Java Internal & Client Interview Questions

**City Client Interview:**

1.Given list of string "cat" , “act”, "dog", "god" , "eat" , "ate"

Group the element based on anagram words, you need to use streams

eg, eat and aet should come in one group

2. given list of string "potato:veg" , "mango:fruits","banana:fruits"

use stream print the elements divided into two categories fruits and vegs

output should be fruits=[banana, mango], veg=potato

3. How many elements would there be in HashMap

Map hm = new HashMap(); you need to explain the reason as well

hm.put("hello", 1);

hm.put(new String("hello"),2);

Spring boot feature/advantages

database connection

Contract b/w hashmap and equals

what is immutable?

Spring annotations

REST API

Sum of salary for each emp in each year

Java 17

**Fiserv client:**

@Request mapping

Polymorphism

Mulithreading

Excuter service

Synch block and Asynch block in thread

Compartable feature - java 21

Treemap and linkedlist

Linked hashmap insertion order

Rest Apis

Db configuration

Pagination

hashcode and equals in hashmap

Service discovery and load balancer, circuit breaker, kafka and rest template

Sql - DML, Inner join, Sequence , how to increment the primary key.

Circuit breaker implementation

API gateway and service discovery coding?

JWT dependency ?

Rest template implementation coding?

sort the list of employees and group the employees

List of string print the first non repeated character ?

How to know application.properties ?

**SCB Client:**

static disadvantage?

class ,instance and local variable?

hashmap internal working? collision?how to overcome?

interface1,interface 2 both have a same method how to implement?

String test=" } Hi everyone,} welcome. Have{} a great { day {"; ? check the braces open and close?

where u will use the comparable and comparator?

Linked and array list ?where u will use?

Springboot Advantage?

Abstract class and interface class? java 8 which scenario used?

constructor when u overload and override?

Thread?

Sql?

shell script?

junit?

**Questions from other client:**

1)List<Integer> myList = Arrays.asList(10,15,8,49,25,98,98,32,15);

print the list if numbers in ascending order using streams API.

What are the terminal and intermediate methods that you use to write code.

explain method reference that you use to write code.

.what are the features of java 8.

2).what is internal process functional interface and lambda expression.

3).write sum of two integer numbers using lambda and its internal process.

4).(int a,b) -> {(a+b)}; how to write in correct format and i need datatypes also.

5).what is Optinal class and its methods and when do you use.

6).List<Integer> myList = Arrays.asList(10,15,8,49,25,98,98,32,15);

print the list if numbers in ascending order using streams API.

What are the terminal and intermediate methods that you use to write code.

explain method reference that you use to write

7) java 17 features with examples?

8) What is immutability in java?

9) what are the features of java 8?

10) what are the intermediate and terminal Operations of Stream?

11)What are marker interfaces in java?

12)What is serialization in java?

13)What are Exception Handling mechanisms in java?

14)What are default and static methods in interfaces?

15)Explain about Tree – order traversal with time complexity?

16)If two threads trying to access same resource at same time what are the problems will happen and what is the solution ?

core java to spring boot . No Microservice Question .

Functional Interface ?

Interface vs Abstraction ?

Internal working Mechanism of Hash Map ,Hash set and Hash Table ?

How map works ?

coding questions to get the numbers from the list starts with one ?

Spring over Springboot ?

Threads ?

Maven , how you update dependencies ?

Yml file in production

How to Debug the large set of logs

Parent child table update using jpa

Static analysis ?

scheduler in microservice

api gateway ?load balancer ? in code

@primary @secondary ?

program:

List of string ==>starts with A:

Polymorphism

@component and @bean?

Check the logic:

class a

class b extends class a

A a = new B();

B b = new A();

this is correct approch?

Springboot restapi how to create postmethod?

@path vs @request param?

@bean used for which scenario?

OOps concepts:

arraylist vs linked list?

Internal working of hash set and hash map?

where we are using singleton? how to make a singleton?

Java8 features?

program:

List of string ==>starts with A:

@component and @bean?where we used?

In spring, how to call super class from sub class

@autowired?

Multithread and concurrent multi-thread ?

jdk vs jvm vs jre ?

mockito ?

variables types ?

static disadvantage?

A major disadvantage of static elements is their lack of flexibility, requiring manual updates and limiting scalability.

class ,instance and local variable?

hashmap internal working? collision?how to overcome?

- Separate Chaining: Each bucket contains a linked list (or a balanced tree in Java 8+) to store multiple entries that hash to the same index.

- Open Addressing: Instead of storing multiple entries in a bucket, this method finds another available slot in the array.

- Improved Hash Functions: Using a well-distributed hash function reduces the likelihood of collisions.

- Balanced Trees (Java 8+): When collisions exceed a threshold, HashMap switches from linked lists to balanced trees for better performance.

interface1,interface 2 both have a same method how to implement?

interface Interface1 {

default void show() {

System.out.println("Interface1's show method");

}

}

interface Interface2 {

default void show() {

System.out.println("Interface2's show method");

}

}

class MyClass implements Interface1, Interface2 {

@Override

public void show() {

System.out.println("Overridden show method in MyClass");

}

}

public class Main {

public static void main(String[] args) {

MyClass obj = new MyClass();

obj.show(); // Calls the overridden method in MyClass

}

} Output: Overridden show method in MyClass

String test=" } Hi everyone,} welcome. Have{} a great { day {"; ? check the braces open and close?

class Sample{

public static void main(String args[])

{

String s= "{}}{";

int count=0,i;

for(i=0;i<s.length();i++)

{

if(s.charAt(i)=='{')

{

count++;

} else if (s.charAt(i)=='}') {

count--;

if(count<0)

{

break;

}

}

}

if(count==0)

{

System.out.println("Balanced");

}

else

{

System.out.println("Unbalanced");

}

}

}

where u will use the comparable and comparator?

- Comparable (Natural Ordering): Defines the default sorting logic inside the class using compareTo(). It’s used when objects have a single natural order, like sorting students by age.

- Comparator (Custom Ordering): Used outside the class to define multiple sorting criteria, like sorting students by name or age dynamically.

Linked and array list ?where u will use?

Springboot Advantage?

Abstract class and interface class? java 8 which scenario used?

// Interface with default method (Java 8 feature)

interface Animal {

void sound(); // Abstract method

default void sleep() { // Default method (Java 8)

System.out.println("Sleeping...");

}

}

// Abstract class with concrete and abstract methods

abstract class Mammal {

abstract void walk(); // Abstract method

void breathe() { // Concrete method

System.out.println("Breathing...");

}

}

// Class implementing interface and extending abstract class

class Dog extends Mammal implements Animal {

@Override

public void sound() {

System.out.println("Barking...");

}

@Override

void walk() {

System.out.println("Walking on four legs...");

}

}

public class Main {

public static void main(String[] args) {

Dog dog = new Dog();

dog.sound(); // Calls interface method

dog.walk(); // Calls abstract class method

dog.breathe(); // Calls concrete method from abstract class

dog.sleep(); // Calls default method from interface

}

}

constructor when u overload and override?

constructors can be overloaded, but they cannot be overridden.

Why Constructor Overriding is Not Possible?

- Constructors are not inherited in Java.

- Overriding requires inheritance, but constructors belong to their own class.

- A subclass can call a superclass constructor using super(), but it does not override it.

Thread?

Sql?

shell script?

A shell script is a text file containing a series of commands that are executed by the shell (such as Bash, Zsh, or sh).

These scripts can automate repetitive tasks and improve efficiency.

junit?

Exception (in catch block exception)?

- If either exception occurs, the same catch block handles it.

- This approach prevents code duplication if the handling logic is the same.

try {

int result = 10 / 0; // ArithmeticException

} catch (ArithmeticException | Exception e) {

System.out.println("Exception caught: " + e.getMessage());

}

Asynchoronus thread how to implement? thread lifecycle?

- Synchronous: Execution happens sequentially, one task must finish before moving to the next.

- Asynchronous: Execution happens independently—multiple tasks can run concurrently.

- Asynchronous: Uses start(), creating a new independent thread.

- Synchronous: Uses run(), executing within the main thread without multithreading.

Synchronous version:

class MyThread extends Thread {

public void run() {

System.out.println("Thread running synchronously.");

}

}

public class Main {

public static void main(String[] args) {

MyThread thread = new MyThread();

thread.run(); // Executes in the main thread synchronously

System.out.println("Main thread continues after execution.");

}

}

Asynchoronus version:

class MyThread extends Thread {

public void run() {

System.out.println("Thread running asynchronously.");

}

}

public class Main {

public static void main(String[] args) {

MyThread thread = new MyThread();

thread.start(); // Starts the thread asynchronously

}

}

paralellel stream?

Key Differences: Sequential vs. Parallel Stream

| Feature | Sequential Stream (stream()) | Parallel Stream (parallelStream()) |

| Execution | Processes elements one-by-one | Processes elements concurrently |

| Thread Usage | Uses single thread | Uses multiple threads |

| Performance | Good for small data sets | Faster for large collections |

| Ordering | Preserves order | May not preserve order |

intermediate and terminal in stream?

performance tunning and optimize in sql?

- Indexes speed up data retrieval by allowing the database to search efficiently instead of scanning the entire table.

- Without an index, searching for name = 'Alice' would require checking every row.

- With an index, the database can locate Alice much faster.

2. Breaking Down the Statement

CREATE INDEX idx\_customer\_name ON customers(name);

- CREATE INDEX idx\_customer\_name → Creates an index called idx\_customer\_name.

- ON customers(name) → Index is applied to the name column of the customers table.

Joins AND subquery ?

✅ Use JOIN when pulling data from multiple tables efficiently

✅ Use Subquery when filtering based on another table’s data

Unix commands?

In Java, you can execute Unix commands using the ProcessBuilder or Runtime.exec() methods.

These allow Java to run shell commands like ls, grep, cat, and more within a Unix-based system.

Use ProcessBuilder for complex commands and argument passing

✅ Use Runtime.exec() for simple execution but avoid for multiple arguments

✅ Always handle exceptions when executing system commands

**Internal Interview:**

Current project details

What is the difference between execution in executor framework?

What is blocking queue?

https://www.google.com/search?q=what+is+blocking+queue+in+java&sca\_esv=e948bafec9f39f8c&ei=\_6SIaJP8I9nH4-EPgaCgsQ4&oq=What+is+blocking+queue%3F&gs\_lp=Egxnd3Mtd2l6LXNlcnAiF1doYXQgaXMgYmxvY2tpbmcgcXVldWU\_KgIIADIFEAAYgAQyBRAAGIAEMgYQABgWGB4yBhAAGBYYHjIGEAAYFhgeMgYQABgWGB4yBhAAGBYYHjIGEAAYFhgeMgYQABgWGB4yCxAAGIAEGIYDGIoFSPcrUPAIWKAUcAN4AZABAJgBgwGgAYMBqgEDMC4xuAEDyAEA-AEC-AEBmAIEoAKNAcICChAAGLADGNYEGEeYAwCIBgGQBgKSBwMzLjGgB6oGsgcDMC4xuAeGAcIHAzAuNMgHBw&sclient=gws-wiz-serp

How do we know thread pool is exhausted or has empty thread?

Different design patterns?

What if we initialize Date class object in singleton class, will the date object be singleton too?

What design pattern is used with async data update between different clients?

Different design patterns.

What is the return type of different List interface methods

What is the role of zookeepers?

How to handle failure retry in Kafka

Spring boot annotations

Write controller class

stream api v/s parallel streams api

Pagination Vs JPA Repository

Different – different design patterns

database related questions, Cursor, Normalization, query

Spring security – Login process / how to implement the login and which springboot application you use

Annotations

Database connections

which version of java you worked on

Any exposure on using Lambda function

multithreading

Which framework of java is used for multithreading

Which executor services have you used

What is the type of Executor service and for what you have used executor service

What are thread pools? Have u used cache thread pool

New Fixed thread pool and new Fixed Executor

What are different types of queue in java, synchronous queue

Which version of spring boot you worked on

How do you claim that you have worked on microservice?

wallet service in java spring boot

What is the use of map and flat map in stream

What is a weak hash map

How will you convert string in to array and make it use

If 25 connections are required, then how will you set it up in microservices

What is dialect

Return type of Void method

Java 8 or Java 17

Difference in java 8 and java 21

Explain pagination concept and types of pagination with working example

Explain Kubernetes config

project related question

Design pattern in microservices

You have call system A. how to call system B using service line

What is the purpose of Lombok

What is the purpose of records in Java? Have you used Java records earlier?

Rest end point

Programming questions

User input: abc-123+xyz\_897^a12

Output: [abc,xyz]

[123,897]

[a12]

User input: abc-123+xyz\_897^a12 fgh

Output: [abc,xyz]

[123,897]

[a12]

{

"inputs": [

{

"input": ["abc", "lmn", "ijk"]

},

{

"input": ["hg", "ieo"]

},

{

"input": ["df"]

}

]

}

{"output":["lmn","hg","df"]}

Write controller and create Get API to fetch employee record and Post API to create employee record

Create database connection.

Explain H2 database with example

Write third party api mappings

**Apache Kafka: -**

1. What is Apache Kafka, and how does it differ from traditional message brokers?

2. Explain the role of Kafka producers and consumers. How does Kafka ensure fault tolerance and data durability?

3. Discuss the concept of topics and partitions in Kafka. How does partitioning contribute to scalability and parallelism?

4. How can you achieve message ordering and exactly-once semantics in Kafka?

5. Describe Kafka's role in event sourcing and stream processing architectures.

6. Explain Kafka's use in a microservices environment. How can microservices communicate and share data using Kafka?

7. What is Kafka Connect, and how can it be used for data integration between Kafka and external systems?

8. Discuss Kafka's role in log aggregation and real-time data processing.

9. Explain how Kafka ensures data retention and handles data compaction.

10. Describe Kafka's support for transactional messaging. How are transactions managed in Kafka?

<https://www.hackerrank.com/domains/java>

**Easy Medium Hard & Advanced Level Of Interview Questions:-**

1. Easy Level Spring Boot Interview Questions

# Question

1 What is Spring Boot?

2 What are the advantages of using Spring Boot?

3 How is Spring Boot different from the traditional Spring framework?

4 What is the role of @SpringBootApplication annotation?

5 What is an embedded server in Spring Boot?

6 What is the purpose of the application.properties or application.yml file in Spring Boot?

7 What is spring-boot-starter?

8 How do you run a Spring Boot application?

9 What is a Spring Boot actuator?

2. Medium Level Spring Boot Interview Questions

# Question

10 What is auto-configuration in Spring Boot?

11 What are Spring Boot profiles?

12 How does Spring Boot handle logging?

13 What are @RestController and @Controller in Spring Boot?

14 What is Spring Boot DevTools?

15 How can you configure a Spring Boot application to connect to a database?

16 How do you configure custom properties in Spring Boot?

17 What is the role of @EnableAutoConfiguration in Spring Boot?

18 How do you create a custom starter in Spring Boot?

19 What is Spring Boot’s embedded Tomcat server?

20 How do you perform validation in Spring Boot applications?

3. Hard Level Spring Boot Interview Questions

# Question

21 How does Spring Boot handle security?

22 How do you manage externalized configuration in Spring Boot?

23 How can you integrate Spring Boot with Spring Data JPA?

24 How can you manage Spring Boot application versions?

25 What are Spring Boot Actuator endpoints?

26 How do you implement custom error handling in Spring Boot?

27 How do you deploy a Spring Boot application to production?

28 How do you handle transactions in Spring Boot?

29 What is Spring Boot’s testing support?

30 What is the difference between @SpringBootTest and @WebMvcTest?

31 How does Spring Boot support asynchronous processing?

4. Advanced Level Spring Boot Interview Questions

# Question

32 How do you configure Spring Boot for microservices architecture?

33 How do you implement Circuit Breaker in Spring Boot?

34 What is Spring Boot’s support for Docker?

35 How do you implement distributed tracing in Spring Boot?

36 What is Spring Boot’s support for monitoring?

37 How does Spring Boot handle background tasks?

38 How do you configure Spring Boot for handling large-scale applications?

39 How do you implement multi-tenancy in Spring Boot?

40 How do you handle state management in Spring Boot for stateless applications?

**Internal & Other Client Interview Questions:-**

Current project details

What is the difference between execution in executor framework?

What is blocking queue?

How do we know thread pool is exhausted or has empty thread?

Different design patterns?

What if we initialize Date class object in singleton class, will the date object be singleton too?

What design pattern is used with async data update between different clients?

Different design patterns.

What is the return type of different List interface methods

What is the role of zookeepers?

How to handle failure retry in Kafka

Spring boot annotations

Write controller class

stream api v/s parallel streams api

Pagination Vs JPA Repository

Different – different design patterns

database related questions, Cursor, Normalization, query

Spring security – Login process / how to implement the login and which springboot application you use

Annotations

Database connections

which version of java you worked on

Any exposure on using Lambda function

multithreading

Which framework of java is used for multithreading

Which executor services have you used

What is the type of Executor service and for what you have used executor service

What are thread pools? Have u used cache thread pool

New Fixed thread pool and new Fixed Executor

What are different types of queue in java, synchronous queue

Which version of spring boot you worked on

How do you claim that you have worked on microservice?

wallet service in java spring boot

What is the use of map and flat map in stream

What is a weak hash map

How will you convert string in to array and make it use

If 25 connections are required, then how will you set it up in microservices

What is dialect

Return type of Void method

Java 8 or Java 17

Difference in java 8 and java 21

Explain pagination concept and types of pagination with working example

Explain Kubernetes config

project related question

Design pattern in microservices

You have call system A. how to call system B using service line

What is the purpose of Lombok

What is the purpose of records in Java? Have you used Java records earlier?

Rest end point

Programming questions

User input: abc-123+xyz\_897^a12

Output: [abc,xyz]

[123,897]

[a12]

User input: abc-123+xyz\_897^a12 fgh

Output: [abc,xyz]

[123,897]

[a12]

{

"inputs": [

{

"input": ["abc", "lmn", "ijk"]

},

{

"input": ["hg", "ieo"]

},

{

"input": ["df"]

}

]

}

{"output":["lmn","hg","df"]}

Write controller and create Get API to fetch employee record and Post API to create employee record

Create database connection.

Explain H2 database with example

Write third party api mappings

https://capgemini.udemy.com/course/apache-kafka-for-beginners/?start=0

https://capgemini.udemy.com/course/spring-boot-and-apache-kafka/

https://medium.com/@fromFullStack/top-25-kafka-interview-questions-867a5d8f31d8

**Interview coding questions:**

1. List<String> Str = Arrays.asList("Apple", "Bat", "Orange", "Avacado", "Banana", "Aeroplane");

Find the string which starts with 'A'

List<String> Str = str.stream().map(s -> s +" ").filter(i -> i.startsWith("A")).collect(Collectors.toList());

Find the longest string word

String logestStringWord = str.stream().max(Comparator.comparingInt(String::length).orElse(null);

2 String s="swiss";

Find the first count of each char.

Map<Character, Long> countOfEachChars = s.chars().map(c -> (char)c).collect(Collectors.groupingBy(Function.identity(),Collectors.counting());

System.out.println(countOfEachChars)

Find the first non repeated character

**Scenario-based questions:**

1. Microservices Architecture

1. Scenario: Your organization is planning to move from a monolithic application to a microservices-based approach. What are the major architectural changes you would consider?

Answer:

-------

Moving from a monolithic application to a microservices architecture involves significant architectural changes. Key areas to consider include breaking down the application into smaller, independent services, defining service boundaries, establishing communication mechanism between services, and adapting deployment and monitoring processes

2. Scenario: You have a microservices architecture with multiple services communicating over HTTP. One of the services is becoming a bottleneck due to heavy traffic. How would you identify and mitigate this issue?

Answer:

-------

Monitoring and Alerting:

Distributed Tracing:

Profiling:

Scaling:

Optimization:

Asynchronous Communication:

Load Balancing:

3. Scenario: Your microservices are built using different technologies (e.g., Java, Node.js, Python). What challenges might arise from such heterogeneity, and how would you handle them?

Answer:

-------

Inter-Service Communication:

Security Consistency:

Standardized Communication Protocols:

Containerization and Orchestration:

Centralized Observability:

API Gateway:

Cross-Functional Teams:

Automated Testing and CI/CD:

4. Scenario: You are tasked with designing a highly available microservices system. What architectural strategies would you use to ensure fault tolerance and high availability?

Answer:

---------

load balancing, redundancy, circuit breaking, health checks, graceful degradation, and automated failover.

5. Scenario: A new service needs to be added to the microservices architecture. How would you ensure that it integrates seamlessly without impacting existing services?

Answer:

-------

utilize API gateways for routing and security, implement robust testing strategies (including end-to-end testing), and ensure loose coupling between services.

6. Scenario: How do you handle transactions in a distributed system where microservices maintain their own databases? What patterns could you use to ensure consistency?

Answer:

-------

Sagas, eventual consistency, and compensating transactions are employed.

7. Scenario: Your microservices are handling sensitive data. What security practices would you implement to ensure data protection in a distributed environment?

8. Scenario: A service in your microservices architecture requires heavy state management. How would you design it to avoid any performance issues or bottlenecks?

Answer:

-------

Event Sourcing and CQRS (Command Query Responsibility Segregation):

Caching:

Database Sharding and Partitioning:

Asynchronous Processing and Message Queues:

Optimized Data Models:

Monitoring and Observability:

9. Scenario: You want to implement service discovery for your microservices. How would you go about doing that in a Kubernetes environment?

Answer:

-------

DNS-based Service Discovery:

10. Scenario: Your organization needs to scale microservices efficiently to handle growing traffic. What patterns or technologies would you use to manage scalability?

Answer:

-------

Horizontal scaling, load balancing, and container orchestration

2. Communication Between Microservices

1. Scenario: You need to communicate between two microservices in real-time with minimal latency. What communication protocol would you choose, and why?

Answer:

-------

gRPC is generally the best choice

2. Scenario: Microservices need to send large payloads to each other. How do you ensure that the data transfer is efficient and doesn't cause performance degradation?

Answer:

-------

Optimize Communication Protocols: gRPC, Message Queues/Brokers:

Payload Optimization: Compression, Serialization, Data Partitioning/Chunking:

Offload Large Data to External Storage: Object Storage, Dedicated Database:

Asynchronous Communication:

Network Optimization:

3. Scenario: Service A needs data from Service B, but Service B is unavailable. How would you handle this situation without affecting the user experience?

Answer:

-------

Implement Fallbacks:

Utilize the Circuit Breaker Pattern:

Asynchronous Communication and Retries with Exponential Backoff:

Graceful Degradation:

Caching:

User Notification and Options:

4. Scenario: You need to ensure that Service A does not overload Service B with requests. How would you design the communication to avoid this scenario?

Answer:

-------

To prevent Service A from overloading Service B, a design incorporating rate limiting, circuit breakers, queues, and potentially a message broker should be implemented.

5. Scenario: Your microservices are communicating over an asynchronous messaging system (e.g., Kafka). How do you handle message ordering in the system?

Answer:

-------

Single Consumer per Partition:

Idempotency and Retries:

Sequence Numbers or Timestamps:

Compacted Topics:

Transactional Producers and Consumers:

6. Scenario: You are experiencing inconsistent responses from multiple microservices when using HTTP for communication. What debugging steps would you take to find the root cause?

Answer:

-------

Use monitoring and logging: Implement monitoring and logging tools to track the health and performance of microservices. Use this data to ...

How to handle data inconsistencies in microservices?

Here are 10 effective methods:

Saga Pattern.

Event Sourcing.

CQRS.

Distributed Transactions.

API Composition.

Database per Service.

Eventual Consistency.

Change Data Capture (CDC)

7. Scenario: You need to ensure that communication between microservices is secure. What steps would you take to ensure end-to-end encryption and authentication?

Answer:

-------

To ensure end-to-end encryption and authentication for microservices communication, use HTTPS with TLS/SSL for encryption, implement mutual TLS (mTLS) for mutual authentication, utilize API gateways for request routing and security policies, and consider service meshes for more advanced security features.

8. Scenario: Microservices need to interact with each other frequently but with low overhead. How would you design the communication to ensure performance is not negatively impacted?

Answer:

-------

Point-to-point communication ensures that only relevant services interact, reducing unnecessary overhead. b. Data Sharing: When one microservice needs data from another, it can request it directly through APIs or queries. This is particularly useful for applications requiring real-time data access.15 Sept 2023

Common Protocols Microservices Use in Communication:

-----------------------------------------------------

REST API Calls. REST API calls are common methods of microservices to communicate. ...

gRPC. gRPC, or Google Remote Procedure Call, is another high-performance protocol that enables microservices to effectively and efficiently communicate. ...

Message Brokers.

9. Scenario: A service is getting more traffic than expected, and it starts dropping requests. What strategies would you implement to manage the communication load efficiently?

Answer:

--------

1. Rate Limiting and Throttling:

2. Load Balancing:

3. Caching and Edge Computing:

4. Circuit Breakers:

5. Queueing and Prioritization:

6. Infrastructure Optimization:

7. Monitoring and Alerting:

8. Communication and Transparency:

10. Scenario: You need to set up communication between microservices while avoiding tight coupling. What design pattern would you use to keep the services loosely coupled?

Answer: Event-Driven Architecture (EDA) pattern

**Barclays Client Interview:**

1) find the string end with Y

List<String> monthsList =List.of("janvary","febrary","march");

List<String> endwithYmonthList=monthsList.stream().filter(month->month.endsWith("y")).collect(Collectors.toList());

System.out.println(endwithYmonthList);

2)Explain java 8 feature

3)Explain function interface. real time example

Ex. Predicate<String> predicate = (String s)->s.length()>5;

System.out.println(predicate.test("hello"));

4)when we used hash set & tree set

5)difference between array list & hash set

6)difference between authentication & authorization

7)Explain flow of spring security

8)Explain jvm memory management

9) difference between @Bean & @Componant annotation

10) which annotation was used in previous project

**Comcast Interview:**

1.project details, last implemented feature and challenges faced while doing

2. Java garbage collector

3.string joiner class

4.utility class and methods in it.

5.java 8 feature

6. Executor servise,future and completable future

6. Program to find non repeated character

7. Sql query to find second highest salary and count of male and female in each department

8. Scenario based on spring boot

**Sony Interview Questions:-**

Java

Sort the array based of O(n) or O(log n) complexeity

In an array list of employees find the employees with the third highest salary even if there are multiple entries.

Why are strings immutable and why are they immutable compared to other data types.

How does a HashSet work internally? What is the key and value here? What Data Structure does it internally use?

Recall any functional interfaces in Java

Marker Interfaces

How do you make a class immutable

Function, Predicate, BiConsumer, Consumer, Supplier

@Mock, @InjectMocks and @Spy

**Spring boot**

@SpringBootApplication consists of three annotations what are they

@Primary vs @Qualifier

@Component vs @Bean

@Transactional why we use

What is spring boot auto configuration

Application Context vs BeanFactory

Spring boot security(Oauth) bearer token and pop token

Authentication vs Authorization

Spring security flow by username and password

**Spring boot microservices**

What patterns I used(SAGA-Choreography and Orchestration)

Aggregator

Reverse Proxy

API Pattern

Two Face

Circuit breaker pattern

Kubernetes

How do we access the Kubernetes clusters

Ingress vs Egress

What is service in kubernetes.

Autoscaling

**BNPP interview:**

1. For the Given list of sales transactions (transactionID, productID, quantity, price), you have to find the top N products (by revenue)

sold in the last 3 months using Java 8.

2. Find the occurrence of given string - Her friends always walked hand in hand with her in her difficult times.

3. How to handle authentication and authorization in rest api

4. Have you worked kafka

5. How to exclude the class in your application

6. How handle exception in java

7. How to connect multiple database

8. How to disable password before running springboot application

9. How to replicate the production issue after deployment.

**Internal & Other Client Interview Questions:-**

find the longest substring without repeating characters without stream, pre-implemented method?

 Find the occurrence of the nth character in a given string like "HelloWorld".

Frequency of characters(need to write above code in java  (dont use streams)

Transactional Management Implementation

use of Transactional management

At which level we can use class level or method level at which layer

corsPolicy error

diff errors and exceptions

diff class and func component

Autowiring

types of autowiring

Springboot annotations(class level and method level)

What is DI and types, how many ways we have for configure DI?

useEffect

in frontend if we r not getting data from database which error it will b show and how it will b resolve

Due to infinite method call which error we will get and how can we resolve this issue?

In Spring Boot, how does data flow from the user interface to the database and backend

**B1-B2 Candidates**

1. Explain about your previous project

2. Explain about Spring boot Starter annotation – there is no annotation like but have dependencies

3. Explain about @SpringBootApplication

4. Explain about @EnableAutoConfiguration, @ComponentScan, @Configuration

5. How @ComponentScan internally work

6. When to use @Component and @Service

7. Find the first non-repeated character in string (not specifically mention use core java or using Streams) – which Map we can use in this program.

8. @Transactional and @Qualifer

1.Actuator

2.@Component Scan

3.@Spring Boot

4.Panagram

Find first character in a String that doesn't repeat ( Non Repeating Character).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

find second highest salary in mysql

2. How would you count the frequency of each word in a sentence using a HashMap in Java? Write the code.

3.what is @inject,@resource,@autowired annotations.

4. what is get and load methods

5. what is lazy and early

6. How to implement caching in springboot

**Disney project questions:-**

1. What is anonymous inner class

Anonymous inner class in java without its name of the class, defined and instantiated at the same time. it is defined through compiler. It is used to Override

the method of a class or interface for one time use.

2. write one anonymous inner class

Runnable r = new Runnable() {

@Override

public void run(){

SysOut();

}

};

Thread t = new Thread(r) {

t.start();

}

3. what is functional interface

Functional interface in java 8 is a key enablers of Lambda expression and Method References which made code more concise and expressive.

4. what is lambda expression

lambda expression is used to provide the implementation of Functional interface. it's used to avoid the boilerplate code.

5. what is the use of functional interface

A Functional interface has exactly only one abstract class making it ideal for representing a single behavior/action.

6. how to remove element in arraylist.

(i) ArrayList<Integer> list = new ArrayList<>(Arrays.asList(1, 2, 3, 4, 5));

Iterator<Integer> itr = list.iterator();

while (itr.hasNext()) {

if (itr.next() % 2 == 0) {

itr.remove(); // Removes even numbers

}

}

System.out.println(list); // Output: [1, 3, 5]

(ii) List<Integer> list = new ArrayList<>(Arrays.asList(1, 2, 3, 2, 4));

list.removeIf(n -> n == 2); // Removes all 2s

System.out.println(list); // Output: [1, 3, 4]

7. what are the features are in spring boot

Auto configuration

Stand alone application

Spring boot starters

Build in Testing support - @SpringBootTest, @DataJpaTest, @WebMvcTest

Application Configuration

Spring Boot actuator

Spring Boot DevTools

spring security Integeration

Build in support for kafka, MongoDB, JPA, JDBC etc

Externalized configuration and Profiles - application-{profile}.properties for environment specific configs(dev, test, Pre-prod, prod)

Command Line Runner/Application Runner - Run logic at application startup

8. what is the advantage and disadvantage of microservices

Advantage:

Scalability

Flexibility in Technology Stack

fast development and deployment

Improved fault isolation

Better maintainability

support DevOps and CI/CD

Disadvantages:

Complexity in distributed systems

Difficulties in testing

Increased operational overhead

latency overhead

9. what are the components in load balancer

Listener

Load Balancing Algorithm

Backend pool/Server pool

Health check mechanism

Routing rules/Path-based routing

DNS integration/Virtual IP

Logging and Monitoring

10. what are the ways u can call services class in controller class in spring boot

Constructor Injection

Field Injection(Not recommended for production)

Setter Injection

@autowired on Constructor

11. what is thread pool and types of thread pool

Java Thread pool represents a group of worker threads that are waiting for the job and reused many times.

Fixed Thread Pool

Cached Thread pool

Single Thread Executor

scheduled Thread Pool

Work stealing pool

12. what is rest template

Rest template is used to call the other services

**Morgan Stanly Interview Question:-**

1.Q:What is immutability, Available Immutable Class, How to write custom Immutable Class

2.Q: What is singlton, How to achive it, How to create Custom singlton class

3.Q: Create a custom functional interface that should take three Integer arguments And return an integer value.

Use it as lambada expression in java

4.Q: Employee Object having:

Name, Age, Salary and Designation

Output:

Find list of Designation from given employee list,

Employee age should be b/w 30 to 40 years,

Employee Salary should be greater than 40,000,

Designation list should be in reverse alphabetic order like –

[ VP, Manager, Director]

Using Stream API.

5.Q: There are three tables

Employee - ( EmpId, Name)

State - (StateId, EmpId, Name)

City - (CityId, StateId, Name)

You need to write a join query to get all employee records whether they have records in state or not, whether they have records in city or not.

6. What is JWT, How to implement is , what will be the flow if have to implement

7. What is Spring security, how to implement, how to achieve JWT into it

8.Q: How to secure rest services

9.Q : explain @SpringBootApplication annotation

**Java Coding Questions:-**

1) print helloworld 10 times.

o/p= Hello world 1

Hello World 2 ....

Hello World 10

2) print the 3rd highest from array.

3) reverse a given string. i/p="Capgemini"

4) write a code to find missing number in Integer List

Test cases :

a) 1,2,3,4,5,7,8,9,10

b)2,4,6,10,12,14,16,18,20

c)1,3,5,9,11,13,15,17

5) Write a program to get the occurrence of a specific letter in a given String.

i/p="Hello"

i/p=l

o/p=2

6) Write a java program for palindrome in normal way as well as using Command line arguments.

i/p="MADAM"

o/p MADAM is a Pallendrome

7) Write a program to sort a given list of intergers in ascending order

i/p list={2,4,0,1,56,23,4}

o/p list={0,1,2,4,4,23,56}

8) Write a program to check a given number is prime or not

i/p n=34

9) write a program to a sort a Array of String in ascending oreder alphabetially

i/p String s=["Capgemini","Accenture","TCS","EPAM"]

o/p =[Accenture,Capgemini,EPAM,TCS]

10) Write a program to check a given number is even or odd

i/p n=34

o/p = 34 is even number

11) write a program for fibonacci series;

12) write a program to return 2nd largest number from given array

i/p=[2,3,5,6,1,0]

o/p=6

13) Write a program to implement the defalut method in a Fuctional interface and override it.

14)write a program to Count the number of alphabets in a given sentence

i/p="Capgemini"

o/p = {C-1,a-1,p-1,g-1,e-1,m-1,i-2,n-1}

15)Write a program to store the List of prime numbers in a arrayList and print it.

16)Write a program to create POJO class of Product (Product Id, ProductName) and Store all values in any collection and Sort that values by ProductName in ascending order

17)write a program To count number of words in a given sentence.

i/p "Hello world"

o/p = 2 words

18)Write a program to Reverse a string="Capgemini Training"

19)Write a program count Each vowel frequency in a String(String="capgemini training")

20)Create a class having Fields id,name ,salary and sort it based on id and name using comparator and comparable

21) Given a string sentence, count the number of words in it and display count. The words are separated by the following characters: space (‘ ‘) or new line (‘\n’) or tab (‘\t’) or a combination of these.

22) write program to check whether a number is palindrome or not and

1) n=101

2) n=22022

23).Design a Prime Number between 2 to 30.

24).Swap 2 numbers without using 3rd variable.

25).Design a program to display magic number, accept a number and check it is a magic number or not.

26).Display a L shape \* symbol.

27).Accept a number , find the reverse of it.

28).Write a program to display Pyramid symbol.

29).Declare a TreeSet and store the 2 Employee Objects.

30).Accept a string line and check how many vowels are there.

31).How can you find first non-repeated character in a word.

32).How can you check a string can only have alphabets and not digits.

33).How to get matching characters in a string.

34).How to get non-matching character in a string.

35).How do you prove 2 strings are anagrams.44).Find out the count of occurance of particular character in a string.

36).How to verify if 2 strings are rotation mutually.

37).How to find out first character of string , which is not repeated.

38).How to search a missing numbers in an array, which contains integer from 1 to 100.

39).Determine largest and smallest elements of an array , which is not sorted.

40).Explain the Bubble Sort algorithm.

41).Explain the Merge Sort algorithm.

42).How to remove special character in a string that is lowercase.

43).what is process of deleting matched element from a LinkedList which is not sorted.

44).Find out the Fibonacci series between 1 to 30.

45).Find out factors of given numbers. n=34

46).Write a java program to count number of words in a string using HashMap.

47) given a Char array is there {'a','b','a','c'} Find out not duplicate character i.e. 'a'

48) Write a program to remove the duplicate elements from a list =[2,3,3,3,65,4,3,3,2]

49)Write a Java program to illustrate multiple catch block

50)Write a Java program to get the character at the given index within the String

51)Write a Java program to get the character (Unicode code point) at the given index within the String

52) Write a Java program that checks the letters of the second string are present in the first string. Return true otherwise false.

Input the first string:

Java

Input the second string:

Ja

Check first string contains letters from the second string:

true

53) Write a Java program that removes a specified word from a given text. Return the new string.

("Exercises Practice Solution", " Solution") -> "Exercises Practice"

("Red Green Blue", "Green") -> "Red Blue"

("Java Number Exercises”, “Java”) -> "Number Exercises"

54) Write a Java program to remove a specified character from a given string.

Original string:

abcdefabcdeabcdaaa

remove char from given string : "a"

Second string:

bcdefbcdebcd

55) Write a Java program to sort in ascending and descending order by length of the given array of strings.

Original unsorted colors: [Green, White, Black, Pink, Orange, Blue, Champagne, Indigo, Ivory]

Sorted color (descending order): [Champagne, Orange, Indigo, Green, White, Black, Ivory, Pink, Blue]

Sorted color (ascending order): [Pink, Blue, Green, White, Black, Ivory, Orange, Indigo, Champagne]

56) Write a Java program to create a new string repeating every character twice of a given string.

Sample Output:

The given string is: welcome

The new string is: wweellccoommee

57) Write a Java program to reverse words in a given string.

Sample Input / Output:

The given string is:= Reverse words in a given string

The new string after reversed the words:= string given a in words Reverse

58) Write a Java program to find the maximum occurring character in a string.

Sample Output:

The given string is: test string

Max occurring character in the given string is: t

59) Write a Java program to test if a given string contains the specified sequence of char values.

Sample Output:

Original String: PHP Exercises and Python Exercises

Specified sequence of char values: "and"

output : true

-------------------------------------------------------------------------------------

60) Given an "out" string length 4, such as "<<>>", and a word, return a new string where the word is in the middle of the out string, e.g. "<<word>>". Note: use str.substring(i, j) to extract the String starting at index i and going up to but not including index j.

makeOutWord("<<>>", "Yay") → "<<Yay>>"

makeOutWord("<<>>", "WooHoo") → "<<WooHoo>>"

makeOutWord("[[]]", "word") → "[[word]]"

public String makeOutWord(String out, String word) {

return out.substring(0,2)+word+out.substring(2,4);

}

--------------------------------------------------------------------------------

61) Given a string of even length, return the first half. So the string "WooHoo" yields "Woo".

firstHalf("WooHoo") → "Woo"

firstHalf("HelloThere") → "Hello"

firstHalf("abcdef") → "abc"

Q1) Given a list of integers, find out all the even numbers exist in the list using Stream functions? list=[2,34,678,3]

Q2) Given a list of integers, find out all the numbers starting with 1 using Stream functions?

Q3) How to find duplicate elements in a given integers list in java using Stream functions?

Q4) Given the list of integers, find the first element of the list using Stream functions?

Q5) Given a list of integers, find the total number of elements present in the list using Stream functions?

Q6) Given a list of integers, find the maximum value element present in it using Stream functions?

Q7) Given a String, find the first non-repeated character in it using Stream functions?

Q8) Given a String, find the first repeated character in it using Stream functions?

Q9) Given a list of integers, sort all the values present in it using Stream functions?

Q8) Given a String, find the first repeated character in it using Stream functions?

Q9) Given a list of integers, sort all the values present in it using Stream functions?

Q10) Given a list of integers, sort all the values present in it in descending order using Stream functions?

Q11) Write a Java 8 program to concatenate two Streams?

Q12 ) Write a Java 8 program to find the number of Strings in a list whose length is greater

than 5?

Q13) Explain StringJoiner Class in Java 8? How can we achieve joining multiple Strings

using StringJoiner Class?

Q14)Find the Word/number got repeated by using Stream api

input:2,3,34,,5,5,4,3,3,3

output:2-1,3-4,34-1,5-2,4-1

15) Write a Java Lambda Expression to Create Thread

16) How Lambda Expression and Functional Interfaces are Related?

17)Given a list of Strings .sConvert String to uppercase and Join them with coma (,) using Stream api

List=["USA", "Japan", "France", "Germany", "Italy", "U.K.","Canada"]

18)Given a list of Integer find square of all distinct numbers By using stream api

list=[9, 10, 3, 4, 7, 3, 4]

19)Given a List of Integers find total count, min, max, sum, and the average for numbers by using Stream api

input :list=[2, 3, 5, 7, 11, 13, 17, 19, 23, 29]

output: count=10, sum=129, min=2, average=12.900000, max=29

**Air France client Interview Questions:**

String s="welcome capgemini"; ---- o/p shoud come W & C (Welcome Capgemini)

String s="welcome capgemini"; ---- o/p come,

List<Integer> list = Arrays.asList(1,2,3,1,5,4,2); o/p [1, 2].

only these sceneria they asked

String s = "welcome capgemini";

//o/p Welcome Capgemini

String result= Arrays.stream(s.split(" "))

.map(word->word.substring(0,1).toUpperCase()+

word.substring(1).toLowerCase())

.collect(Collectors.joining(" "));

System.out.println(result);

//o/p come

String result1= Arrays.stream(s.split(" "))

.filter(word->word.contains("come"))

.map(word->word.substring(word.indexOf("come"),

word.indexOf("come")+4))

.findFirst().orElse("Not Found");

System.out.println(result1);

//o/p [1, 2].

List<Integer> list = Arrays.asList(1,2,3,1,5,4,2);

List<Integer> result2 = list.stream()

.collect(Collectors.groupingBy(Function.identity(),

Collectors.counting()))

.entrySet().stream()

.filter(entry->entry.getValue()>1)

.map(Map.Entry::getKey)

.collect(Collectors.toList());

System.out.println(result2);