**Consolidated Lowe’s Interview Experiences**  
  
**Madhavi Interview 1**:

Explain microservices architecture

What are the challenges faced by you during development of a microservices based architecture

Advantages of microservices architecture

Explain in your recent project, how you develop the solution from scratch

Explain architecture of your current project

Java 8 features

Madhavi Interview 2:

Write a java program to count the repeated characters in a string

Write java program to implement stack using array

Data structure binary tree implementation

Have you worked on Event based programming? Explain

Reach programming and spring flux

Kafka

API integration testing

What all components involved in developing API

BDD framework

**Interview 1:**

1. Oops concepts with examples (polymorphism: static & dynamic, Inheritance, Abstraction)  
2. Write singleton design pattern. Can we extend a singleton class? If yes what we can do in singleton class to avoid inheritance?

3. Write sql and hql query foe insert, update, groupby and many to one mapping.

4. Write Restful webservices flow from controller to Dao with all annotations.

5. Can we make classes static? Constructor can be static? Static inner classes.

6. Microservice overview

7. Difference between rest and soap.

8. Spring Boot configuration and where to keep properties to override the defaults.

9. Microservice architecture, how to consume a microservice, security in services.

**Interview 2:**

Java Microservices with NoSQL

**First Round**

1. Write a REST API (on paper) to count the number of hits to that API. How would you handle multiple request coming together?

2. What is problem with Singleton objects?

3. Hashtable vs concurrent HashMap.

4. Where would you use concurrentHashMap and HashMap?

5. NoSQL vs RDBMS and where would you use which one.

6. Monolithic application vs microservices architecture. How would you break a monolithic app into microservices?

7. What are the advantages of microservices over monolith and vice-versa?

**Second Round**

1. Monolithic vs microservices.

2. Microservices communications

3. Queues vs direct calling the microservice.

4. ***Functional Programming*** : Functional programming is a paradigm that allows *programming using expressions i.e. declaring functions, passing functions as arguments and using functions as statements*

5. **Lambda expressions**: *A lambda expression is a short block of code which takes in parameters and returns a value.*

*Lambda expressions are similar to methods, but they do not need a name and they can be implemented right in the body of a method*.

6. How would you merge two separate integer lists into single list and then iterate and print some message for odd and even int. using java 8.

**Interview 3**

1. Internal working of collections data structures. TreeMap and concurrentHashMap was asked from me and their complexity.

2. If only elements in decreasing order, like 1000, 999, 998....3, 2, 1 are stored in a TreeMap, then all the elements will be stored in left node of its previous node and right node will be null for all. In this case, it is similar to a LinkedList. What difference it makes to use TreeMap over linked list in this case.

3. Write pseudo code for finding the depth of a tree.

4. Lifecycle of a spring bean and questions around when the lifecycle methods are called.

5. Difference between RDBMS and NoSQL DB.

6. How is storing data in files as BLOB different from storing data as document file in mongo?

7. Components of spring boot that one has worked upon.

**Interview 4:**

The interview went good. The interviewer asking more of conceptual topics.

viz., Data Structure (tree, sorting), Config/Discovery Service, MQ data transfer concepts, Circuit breaker, spring security etc.

**Interview 5:**

Java Microservices with NoSQL Db - Req. 514-517– Madan Kumar

1) String concepts in depth

2) Develop a microservices

3) How to have communication between two micro services

4) Write code to create Rest API

5) Encapsulation concepts

6) How you a service is micro service

7) What is @RestController?

8) What is difference between @RestController and @Controller?

9) How you design the simple micro service application

10) Explain Http client in web services

11) Explain web services

12) Java 1.8 features

13) Pseudo code for streams concept

14) What is a thread?

15) String buffer concept?

16) Explain project architecture?

17) ***Why string is immutable***: The string is Immutable in Java because String objects are cached in the String pool.

Since cached String literals are shared between multiple clients there is always a risk, where one client's action would affect all other clients.

18) What is purpose of string design as immutable? : *The String is immutable in Java because of the security, synchronization and concurrency, caching, and class loading. The reason of making string final is to destroy the immutability and to not allow others to extend it. The String objects are cached in the String pool, and it makes the String immutable*

19) What is string constant pool?

20) How class make as immutable?

**Interview 6:**

*How do you display the duplicate words in a file when below are the resources you have: File size is 5GB, it can be above in future. RAM size is 1GB.****Solution:***<https://www.geeksforgeeks.org/sorting-larger-file-with-smaller-ram>  
Suppose we have to sort a 1GB file of random integers and the available ram size is 200 Mb, how will it be done?

The easiest way to do this is to use external sorting.

We divide our source file into temporary files of size equal to the size of the RAM and first sort these files.

Assume 1GB = 1024MB, so we follow following steps.

Divide the source file into 5 small temporary files each of size 200MB (i.e., equal to the size of ram).

Sort these temporary files one by one using the ram individually (Any sorting algorithm: quick sort, merge sort).  
-----

How do you read the data in it, if you are able to overcome the OutOfMemoryException kind of issues?

What are the solutions that you can think of?

How do you handle the Scenario #1 in a clustered environment? Do you use any 3rd-party software's/tools to process? If yes, what are they? If No, then how you are going to handle the 'Duplication' issue?

What is Docker? What is it used for? How it is different from traditional software's? What are Pros and Cons?

How do you check if the image is not responding/running? What logs you will check? What are they called?

What types of logs you know? Name the types of logs?

One image is running fine in 'Local/Dev environment' but not in 'Production Environment'. How do you handle the situation? Are there any procedures you will follow? If yes, what are they and how you are going to use?

Scenario: Client has a Monolithic application, which was working fine until few months back. Now performance got degraded.

Requirements:

Client wants you to solve this problem.

Client wants you to change the architecture i.e., Upgrade the system/application.

Client wants you to provide optimal solution, where less effort/money involves.

Client wants you to give your first thought/solution which comes to your mind and for which reason.

How do you convert Monolithic application to 'Micro Services'? What are the procedures/patterns you will follow while converting/changing?

How do you handle/manage when there are many types of logs? Do you write the software or use the existing 3rd-party?

What do you know about the Clustered environment and how do you create/add a node (i.e., Adding one more to the existing)?

What is a load balancer in AWS? What are the Pros & Cons? How do you handle it?

How 2 or more microservices can communicate each other? What are the ways to do and how many?

Difference between SOA and Microservices? Is the 'Microservices' architecture is the ultimate solution for all?

Common Question: For every solution, how it is going to fail and Where/When/How/What?

**Interview 7:**

What is HashMap? How hashing works internally?

Why hashCode() returns int instead of long? : *HashCode based data structures (HashSet, HashMap) use an array to store the bins, and arrays are limited to int indices (MAX\_VALUE). You will gain nothing by a long hashCode() if you must map it to an int array index.*

Which data structure you suggest for ordered/sorted? Can both be achieved?

How to convert ordered data to sorted? How many ways can it be done?

What is the ordering of ArrayList/HashMap?

What are the differences between Array and ArrayList? Which one you would prefer on what type of situations?

Can the hashCode is calculated for numbers (i.e., int values) also? If so why and how?

What's the use of default methods?

Explain Spring Bean life-cycle? What are the stages, methods and processes it will follow to keep track of the statuses?

What is Spring Boot? What's the use over Traditional Spring application?

What are Spring Boot features? Why it is preferred over traditional spring application?

Have you worked on CI/CD process? Which tools/software's you have used?

Have you hands on experience in Docker/Kubernetes?

Apart from the deployment process in Docker/Kubernetes, have you done anything else specifically?

How you have used Jenkins in the process of CI/CD? What are the configurations you did?

Which cloud platform you know?

Have you worked on any NoSQL Databases?

Why go for NoSQL instead of RDBMS? What are the pros and cons? Which one you choose?

**Interview 8:**

1. Have you worked on any Functional Programming language? Is yes which language it is? If not, can you write Functional Programming in Java?  
 2. What is Flux, Mono and Flex Iterator?  
 3. What is Marker interface? Example  
 4. How do you use the serialization process? How do you identify whether the transferred object is modified or not? Example  
  
 5. Which is the best suitable data structure, for processing/fulfilling the below requirements, that you use for below custom classes:  
 Employee(int id, String name, double salary, int deptId) & Department(int id, String deptName)  
 Details:

You cannot use any 3rd party framework to retrieve/build the data or to fulfill the functionality.

Solution should be performance effective. Means time/space complexity should be at best.

Data (i.e., Employee and Department objects) is already present in In-Memory. Imagine your own data structure used to hold the data in In-Memory.

Don't traverse through the data using looping conditions.

Define 4 methods to perform some operations: getAllEmpsByDeptName(), getTotalEmpsSalaryByDeptName(), getSecondHighestSalary() & getAllEmpsByDeptNameAndSalary()

Above 4 methods in class should be generic.

Which is the best searching algorithm you recommend for retrieval of data with the above requirements? Write code

6. What is Functional Interface? What are the methods allowed in there? What is default method?  
 7. How do you convert array/list into streams? Write code  
 8. How do you process the array/list parallel in java 8? Write code

**Interview 9:**

1. What is IOC?

2. Explain spring bean life cycle?

3. Write a program to sort a map by value using stream

4. How to implement spring security?

5. What is functional interface?

6. What is static block?

7. What is default method in interface?

8. Describe the recent project you have worked on

9. Do you have any retail domain experience? If yes, please elaborate

10. Describe how HashMap works internally.

11. Consider there are 4 microservices and communication among them are like A->B->C->D. If C fails, how to overcome the situation?

**Interview 10:**

Previous projects discussion

Explain Hashing.

What is Hash map upgradation in Java 8 & explain?

Program:

Step 1: Palindrome, Fizz-Buzz

Step 2: Define Complexity of the two programs

Step 3: Write code of these two programs in optimize way by reducing Time & Space Complexity.

What is Time & space Complexity?

Difference between ArrayList & LinkedList.

Comparator in Java 8

Java 8 features.

Example in Streams with filter.

Client handling experience

Agile Concepts.

Experience as Scrum master role from previous

**Interview 11:**

What is REST full form?

What is Representation?

What is State?

Is REST Stateful or Stateless? If Stateless, why it is called as 'State' Transfer?

For what, REST is used?

If an endpoint is not defined, means, is it not a web service?

What is the first step in OAuth 2.0 flow?

What are the components called, in OAuth 2.0 flow?

How does an 'Authorization Server', in OAuth 2.0 flow, will understand that what type of authentication mechanism is been used to validate the user?

What are the tokens that you will generate in OAuth 2.0 flow?

If spring dependency Injection concept is 'NOT THERE/PRESENT', how do you create your custom IoC/Dependency Injection concept?

Scenario:

A Book with primary key 'b1' already inserted in horizontally scaled database (i.e., in D1). All are sharded. There are 4 databases used, say D1, D2, D3 & D4, for saving the Book.

Here, D1 has the Book record and D2, D3 & D4 doesn't have the record. Now when I try to insert the Book with the primary key 'b1' (i.e., Duplicate Book Record) again, it should throw an error saying 'Duplicate Record'.

How do you handle this scenario? What are the steps that involved in this process?  
  
<https://docs.mongodb.com/manual/core/sharding-shard-key/>

**Shard Keys**

The shard key is either a single indexed [field](https://docs.mongodb.com/manual/reference/glossary/#std-term-field) or multiple fields covered by a [compound index](https://docs.mongodb.com/manual/reference/glossary/#std-term-compound-index) that determines the distribution of the collection's [documents](https://docs.mongodb.com/manual/reference/glossary/#std-term-document) among the cluster's [shards](https://docs.mongodb.com/manual/reference/glossary/#std-term-shard).

MongoDB divides the span of shard key values (or hashed shard key values) into non-overlapping ranges of shard key values (or hashed shard key values). Each range is associated with a [chunk](https://docs.mongodb.com/manual/reference/glossary/#std-term-chunk), and MongoDB attempts to distribute chunks evenly among the shards in the cluster.

**Interview 12 – Full Stack**

1. Brief introduction about your Roles and Responsibilities

2. Scenario: Print the name of employee who is a manager and an employee in an organization from the Employee table

3. Scenario: Create a service which displays Employee based on the Department Name passed using SpringBoot Rest + JPA

4. Explain the bean scopes available in spring

5. Create a Singleton class. How is it different from Spring Singleton?

6. Scenario: An array data structure has set of string variables, print the number of occurrences of the variables without using collection API’s like Map

**Interview 13 – Full Stack**

1. Brief introduction about your Roles and Responsibilities

2. Scenario: How do you build Angular app and SpringBoot app together in container like Docker

3. How do you access alone hosted Angular component with static json data in an HTTP request

4. Scenario: How do you migrate Spring Applications to spring boot and what all need to be considered to do?

5. How Spring boot is different from its earlier version and what are its benefits

6. Have you worked on Message Broker or streaming application? Could you explain Kafka or RabbitMQ configuration with SpringBoot

7. Could you write or tell about complex queries. Can you write query where it prints second highest salary from an Employee table?

8. Could you explain annotations required to build a service using SpringBoot end to end?

9. Scenario: Explain how authentication works in spring rest services

10. Have you worked on JdbcTemplate using spring over JPA?

11. Could you tell me about the different type’s error codes available and their usage in services?

12. How different services can communicate each other securely

**Interview 14**

Java, Spring Boot Microservices and database questions

Write java code for the programme to check whether the character of the string is unique or not?  
      Input String = INDIA; it should return false  
      Input String = CAR; it should return true

What is the difference between String and character array?

The scenario was to design the database for the twitter project. The database design requires to store and retrieve messages of the user. We can use any database however the data stores in the database should be unique.

Retrieve all the employee details from database using Spring Boot Micro services.

Write the step by step code to retrieve the data. Based on the above scenario, what’s the difference between @Component, @Repository & @Service annotations in spring?

**Interview 15:**

Explain java memory model - heap, stack memory, garbage collectors

How do you secure the rest service?

Explain spring bean lifecycle

What are the annotations and their usage in spring boot application?

How do you connect to the database from service?

Procedures/patterns to convert monolithic to microservice

Microservice architecture and communication between services

Design patterns used in microservices

Write pseudo code to design a bookshelf.

Bookshelf can have no of rows

Each row can contain no of books

Interview 16:

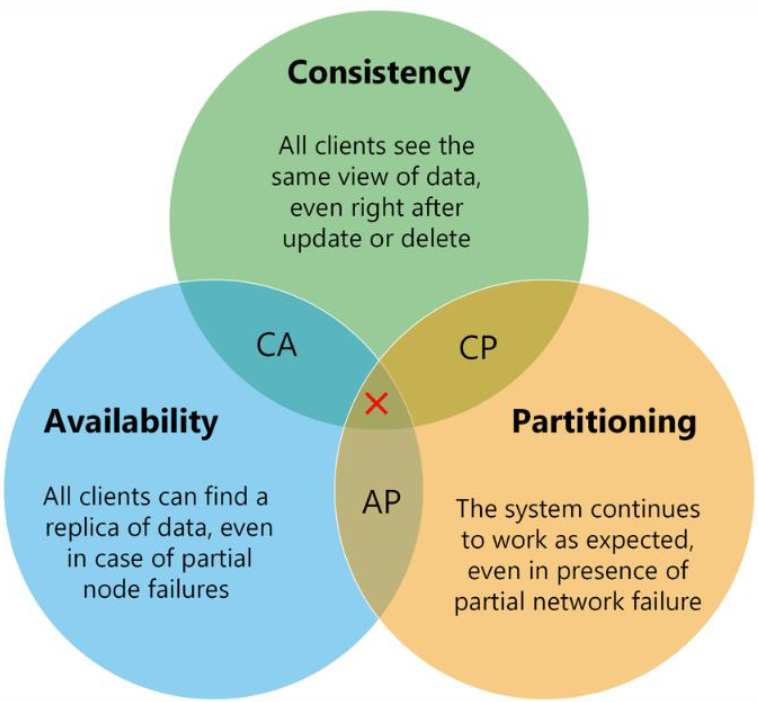
Design of microservices migration for a retail monolithic application -use case

Handling both synchronous and asynchronous communication

**CAP theorem  
Consistency:** Consistency means that if the system get any read request, the data should return last updated value from database under all circumstances. If the data cannot be retrieved, an error should be throw and if data is not up-to-date, then it should never be returned. So, when consistent not provide, the system must block the request until all replicas update.

**Availability:** The ability of a distributed system to respond to requests at any time. If distributed system can respond all request any time, we can say that the system has high availability. Even if one node in any cluster is down, the system should be able to survive with other nodes. Also high available systems can be fault-tolerance in order to accommodate all requests. Availability in a distributed system ensures that the system remains operational 100% of the time.

**Partition Tolerance:** Partition Tolerance is actually network partitioning. That means, parts of your system are located in different networks. Partition Tolerance is the ability of the system to continue its life in case of any communication problem that may occur between the nodes. Its basically guarantees the system continues to operate even if one data node is down.

  
  
<https://medium.com/design-microservices-architecture-with-patterns/how-to-choose-a-database-for-microservices-cap-theorem-d1585bf40ecd#:~:text=Before%20we%20Choose%20a%20Database,achieved%20at%20the%20same%20time>.

Mongo document design for an employee table with department for single retrieval

NoSQL

REST API design

**Interview 17:**

The interview as such was good and hope I catered to all what they asked me to. Below is what I felt based on my discussion,

Though they are not very particular about the domain exp. in Retail/FMCG but if you can give them someone with that, he or she will have a better chance. I had done only 1 big project on travel domain which came close to a retail domain.

From a technical point below is what I went through,

Micro-services: I understand they are also in the process of modernizing their stack and this is the primary requirement which I answered using the Spring Boot framework.

Core JAVA/J2EE,

Spring frameworks as mentioned above,

ReST services and architecture, (I answered this using the Spring Rest template in MVC or Boot applications)

HTML-5 (experienced in method usage like PATCH on top of PUT, DELETE, GET, POST etc.),

Domain driven data segregation/domainization. (Micro-services with data binding was what I mentioned though in the architecture I drew I showed the database creating the domain components on demand from the mainframe with the exception of reference and search indexing data. I am not sure how well he liked the idea though he agreed to the concept)

Collections (I felt this because he asked me about them especially maps, though not in detail. which is very common but thought I should mention),

Java threading concepts (Multithreading, threadgroups, taskexecuters etc.). He was specifically interested in delayed thread execution and termination which can be done in many ways in JAVA. I chose the wait(), clear and terminate though I am not sure if he was completely satisfied with the answer.

Spring profiling (Not a very common thing which is asked in an interview but he asked me that and which I couldn't answer satisfactorily because I have not specifically used the profiling directly out of the box in my projects previously but have catered to such requirements using other tools).

Google cloud experience is expected (Though I could answer most of the things he asked based on my experience with AWS and Azure, I did tell them that I have not worked on a live project in Google cloud).

Containerization technology and High availability which I answered based on my experience with Docker and Kubernetes.

Service Orchestration especially inter-service over service-mesh. I had used NGNIX which internally uses Istio (I forgot this name during the interview though) capabilities and my answer to that was based on my this experience.

3. He did ask me a couple of times on when I joined IBM and I told him it’s been just a week plus. I am not sure how relevant this is but since he asked me a few times I thought I should mention it.

**Interview 18:**

He is expecting good at Docker, example how you deploy code to Docker.

He is expecting some good knowledge at Linux scripting.

Core java he asks a scenario like some strings he will give, and ask some kind logic to build, example he is asking to compare the string inputs and see if two strings are common and how many common letters matching in that. These sort of code you have build and convince him.

He asks to write code in microservices by giving some example. Like employee, department and will explain the how you build basic crud operations end to end.

**Interview 19:**

1. Current project structure

2. JDK 8 Streams. (Question was more focused writing on syntax in java 8 only instead of concept)

3. Idempotent concept of HTTP method

4. Joins

Interviewer asked candidate to open IDE and gave a JSON sample having a list with employee id, name, age, etc.

Then asked to write Java 8 one line code to filter out list with age>21

Hope you have some user friendly IDE installed on laptop or else you may find it difficult to do on notepad since it will not give autosuggestions.

**Interview 20:**

New Features in java 8

Collection Streams

Lazy loading in Hibernate

Http methods (Get, Put, Post etc.) which one is used for what

Restful API using Spring Boot

Spring Boot package structure

How to convert standalone spring to Restful what all changes would be required

Why to use microservices

**Interview 21:**

- Scenario Given and asked for Design Approach (including API calls, Data required) [Sterling Based]

{Once user login to the front end app and wants to see only his orders, 6 months old orders, in this case what could be end to end full flow, what data required and what API's get triggered}

- Difference between operation and transaction [General Case]

- About Java Collections [Hash Map, Hash Set, Array List]

- Ask for code base [need to write the logic (Scenario given)] - How to store bunch of Orders (e.g.: 100 orders) and access them (iteration). If it store in map how to iterate them.

- Few Questions from Database.

**Interview 22:**

Interviewer asked me 3 coding questions in eclipse.

Display distinct elements in any array. - Done n run it to show output

2nd highest number in an array - Done n run it to show output

Control object creation in an array + you should know how many object created for that object

I was coding it by creating a singleton & applying approach when a singleton will not be singleton n counting its objects.

**Interview: 23**

1. Brief me about your experience and skills.

2. Ask 2 write 2 programs in Java 8 and explain the same

    2.1 .Write Custom Functional Interface, Use the same, execute the same also explain

    2.2. Write a program to find the 4 highest number in the given array.

My interview went for only 10-15 mins, it was very short discussion.

**Interview 24:**

1. Understanding about current project and counter questions will be there on project.

2. How to make any class as complete Immutable.

3. Parent class contains run method and handle exception through RuntimeException and Child class override the run method and handle exception through IOException. Is it valid?

4. How to customize health check of spring boot application.

6. What is the command for finding port number for container in Docker?

7. What is the command for finding all the running container in Docker?

**Interview 25:**

1) Self Intro

2) Architecture of latest projects worked?

Core Java:

Java 8 features

Program to sort list, reverse list, map iteration with java 8 syntax.

Design patterns worked?

How we can break singleton?

How we can ensure singleton while serialization?

Faster searching Data structure?

Faster manipulation Data Structure?

Predicate use with sample?

Find the first non-repeating character in a string?

Final in inner class?

Spring boot:

@Qualifier annotation?

How to handle too many requests to a Rest API?

Secure rest API?

Sample rest API flow with ci/cd pipeline?

Diff between put patch?

Handle timeout in API?

Microservice:

How microservice communicate each other?

Fault tolerance in microservice?

Circuit breaker in microservice?

Log management in microservice?

Deployment in microservice?

Disadvantage of microservice?

Migration monolith to ms?

Resiliency in spring boot?

Docker:

Docker hub?

Process of Docker deployment?

Kubernetes:

Pod auto scaling in Kubernetes?

Log check in k8s?

nosql:

Disadvantages of nosql?

When to choose what dbms vs nosql?

Advantages of nosql?

Kafka:

Why Kafka not rabbit mq?

**Interview 26:**

1. Brief about yourself.

2. What is the benefit of using AWS API Gateway?

3. In which case we have to use AWS Lambda Function.

4. What is Circuit Breaker?

5. Have you worked on NoSql.

6. Have you worked on any data structure?

7. As per my answer He asked about LinkedList.

8. How to find the middle element of the LinkedList.

9. As per my answer He has given scenario, suppose there are odd or even no of elements in LinkedList. Find the middle element

**Interview 27:**

Asked to write java8 code for given problem

Question on Design patterns  
Question on Service Discovery  
Question on Locks/Synchronizations  
Design sequential validation framework (if one validation fails then it should not proceed further)  
Question on Compleatable Future  
Question on Circuit Breaker

Few questions on Spring Boot configurations

**Interview 29**

 Interview was good. Mostly they were asking about Java 8  
  
1: Java 8 filtering related questions  
2: merging two arrays using Java 8 .

3: Basics of Kafka not mandatory to answer everything but if you know any Messaging Queue that should be fine.

4: Fetch documents from mongoDB or any NoSQL based on certain condition.

5: Microservice related Exp, means what was the project Architecture and roles and Responsibility.  
6: System designee related questions with use of Kafka, ELK stack with microservice.

**Interview 30:**

Walkthrough of experience.

Explain most recent project tech stack & architecture.

Write the code to perform any Get operation end to end. Explain while writing the code.

What is @RequestMapping?

How do u maintain versioning of API when you have to retain older version as well?

Difference between PUT & POST?

Difference between PUT & DELETE?

What if you have to do if front end is not passing id as path variable or query param? How else can it be passed? Can we use GET operation if it is passed in any other manner?

How will you implement/ send paginated data to front end?

How will you send sorted data to front-end?

Java 8 features. Explain the one's you have used in earlier projects

Write a code to sort a list and append "aaa" using Java 8

What is Docker?

Have you worked on reactive programming? GCF?

Have you worked on swagger? How to generate it? What is a yaml file?

**Interview 31:**

1. Technology stack used in previous project  
2. Find 3rd highest element from the given list (collection) - coding  
3. Separate even and odd numbers using streams - coding  
4. Design patterns used in our project.  
5. Functional interfaces   
6. Why only one abstract method allowed in functional interfaces.

**Interview 32:**

1. How can we make Singleton design pattern and also make it thread safe.

2. Scenario based question on design pattern: Home Alarming System designing

3. Transaction Management in cases where server had failures triggered.

4. Saga pattern for Microservices

5. Design Code for finding occurrences of characters in a String like

     Input:  aabedrdraa

     Output: [a=4, b=1, e=1, d=2, r=2]

6. Functional Interfaces in java 8

7. Why multiple inheritance is not supported in Java

8. Static Binding and Dynamic Binding

9. An example in your project where you have implemented Multithreading concept

10. Thread local variable in java

11. Memory leak in java and scenario where you have implemented this concept.

12. Approach for handling issues in legacy/monolithic systems like System ran OutOfMemoryError exception.

**Interview 33:**

As per my experience the client is more focused java 8, and Microservices implementation apart from this please find the below questionnaires which were asked to me in the interview :

1. Java 8 feature

2. Two coding questions on streams (1st. Check if two strings are anagram. 2nd. Check if integers are meeting the criteria and store them in list)

3. Design patterns used in Microservices Architecture.

4. Spring MVC flow.

5. Spring bean scope and lifecycle.

6. Spring boot autoconfiguration

7. Rest API example with db mapping.

8. How to use Docker

9. How to scale up Microservices using kubernetes

10. Few questions on the last project which I worked on.

11.  HashMap internal implementation

**Interview 34:**

***First Round:***

Walkthrough Experience and Technology Stack.

Explain about project architecture

OOPS concept with real world example

What will be the output of the below program and why?  
String x = "Xyz abc";  
x.append("pqr");  
x.append("qwe");  
System.out.println(x)

Explain java 8 features.

Difference between Map and Flatmap in java 8

Write a program to exclude given words from a given list.   
Given List: List<String> lines = Arrays.asList("demo", "test", "IBM", “test”);  
Word to exclude: test

Count occurrence of characters from given string.   
String x = "aaabbabbadddccc"

Find missing number from given list of 1 to 100 numbers

Difference between @PutMapping and @PatchMapping

Advantages of spring boot over spring.

Difference between Prototype scope and Request Scope of bean.

***Second Round***:

Walkthrough Experience and Technology Stack.

Explain about project architecture

How will you find the most selling product from an ecommerce site?

Write proper flow from controller service dao layer for @Get, @Post @Put @Patch mapping.

Difference between @Get and @Post? (What is idempotent method)

Explain about RentrantLock? Asked to write thread implementation, Executor Service.

How to maintain Transaction in the dao layer in two different methods.

How to maintain distributed transaction in Microservice architecture

What is saga design pattern?

Circuit breaker design pattern

Singleton design pattern and how many ways we can break it? And how to prevent it from breaking.

**Interview 35:**

Java 8 features:

What are all the new features in java 8?

Questions on streams, filter we need to write a code for given scenario?

Explain what is functional Interface why default & static methods in interface?

Collections: List, LinkedList, Map & Set implementations

Microservices:

Project Architecture.

Monolithic application vs microservices architecture. How would you break a monolithic app into microservices?

What are the advantages of microservices over monolith and vice-versa?

What is Circuit Breaker, Explain?

How do microservices communicate with each other?

Explain me about logs

Other:

 Some questions on Kafka

 Questions on design pattern

 Write a program to count number occurrence of word for a given String

**Interview 36:**

Describe about current project

Write a code to check the string is palindrome, after ignored the first letter

What design patterns you are aware of.

How microservices communicate each other

Function overloading and overriding

Differentiate Hashtable and HashMap

Failsafe and fail fast

List the request methods in rest API, and differentiate PUT vs POST vs FETCH

Serialization in spring boot

Constructor overloading and copy constructor: *A copy constructor in a Java class is a constructor that creates an object using another object of the same Java class. That's helpful when we want to copy a complex object that has several fields, or when we want to make a deep copy of an existing object*.

**Copy Constructor vs. Clone:** In Java, we can also use the clone method to create an object from an existing object. However, the copy constructor has some advantages over the clone method.

1. The copy constructor is much easier to implement. We do not need to implement the Cloneable interface and handle CloneNotSupportedException.
2. The clone method returns a general Object reference. Therefore, we need to typecast it to the appropriate type.
3. We cannot assign a value to a final field in the clone method. However, we can do so in the copy constructor.

Final constructor is possible in Java? **No, a constructor can't be made final**. A final method cannot be overridden by any subclasses

Write a query to find the employee in each branch who‘s is salary greater than 10000.

Autoboxing and unboxing

Transaction in spring

What is IOC and dependency injection?

What is autowiring?

What is qualifier annotation?

What is cyclic dependency?

Httpv2

Caching in spring and distributed cache

Life cycle bean

Spring session

gRPC

**Interview 37**

1. Introduce yourself.

2. What are the technologies you have worked upon in your last project?

3. Have you worked upon Microservices? What do you understand basically by Microservices?

4. What all design patterns you have worked upon Microservices. Explain any one of them.

5. What are the 12 factor principles for Microservices?

6. What do you understand by functional interface?

7. Why do we have functional interface?

8. Why do we use Streams?

9. Difference b/w HashMap and LinkedHashMap?

10. Have you worked on ConcurrentHashMap? If Yes, Explain.

11. Is HashMap thread safe? Please Explain in both the cases either yes or no.

12. Suppose there is a class. You have to write a junit for that class and also that class calls Rest API and it receives JSON response and that JSON response can be inserted in mongo DB.

13. How you'll mock the database?

14. What all design patterns you have worked upon in Java?

15. Explain Singleton Design pattern.

16. How you'll ensure that only one instance is present?

17. Why do we use private constructor in Singleton?

18. Difference b/w notify and notifyAll?

19. What are different scope of spring bean? Explain each one of them.

20. What is the use of @Autowired annotation?

21. What is the usage of @Component?

22. Which cloud technologies you have worked on?

**Interview 38**

1. Introduce yourself.

2. Explain the last project you worked upon?

3. How many microservices your project were using? How connection was established b/w the microservices?

4. What kind of use cases is required for sync and async connection b/w microservices i.e. in which scenario we'll use sync or async?

5. Explain design patterns you worked upon in your last project?

6. Explain Singleton design pattern.

7. Have you ever used circuit breaker pattern? Why we use this pattern? We can let anyway fail. So what is the basic requirement for this implementation?

8. Main collection on which you have worked upon?

9. Suppose you have ArrayList containing String value "Shivangi" and HashMap as well having key and value String respectively where key is "Shivangi" and value as "Shivangi”. So how we'll get the value from ArrayList and HashMap? What is the method we'll use to get the values from HashMap?

10. How we'll do search by object in ArrayList?

11. What is the difference in the get method of ArrayList and HashMap?

12. Which one among ArrayList or HashMap is more optimised and faster?

13. Suppose you have a stream of employee objects which have name, id and age respectively. So Among those employees, I want to filter out employees having age >20. How can I do that?

14. Instead of collecting it into list I want to convert it into a map having key as employeeId and the value as name. How can we do that?

15. Write a code to group all the employees together having age 20, 21 and 22 in the streams and collect.

16. Write pseudo code for Singleton design pattern.

17. How we implement thread in Java?

18. Write odd and even number from a list of integers using two threads. You can use 1.7 or 1.8.

19. Write a pseudo code for implementation of threads in both the ways by extending thread class and by implementing Runnable interface.

20. Explain Kafka architecture.

21. We are having one microservice "A" which is sending message and another microservice "B" which is receiving message. How that interaction will happen with Kafka?

22. What is role of Broker?

23. What is Zookeeper?

24. Let’s say we have Microservice 1 which sends the msg to topic "A" and there is microservice 2,3and 4 which want to listen to the same Kafka topic "A”. Microservice 2 listens to the message first. So now can Microservice 3 and 4 listens to the same message again or will it be taken by Microservice 2 as it has listened to the message first?

25. How we'll migrate from Spring to Spring Boot? What all basic things are required to do the same?

26. Do you know stereotypes?

27. Do you know profiling?

28. What is the dev tool in Spring Boot? Its purpose.

29. What is Qualifier annotation?

**Interview 39**

Write a program that, given an array A[] of n numbers and another number x, determines two elements whose sum is exactly x.

Examples:

Input: arr[] = {0, -1, 2, -3, 1}

sum = -2

Output: -3, 1

Migrate spring application to Spring boot application

Spring validation

Previous Project Architecture

Docker file to run spring boot application

Functional Component vs Class Component in react

How to incorporate external configuration in Spring boot

Microservice vs Monolithic application

**Interview 40**

Write a program that, given an array A[] of n numbers, determines number of occurrence for first repetitive number

Examples:

Input: arr[] = {1, 2, 7, 2, 1, 6, 8, 1}

Output: 1 = 3

Messaging Queue concept

Spring bean life cycle

How internally works concurrent HashMap

Different types of thread lock

Microservice advantages and disadvantages

Dependency Injection concept

**Interview 41**

Write a program to find all permutations of a given string

Previous Project Architecture

Microservice design pattern

Controller vs Rest Controller

MongoDB vs RDBMS

Sharding in MongoDB

How to manage distributed logging mechanism in application

Spring security implementation

**Interview 42**

Pravin Patil team

Q-1) Java 8

Given Object Structure

Institute

List<Subject> listSubject

List<Student> listStudent

Find the list of All Student Name

Find Each Subject Highest Marks

Q-2) Design Pattern

Composite Design Pattern

Singleton Design Pattern

Q-3) Java OOPs Concept

Which one you prefer and Why?

1. class A extends B

2. class A

private B

Q-4) Spring Boot Pagination and Sorting

Q-6) @Service vs @Configuration

Q-7) Kafka security certificate path configuration

Q-8) Kafka service schema registry and serialization

Q-9) Kafka topic and offset

Q-10) how to apply fix on production environment

Interview Questions:

**Interview 43**

Reactive programing? Map vs FlatMap in reactive programing using webflux?

Difference between Mono and Flux?

BDD testing framework experience?

Different testing strategies used?

Spring boot Test containers?

Spring cloud and Config?

How to setup Jenkins pipeline and how-to setup the stages?

Linear data structures and multilevel data structures?

Programing:

Print duplicated character count in a string?

Sort List or Stream of strings based on its length?

Implement stack data structure?

Interview Questions:

->solve the below program using recursion:

The Tribonacci sequence Tn is defined as follows: T0 = 0, T1 = 1, T2 = 1, and Tn+3 = Tn + Tn+1 + Tn+2 for n >= 0.

Given n, return the value of Tn.

->explain advantages and disadvantages of microservices.

->explain about some design patterns that you have followed in your microservices project.

->what are the design pattern you used in your spring application directly or indirectly?

->can you give me some example of singleton design pattern and factory design pattern from spring framework?

->when you should go for implementing messaging services in micro services.

->how can you achieve asynchronous calls in spring?

->what you know about NoSQL DB

**Interview 44**

1. Explain architecture of your project

2. Explain java 1.8 features that you have used.

3. Why functional interface and lambda and what are the advantages.

4. Explain what all you know about microservices

5. What is cloud Config in microservices?

6. What are all the annotations you have used in spring boot?

7. What are the ways to initialize a bean in spring?

8. Explain about bean life cycle.

9. Do you used any NoSQL DB and if yes then which DB and tell me your experience.

10. What are all the things you have worked on cloud?

11. What is Docker and what is your experience into that.

12. What is Kubernetes?

13. What are the design patterns you have used in your microservice projects?

14. What is anonymous class and what is the relation between lambda and anonymous classes.

**Interview 45**

Tech Interview:

Tell me about yourself.

Rate yourself as a java developer

What are the new features of java 8?

What is stream?

What do you mean by stream pipeline?

Define terminal operation of a stream and name few terminal operations.

What is filter in stream?

What is Flatmap in stream?

What is the difference between map and Flatmap?

What is lambda?

What are the different forms of lambda?

***What is recursive lambda***: A recursive lambda expression is the process in which a function calls itself directly or indirectly is called recursion and the corresponding function is called a recursive function. Using a recursive algorithm, certain problems can be solved quite easily.

Discussion on the lambda

Write a program to print duplicate integer from a list of integers using stream?

Write a program to find out the number which are start with 1 from a list of integers using stream?

Find out the first non-repeated character from a string using stream?

What are the collection you have mostly used?

When to use ArrayList and when to use LinkedList and why

What is synchronized ArrayList

Is ArrayList thread-safe if not what step need to take to make it thread-safe?

Implementation of HashMap

How can we prevent Concurrent Modification Exception?

What are new features in date time API with respect to java 8?

How much you rate yourself as microservice developer

What is monolithic application?

What are the component you used for microservice application?

How two microservices communicate

What is @RequestMapping and @RestController

What is @RequestMapping and @GetMapping

What is EnableAutoConfiguration and SpringbootApplication Annotation?

How you can change the port for a service in spring boot application

What need to be done to run application in a different container?

What is the advantages of rest?

What is jpql?

What is the ***difference between jpql and sql***: The main difference between SQL and JPQL is that SQL works with relational database tables, records and fields, whereas JPQL works with Java classes and objects. *For example*, a JPQL query can retrieve and return entity objects rather than just field values from database tables, as with SQL.

What is EntityListner?

How to create a custom repository

What are the different mapping present in data jpa give example of many to many?

***Meaning of fetchType=lazy***: It fetches the child entities lazily, that is, at the time of fetching parent entity it just fetches proxy (created by cglib or any other utility) of the child entities and when you access any property of child entity then it is actually fetched by hibernate

How MySQL works

What are the different numeric datatype in MySQL?

Saga Design Pattern

Bucket creation in AWS

Spring life cycle bean

***Volumes in Docker***: Volumes are the preferred mechanism for persisting data generated by and used by Docker containers.

While bind mounts are dependent on the directory structure and OS of the host machine, volumes are completely managed by Docker.

***Daemon port of Docker***: Access to the Docker daemon is gained through the **tcp socket 2375**, available to all that can connect to the host. #C - Third parties can access this Docker daemon. The Jenkins server and colleague's host connect to the host's IP address on port 2375 and can read and write requests and responses using that channel.

About Redis stale data  
Redis cache pain points

SpanId in microservices.

About RabbitMQ