Technical Evaluation

-project experience

-java 8 features

-functional interface

-Do we have any this functional interface before java 8: runnable and callable

-use of default method

-intermediate and terminal operation

-use of terminal operation

-while iterating on map if you do operations collections intermediate operation then terminal operation again want to use intermediate

- diff between map and flatmap

- stream calls which functional interface interface

- consumer

- optional class

- builder design pattern

- why we are using main

- singleton design pattern

- How do you make our class as a Singleton class?

- solid principles

- face segregation principle and defer dependency injection

- Interface segregation

- can hashmap used in multithreaded environment, if not then which to use

- how concurrent hashmap use

- diff between the sync method and synchronize block which use in multithreaded environment

- What is the difference between yield and joint methods in the multi threaded involvement?

- thread pool concept in the multi threading?

-what is the executor class will use for?

- what are the advantages if you go with the springboot compared with the spring?

- So in the in the springboard applications, right, let's say in our current project, we are using the Tomcat servers.

So later I want to use the any other servers.

So what I need to do?

Do we need any modifications required in the configuration layer?

- different between the rate component and add rate pain

- @Bean @Autowired @component @Controller @restController

-what is the user federate?

-what is the user federate

-Asynchronous and synchronous communications in microservices

-API gateway in microservices

-load balancer

-circuite design pattern

A screenshot of a computer program

AI-generated content may be incorrect.

* Benefit of functional interface
* Diff find first and findAny
* orElse() in stream
* how concurrent hashmap is thread safe
* set vs Linkedset
* what is hash collution and how to overcome this
* Synchronized method vs synchronized block
* Diff eagle and joy method in multithreading
* Solid principle
* Inversion of control in spring boot
* Dependency injection and its type
* What is singleton design pattern
* User federate qualifier
* Diff between ethernet controller and adret bean
* Diff etherate post and etherate put method, toget and etherate load
* \*what is advantage of flat DB like mongodb
* Primary key, foreign key, Controller, Service, Logger Factory, Serialization, streams, input stream and outputstream file, data packets, servlet, dependency injection, inversion of controller, application context

A screenshot of a computer program

AI-generated content may be incorrect.

Stellentise:1) fibonacci series code

2) fibonacci with recursion

3) reverse a string without a stringbuilder

4) prime numbers

5) Difference between hashmap and concurrent hashMap

6) @SpringbootApplication, @Bean

7) @Component vs @ComponentScan

8) ArrayList working

9) Inner join vs self join

10) microservices communication

from microservices, monolithic vs microservice, java 8 features, optional class, asynchronous communication then coding question

Question 1:   
Configure a simple springboot project to accept the of the CAR Description(CAR\_MODEL, CAR\_COLOR, CAR\_NUMBER ).  
Expose an API to accept the Car description and print the object through the service layer and  
give a Success Response once the process is completed.  
  
Question 2:   
Write a function that uses the Stream API to filter a list of integers to find all even numbers and collect them into a new list.

 1.what is polymorphism ? its types .

2.final , finally , finalize ... use

3.finally executes everytime but if we have written some code in finally and we dont want to print statements which are in finally block then what changes we have to do in try catch?

4.what is hashing in collections

5. explain arraylist

6.mandatory spring annotations

7. which are  stereotype annoations  in spring

8.diff between .equals and ==

9.what are the types of requests in controller

10.dimond problem - if we define 2 default methods in 2 interfaces with same name then how we will override them as both have same name ?

11. features of stream Api

12.explain functional interface

13multiple inheritance is possible in java? and if not then why ? what is the reason ? and how we will acheive it ?

14.what programming language you used in last year project

15. string s1 = "hello"

       string   s2 = "hello"

 string s3 = new string("hello")

       string   s4 = new string("hello")

what will output of following?

s1==s2

s1==s3

s3.equals(s4)

s3==s4

16. what us singleton class

palindrome code

string reverse without using stringbuilder

anaghram code

prime number code

fabonacci code