kafka Topic as a Message Queue?
6. Apache Kafka vs Messaging Queue(Rabbit MQ, IBM MQ, Active MQ, etc). In depth understanding of what to use when to use?
7. Rest Template and its uses? What is now used as Rest Template is deprecated? Accessing RESTful services on the client in Java- Rest Template.
8. How to consume RESTful webservice in a springboot application using RESTTemplate or WebClient
9. What is Reactive Programming? What were other options in Java existing before reacting programming that was introduced recently. Retry() method.
10. Explain how to implement asynchronous processing in Springboot application. Provide an example using @Async annotation
11. How can you secure your spring boot application. Demonstrate the use of Spring Security for basic authentication and authorization
12. Circuit Breaker Pattern used in Microservices. Why its used?
13. What is ELK i.e. Elastic Search - Logstash - Kibana stack? Why its used?  
What is Splunk?
14. What is Zipkin?
15. What is Prometheus and Grafana and how are they utilized in Microservices?
16. How to do Distributed tracing - zipkin, telemetry ?
17. How to debug the cascading issue?

**18. How to rollback efficiently in case of microservices, circuitbreaker design pattern - Resilience4j, Histerix, Fafka and AWS Kinesis?**

**Situational Java Questions/System Design/Cloud**

1. How to solve the poor memory performance of a live production program?
2. Would adding multi-threading to the sorting algorithm improve its performance?
3. What are the ways to increase the throughput of a multi-threaded java program? (concurrency enhancements - use `java.util.concurrent` package, connection pooling, other resource pooling, caching, GC tuning, etc.)
4. In a client-server architecture, there are multiple requests from multiple clients to the server. The server should maintain the response times of all the requests in the previous hour. What data structure and algorithm will be used for this? Also, the average response time needs to be maintained and has to be retrieved in  $O(1)$ .
5. How will you design a connection pool in Java?
6. How will you design a distributed algorithm for Prime Number generation in Java?
7. [What is a deadlock situation? How will you handle deadlock in development and production environment?](#)
8. Producer Consumer problem with a thread-safe blocking queue. i.e.  
How do you pass a job from Thread 1 to Thread 2 to Thread 3?  
Hint: Some kind of blocking queue is used to safely pass data from one thread to another. It is a typical Producer Consumer problem
9. How will you design a two-way elevator software for a building?
10. How would one design a multi-format converter that supports reading data from multiple data sources(web service, local disk, etc.)? The data from the sources can be in multiple formats. The reader for each format may be different and how does one serialize this abstract data to multiple formats like image, XML, etc? New readers, writers, and data sources can be added later during implementation.
11. How to find server crash reasons?
12. How do you handle the scenario in your java application if the message that was read from the IBM MQ was read and while processing some error occurs. How can we handle this issue such that no data is lost?
13. How to improve the performance of an web ecommerce application that gets multiple API requests(1000+). Apart from lazy loading and Cache what else we should use?
14. How do you design an application with data (metadata, media, etc) and generates lot of api request hit? What will be different databases, api designing, scalability options, etc?
15. Autoscaling in AWS

**Database Questions**

1. ACID Properties in SQL
2. How to find the number of tables and their columns in SQL DB?
3. Different types of Locks?
4. Brush up on all the statements that you write in a query. At what position a group by is written or at what position the having or order by or where clause is added.
5. What is the diff between union and unionall operation in sql
6. Stored Procedures vs Functions
7. Triggers and how it works internally. Update statement when run for 50 rows how many times the trigger is called to store the data from main to log table.
8. What are Joins in SQL and how many different types of Join are there? Explain each join and number of records from each table that will be returned as an output. Also explain in case of left outer and right outer what column values be null?
9. Inner join vs Left outer join vs Right outer join
10. Database Normalization and its various stages
11. Types of Database Indexes? How do you improve the performance of your queries?
12. How to handle parallel updates in a database, where two or more requests are coming in parallel?
13. Many to Many, Many to One, One to Many, and One to one relationships with examples
14. How will you find the nth highest salary of an employee from the Salary table?
15. Database Design of a given application. Database designing for various tables and foreign key relationships, indexing etc.
16. What is Sequence in SQL?
17. How to debug prod issue of Database query or objects running slow.

Ans. Ask the DBA team to provide the detailed AWR report of each query resulting into slowness

### **Frontend/Angular Questions**

1. Display none vs display hidden difference in Javascript?
2. zone.js life cycle hook ngZone
3. AOT compilation in Angular
4. props performance issues downside of using angular over react state management in angular
5. How to load the Module, Components using LazyLoading on a click of a button or after some event occurs?
6. How do you redirect a user to admin page and a normal page based on the login for a new user only using Angular and no backend should be used?
7. How to transfer data between components in Angular?
8. Directives or Angular concepts you worked on in your current project?

9. Different ways to create object in Typescript? Different ways of creating object in Angular?
10. What versions of Angular you have worked on?
11. What are the features of Angular 8 or 9 or 16 etc version?
12. What is the difference between JavaScript vs Typescript?
13. Building blocks of Angular
14. Promise vs Observable
15. What are the different types of Directives in Angular?
16. Life cycle methods in Angular. Purpose of using ngOnDestroy method
17. What is selector
18. Write Angular Code: Radio Button A and Radio Button B. When user selects A then Div with green color needs to be displayed. When user selects B then Div with red color needs to be displayed.
19. Scenario: There are 10 rest api calls. We need to make from Angular and then using the response of all the api call we need to make another api call. How can this be achieved in Angular?
20. Scenario: We are getting a response of List<Employee{Age, EmpId, Salary, Address, Name}>. Write a logic to transform the response to only List<Employee{id, name}>
21. If an Angular application is having a performance issue then how do you identify and rectify it?
22. How do you maintain the Authentication while routing the data between different components? Dependency Injection in Angular. How do you inject services to a component? Different ways to inject services to an Angular component?
23. How do you share data among Parent-to-Parent, Parent-to-Child, Child-to-Parent
24. Practice coding of Angular for services, input, output, observable, promises
25. @Input and @output decorator in angular