**JAVA -**

1. Can we override static methods in java?

Ans: NO, we can't override static methods since method overriding relies on dynamic binding at runtime, but static methods are bonded at compile time with static binding. As a result, we are unable to override static methods.

1. Why is string Immutable in java, How to create an immutable class in java ?

Ans: Immutable class in java means that once an object is created, we cannot change its content. In Java, all the wrapper classes (like Integer, Boolean, Byte, Short) and String class is immutable. We can create our own immutable class as well.

* The class must be declared as final so that child classes can’t be created.
* Data members in the class must be declared private so that direct access is not allowed.
* Data members in the class must be declared as final so that we can’t change the value of it after object creation.
* A parameterized constructor should initialize all the fields performing a deep copy so that data members can’t be modified with an object reference.
* Deep Copy of objects should be performed in the getter methods to return a copy rather than returning the actual object reference)
* There should be no setters or in simpler terms, there should be no option to change the value of the instance variable.

// An immutable class

**final** **class** Student {

    // Member attributes of final class

**private** **final** String name;

**private** **final** **int** regNo;

**private** **final** Map<String, String> metadata;

    // Constructor of immutable class

    // Parameterized constructor

**public** Student(String name, **int** regNo,

                   Map<String, String> metadata) {

         // This keyword refers to current instance itself

**this**.name = name;

**this**.regNo = regNo;

        // Creating Map object with reference to HashMap

        // Declaring object of string type

        Map<String, String> tempMap = **new** HashMap<>();

        // Iterating using for-each loop

**for** (Map.Entry<String, String> entry : metadata.entrySet()) {

            tempMap.put(entry.getKey(), entry.getValue());

        }

**this**.metadata = tempMap;

    }

    // Method 1

**public** String getName() { **return** name; }

     // Method 2

**public** **int** getRegNo() { **return** regNo; }

     // Note that there should not be any setters

     // Method 3

    // User -defined type

    // To get meta data

**public** Map<String, String> getMetadata() {

         // Creating Map with HashMap reference

        Map<String, String> tempMap = **new** HashMap<>();

**for** (Map.Entry<String, String> entry :

**this**.metadata.entrySet()) {

            tempMap.put(entry.getKey(), entry.getValue());

        }

**return** tempMap;

    }

}

1. How can we create custom exception in java ?

Ans: Extends Exception

1. What is the difference between array list and LinkedList ?
2. Difference in hash table and HashMap ?
3. Why do we need concurrent HashMap ?

Ans: ConcurrentHashMap is a thread-safe implementation of the Map interface in Java, which means multiple threads can access it simultaneously without any synchronization issues. It’s part of the java.util.concurrent package and was introduced in Java 5 as a scalable alternative to the traditional HashMap class.

One of the key features of the ConcurrentHashMap is that it provides fine-grained locking, meaning that it locks only the portion of the map being modified, rather than the entire map.

**Key points of ConcurrentHashMap:**

* The underlined data structure for ConcurrentHashMap is [Hashtable](https://www.geeksforgeeks.org/hashtable-in-java/).
* ConcurrentHashMap class is thread-safe i.e. multiple threads can operate on a single object without any complications.
* At a time any number of threads are applicable for a read operation without locking the ConcurrentHashMap object which is not there in HashMap.
* In ConcurrentHashMap, the Object is divided into a number of segments according to the concurrency level.
* The default concurrency-level of ConcurrentHashMap is 16.
* In ConcurrentHashMap, at a time any number of threads can perform retrieval operation but for update in the object, the thread must lock the particular segment in which the thread wants to operate. This type of locking mechanism is known as Segment locking or bucket locking. Hence at a time, 16 update operations can be performed by threads.
* Inserting null objects is not possible in ConcurrentHashMap as a key or value.

1. **Explain internal working of HashMap**
2. **Difference between @RestController and @Controller**

Ans:

@RestController is nothing but the shortcut to use both @Controller and @ResponseBody annotation together. In other words, @Controller is used to controller which can accept and return HTML while @RestController annotation can be used to return JSON response.

1. **Why is method overloading not possible by changing the return type in java ?**

Because you are not required to capture the return value of a method in Java, in which case the compiler can not decide which overload to use.

1. What is the use of optional class in java, Can you please explain with any use-case
2. Which, according to you, are the most significant features of Java 8?
3. What was wrong with the old date and time and How will you get the current date and time using Java 8?
4. What is the need of default and static methods in java 8?
5. How to implement
6. What is functional interface and How can you create a Functional Interface?
7. What is a Stream API? Why do we require the Stream API?
8. What is the difference between limit and skip?

Stream limit() method takes one (long N) as an argument and returns a stream of size no more than N. limit() can be quite expensive on ordered parallel pipelines, if the value of N is large, because limit(N) is constrained to return the first N elements in the encounter order and not just any n elements.

The skip(long N) is a method of **java.util.stream.Stream** object. This method takes one long (N) as an argument and returns a stream after removing first N elements. skip() can be quite expensive on ordered parallel pipelines, if the value of N is large, because skip(N) is constrained to skip the first N elements in the encounter order and not just any n elements.

1. Difference Between Map and flat Map Stream Operation
2. What are the new features introduced in Java8
3. What are functional interfaces and why they introduced in java.
4. How to make immutable classes in java.
5. If you have a user defined class, and stored in hashmap using object as key. How to approach that (equals and hashcode overriding). And if you change one of the state variable for the object then you won’t be able to get the object, how will you approach it (making class immutable – solution).
6. What are Intermediate and terminal operations in Stream API and also examples of them.

**PROGRAMS -**

1. How to reverse string in java?
2. Write a Java 8 program to remove the duplicate elements from the list?
3. Program to print non-repeating character from integer array
4. Given list of integers, sort all values in descending order using stream
5. Program to count occurrence of words in a string using java 8
6. Program to count occurrence of each character in a string
7. Java program - Remove duplicate values(They will share one link)

Input - 1,2,3,4,1,2,3,4,5

Output – 5

1. Roman to Integer
2. Array contains multiple duplicate elements and a single unique element. Find and return that unique element
3. Find unique number from the list(Input is int array)
4. There is a list of number from 1 to 20 and there are three threads which will print the numbers

Thread-1 will print numbers which are divisible by 3

Thread-2 will print numbers which are divisible by 5

Thread-3 will print remaining numbers in the list

How will you approach this and what if numbers to be printed sequentially as well from 1 to 20.

1. There is an array of numbers and given a variable K (let say has value 4)

Need to find pair of numbers in the list such that when you divide them the remainder is equal to K

Or

Need to find pair of numbers in the list such the when you sum then you will get the sum as K.

1. Need to complete the code for the below ‘check’ function:

public boolean check(String target, String[] arr) {

              // To-Do

}

check("abcdef", ["ab", "cd", "abc", "de", "def", "ef"]);  // true

check("xabcdef", ["ab", "cd", "abc", "de", "def", "ef"]); // false

given array of String and a target string need to find out if the strings in the given array concatenates and form the target string. If form then return true else false.

1. Remove duplicate characters in the given string. Below are the constraints needs to be followed:
   1. Can’t use any Collection like Set, Map for duplicate character checking
   2. Time complexity should be O(n)
   3. No use of String comparison method
   4. Not take each char and compare it with string or other character
   5. Return value should be string without duplicate characters.

**SPRING BOOT and SPRING -**

1. Explain DI and IOC
2. Difference in constructor and setter injection.
3. Explain annotations - @component, @autowired, @configuration, @enableautoconfiguration
4. What is difference between PUT and POST http methods
5. Difference in PUT and PATCH http methods
6. Which is idempotent http method?
7. What is Spring Boot and what are the benefits?
8. What is rest Template? Why we need it ?
9. Can you please explain what is the use of @SpringBootApplication Annotation
10. What is the use of starter dependencies in pom.xml ?
11. What is Swagger in Spring Boot?
12. What is the use of Actuators?
13. How can we handle global exception in our application?
14. How do you build a Spring boot service from scratch
15. What DB you have used for your micro service and how to configure/use that in spring boot

MYSQL

* 1. Nothing needs to be done for Entity Manager Factory in spring boot project
  2. For Datasource – 1.Add mysql dependency in pom.xml 2. Add datasource related properties either in @configuration class or in properties file.

1. How conflict in dependent bean creation handled which annotation to use (Qualifier and Primary annotation)
2. Difference between Spring and Spring Boot
3. Features of Spring boot
4. Difference between Spring Boot and Spring MVC (asking about dispatcher servlet flow)
5. How to do transaction management (ACID transactions) in spring boot (jdbcTemplate)
6. Which authentication mechanism you used in your previous project.
7. What are @Autowired and @Resource annotations.

**MICROSERVICES –**

1. What is microservice, does it have any advantage over monolithic architecture ?
2. Explain topics ->
   1. Eureka Server
   2. Circuit Breaker - Resilience 4J and different circuit breaker mechanisms.
   3. Api Gateway - Spring cloud gateway
   4. Kafka
3. What is the use of API gateway?
4. Why do we need Eureka server/ what problems it helps in resolving?
5. What is the reason for implementing circuit breaker?

**General:**

1. Introduction
2. Current project details with roles and responsibilities include technology
3. What about agile methodology include sprint details
4. Did you work on kanban and scrum methodology? If yes which one is best for you?
5. Did you face any difficult in your project and how to solve.

**Miscellaneous:**

1. Have you worked on rabbit or kafka messaging in your current or previous projects?
2. Have you worked on any containers particularly kubernates?
3. How to deploy your projects - We need to explain from coding to cloud.
4. Have you worked on Kafka, Grafana, Kibana, Docker, Kubernates, Radish, AWS and cucumber.
5. How to see logs in Kubernetes

Ans: kubectl logs [container name]

1. Explain dockerFile

Ans: A dockerFile is a file that contains commands to build a docker image.

FROM, ENTRYPOINT, CMD etc.

1. How to pull images from Docker

Ans: docker pull [image name]

1. What are the git commands you are familiar with
2. What is your CI/CD experience? Explain the flow of CI/CD in you last project
3. What are the design pattern you have used and why?
4. How do you write code for better maintainability (I have answered on the basis of SOLID principle)
5. What test framework you have used
6. Why did you use mockito
7. What will be your approach in perspective of testing while wring test cases?
8. What type of test case you have executed (unit test, integration test, load test )

java8 features that are used by u most  
Services communications  
coding methodology  
design patterns used  
steps to follow to deploy code in microservices  
autowiring  
why JPA instead of JDBC  
springboot annotations used   
regex  
design patterns

1. There is an array of integers positive and negative by adding two integer check the another integer is present in array or not?(2,7,8,9,-2,4,-6,3,7,9) x=16 like 7+9=16

Given an array A[] of n numbers and another number x, the task is to check whether or not there exist two elements in A[] whose sum is exactly x.

1. reverse the index like 2 to 9 and 9 to 2
2. difference between search and binary search which is used for better performance?
3. in above array implement binary search?
4. How to perform binary search
5. what annotation used in springboot?
6. Rest template and rest client ?How to implement? any other than rest template what we can use?
7. @component and @bean
8. @postmapping
9. what is PostConstruct?

Java

Explain your previous project structure & security used.  
Explain the steps in creating API in spring boot.  
Explain spring boot security.  
How token is validated in springboot.  
Sort the userlist in reverse order using streams API - Write the program  
How do you version API in springboot or how do we migrate from v1 to v2 API without affecting the existing version.  
A user can have multiple roles, a role can have multiple permission. can your write the class/entity structure? - Write the class details

Angular

Explain routing in Angular, How will you navigate between routes?  
How can we create a common component in angular and how data flow happens in common component?

ViewChild in Angular

MongoDb

How do we join two collections?  
Which operation is used to convert array to object in a collection column?  
How can we restrict the no of columns/ select only 4 columns in Mongodb?  
While using Lookup operation, what will be the structure/type of the joined collection or How it look like after joining two collections?

[6:06 PM] Joy, Rosemary (Cognizant)

this was Bergin interview qtnc

[6:06 PM] Joy, Rosemary (Cognizant)

who got selected

[6:07 PM] Joy, Rosemary (Cognizant)

its another client ok

[6:07 PM] Joy, Rosemary (Cognizant)

i mean not client of satyam

***Programming Questions:***

1. There is a list of number from 1 to 20 and there are three threads which will print the numbers

Thread-1 will print numbers which are divisible by 3

Thread-2 will print numbers which are divisible by 5

Thread-3 will print remaining numbers in the list

How will you approach this and what if numbers to be printed sequentially as well from 1 to 20.

1. There is an array of numbers and given a variable K (let say has value 4)

Need to find pair of numbers in the list such that when you divide them the remainder is equal to K

Or

Need to find pair of numbers in the list such the when you sum then you will get the sum as K.

1. Need to complete the code for the below ‘check’ function:

public boolean check(String target, String[] arr) {

              // To-Do

}

check("abcdef", ["ab", "cd", "abc", "de", "def", "ef"]);  // true

check("xabcdef", ["ab", "cd", "abc", "de", "def", "ef"]); // false

given array of String and a target string need to find out if the strings in the given array concatenates and form the target string. If form then return true else false.

1. Remove duplicate characters in the given string. Below are the constraints needs to be followed:
   1. Can’t use any Collection like Set, Map for duplicate character checking
   2. Time complexity should be O(n)
   3. No use of String comparison method
   4. Not take each char and compare it with string or other character
   5. Return value should be string without duplicate characters.

**Theory Questions:**

* + - 1. What are the new features introduced in Java8
      2. What are functional interfaces and why they introduced in java.
      3. How to make immutable classes in java.
      4. If you have a user defined class, and stored in hashmap using object as key. How to approach that (equals and hashcode overriding). And if you change one of the state variable for the object then you won’t be able to get the object, how will you approach it (making class immutable – solution).
      5. How conflict in dependent bean creation handled which annotation to use (Qualifier and Primary annotation)
      6. Difference between Spring and Spring Boot
      7. Features of Spring boot
      8. Difference between Spring Boot and Spring MVC (asking about dispatcher servlet flow)
      9. How to do transaction management (ACID transactions) in spring boot (jdbcTemplate)
      10. What are Intermediate and terminal operations in Stream API and also examples of them.
      11. Which authentication mechanism you used in your previous project.
      12. What are @Autowired and @Resource annotations.