**Amelia Interview questions**

1. Explain oops concept -> method overloading

class Check

{

public void m1(Integer x/Integer x/object obj)

{

syso (“this is for on 1 arg”)

}

public void m1( String str/string buffer)

{

syso (“this is for 2 arg”)

}

public static void main(String args[])

{

Check obj = new Check()

obj.m1(null)

1. Collections -> arraylist and linklist diff
2. Set and map diff
3. List and set diff
4. How to iterrate on map
5. for(Map.Entry<x,y> maps :map.entrySet())

{

If(maps.getValue()==1){

Syso(maps.getKey());

Map<String,String> map = new HashMap<>();

//map.put("ashish","gupta");

1. How many null value can be in hashmap/ why null and how to call it.
2. Exception
3. How to handle pointer exception and what kind of exception is that
4. What is singleton and its use.
5. Write a singleton class
6. Value of int

**Interview questions (TCS) -> 1**

1. What is the annotation used for caching in spring.
2. What is the diff btw abstract and interface/ how to declare or syntax for interface.
3. How to remove duplicate from set
4. Pom.xml, how to remove dependency.
5. Diff btw static/static variable/static block.
6. What is synchronization.
7. How to deploy in production.
8. What is spring actuator.
9. Diff btw final and finally.
10. What is the java version code used.
11. For spring security points in consideration.
12. Static / non-static ref call

**Interview 2**

1. What is JPA
2. How to read json
3. What and how to use database jdbc
4. Query in sql to insert
5. Code using lambda expression for adding 2 no.
6. How can we update value in hashmap.
7. Springboot application what is that.
8. Let me know about collection.
9. How we handle exception in spring.
10. What is functional interface

**Interview 3rd**

1. What is repository(JPA)
2. What is service class
3. What is obj class
4. Diff btw array and arraylist
5. Diff btw string buffer and builder
6. Diff btw controller and restcontroller
7. Stream filter/interface
8. Rest url
9. Find 2nd highest no.
10. Ansible /CI integration/kubernetes/AWS
11. How splunk server works
12. Tech stack used
13. How to convert Array to set

**Seimens Interview - 1**

1. Java 8 -> static in interface
2. Stream api
3. Use of reduce
4. What is spring
5. Configuration properties
6. What is IOC/DI
7. Spring security
8. Find max sub array
9. What is k+1 in hibernate
10. What is optional

**Seimens Interview - 2**

1. Project/role
2. Java 8 memory management
3. Lambda expression
4. Why optional why not exception
5. Types of optional
6. Static method
7. Interface
8. Spring boot
9. Microservices
10. Component scan
11. Dependency injection
12. Spring/spring boot
13. Actuator
14. Java 8
15. Remove duplicate in map -> equals and hashcode method

**Deutche bank**

1. Details about project/version
2. What is try with resource
3. Diff btw static and final
4. What is immutable class in java
5. Throw and throws keyword
6. Method overloading/overriding
7. What is scp -> why dedicated memory is needed.
8. How we handle exception
9. Diff btw controller and rest controller
10. What is docker
11. Multi-threading
12. How to prevent deadlock
13. What is spring bean/scope of bean
14. What is methodoloy used/how you fix any issues.
15. Devops/git
16. How you do testing.
17. How you store secrets.
18. Sql query to find 2nd max sal
19. What is stream api
20. How to write immutable class in java
21. O/p question

public class Test {

public static void main(String[] args) {

String s1 = "abc";

StringBuffer s2 = new StringBuffer(s1); // hashcode for stringbuffer to toString

System.out.println(s1.equals(s2));

}

}

public class Test {

public static void main(String[] args) {

String str = null;

System.out.println(str.valueOf(10)); //static method

}

}

**Infosys Interview questions**

1. Tel me about your project
2. What is software development cycle
3. Methodology
4. Version control
5. Git/ci/jenkins
6. Oops concept -> abstraction
7. JVM
8. Java 8
9. Multi threading
10. Deadlock
11. Exception typs
12. What happens if in a class we don’t give main method
13. Stream api
14. Diff between == and .equals
15. Functional interface
16. Lambda expression
17. Linked list
18. Collection
19. Transient keyword
20. Coding question -> fibonacci series and check if a number amstrong or not

**Qburst Interview question - round 1**

1. What is indexing and normalise session
2. Functional interface
3. **Find a** missing number
4. Find second max salary using sql query

**Round 2**

1. Endpoints -> for logs to interact with db
2. Entity class parameters for logs
3. For storing logs which storage used
4. How to fetch logs
5. How logs stored in repository
6. Authentication /authorization
7. Service end points
8. Mysql
9. Kibana vs your project
10. Types of Databases
11. Type of storage accounts
12. Database failover/database performance
13. Indexing

**Coforge Interview questions**

1. Tell me about your project
2. Fuctional Interface
3. Concurrent hashmap
4. Use of hashcode and equals
5. What is failsave and failfast
6. Diff btw comparable and comparator
7. Diff btw terminal and intermediate
8. What is multi threading - how to make thread safety
9. Diff between synchronous and locking
10. What is profile annotation
11. What is bean/qualifier
12. Transaction
13. Exception
14. Diff btw put and patch
15. Spring security --> how to make the session active if the current user is using
16. How to communicate from ms1 to ms2
17. Design pattern
18. **xceed interview**
19. How migration was done for project.
20. Why spring boot
21. Stream api
22. Hashset/put/get
23. Aggregation/compostion
24. Each-A and hash-A
25. Serilization
26. Synchronized
27. String immutable
28. How to write immutable class in java
29. Singleton
30. How snchronized block works
31. What is protype
32. Function/procedure
33. Join

**EY interview question**

1. Frontend tool used with version
2. Java 8 version
3. Calling method /method constructor/ref
4. Function interface
5. If more than one abstract method what result
6. Spring boot version
7. How to bypass some user with no authentication
8. Diff btw jpa and crud repository
9. How we manage transaction issue in spring boot
10. Diff btw stream and parallel stream
11. Junit testing
12. Post mapping and put to update
13. Request body to get the payload
14. How to save/token in cookies
15. Microservices

**Motherson Interview question**

1. Tell me about your project
2. Team size
3. Functional Interface
4. Lambda expression
5. Java 8 features
6. Create a main method to implement this

@FunctionalInterface

public interface LengthInterface {

int getLength(String s);

}

**Codevyasa Inteview questions**

1. Find occurrence of each character of string.
2. @Asynchronous
3. What is singleton and use of that.
4. Diff btw controller and rest controller
5. Sleep

**Interview -> datamatics**

1. Java 8 code  
   ("AA", "BB", "AA", "CC")

{CC=1, BB=1, AA=2}

Map<String,Integer> map = Arrays.stream(str).collect(Collectors.groupingBy(Function.identity(),collectors.counting());

syso(map);

1. Object class types
2. Implementation of hashmap/hashset
3. Diff btw final/static
4. What is path variable/request param/query param
5. Create a endpoint for path variable and request param
6. Sql query

Emp -> id, name,salary, dept(FK)

1, ashish,50000,1

2,abc,10000,2

3,xyz,20000,1

4,aha,35000,3

5,ghg,40000,2

dept -> id, name

1, a

2,b

3,c

empname, deptname, max of salary for each dept

Sol -> select d.name,max(e.salary) from emp E join dept D on e.dept =d.id groupby d.name sort by e.salary desc limit 5;  
  
8. component scan/bean

1. spring security

**UKG Interview**

1. Tell me about your project
2. Java 8 features
3. How to handle exception
4. Testing
5. How to write singleton class
6. How to make thread synchronizes
7. Diff btw throw and throws keyword
8. Array and linklist
9. Override -> hashcode/equals
10. Comparable/comparator
11. Binary search
12. Join and types
13. Spring security
14. Bean/spring boot annotations

**Contata Interview questions**

1. Details of project
2. Sql query

Problem Solving - Query Formation

-Table Schema - Student <RollNo(PK), Name, Marks>

-Assume - Min 100 Records are available

-Form a query to fetch the name of a STUDENT that has 5th highest marks in the table

1. Inner join and outer join
2. Thread class -> using thread class and runnable interface
3. Singleton and factory design pattern
4. Create a singleton class
5. Code to write palindome
6. Bean scope type
7. What is stringbuilder
8. Autoboxing and autounboxing
9. Super use
10. Super and this can be used together.

**EPIQ Interview question**

1. Tell me abt ur project
2. Frontent tech stack
3. Opps concept
4. Pojo
5. Interface
6. Method overloading/overriding
7. Diff btw array and link list
8. Hashmap
9. == and .equals
10. String buffer/Builder
11. Synchroized
12. SCP
13. Why string is immutable
14. Reverse a string
15. Count duplicate character in a string

**Delloite Interview**

1. Introduce urself
2. Explain project
3. Java 8
4. Implemenation of hashmap
5. Load factor
6. Failsave and failfast
7. Collection
8. Concurrent hashmap ---> failsave
9. Stream api
10. Optional
11. Off method in optional
12. Java 8 question -> Optional<List<Integer>> abc = list.stream().filter(e->e.getAge()>60).collect(Collectors.toList());
13. Optional.ofNullable(abc)

List<Integer> list = Arrays.asList(1,2,3,6,8,-1);

Optional.ofNullable(list);

isPresent/getIfPresent/orElse/orElseThrow

1. DI -> dependency injection
2. IOC
3. Spring bean
4. Bean scope and typs
5. Bean lifecycle

post construct -> if we want to initialize any bean/variable

pre destroy -> if pwd stored in bean we want to destroy post work we use destroy.

1. 1 to many and many to 1
2. Spring security
3. Regex

**Cardinal Health Interview questions**

1. Introduce the project
2. oath2
3. Authentication and authorization
4. Spring security
5. Token generation
6. Garbage collection
7. Unit testing
8. How we do code quality check
9. Jenkins types

**Ness Interview**

1. Tell me about your project
2. Roles and responsibility
3. Functional interface -> use and types
4. Rulable and callable
5. Predicate/function
6. Optional class
7. Stream api
8. Map and hasmap
9. How to use java 8
10. Serialization and dserialization
11. Trasient
12. Controller and restcontroller
13. Put and post
14. Put and patch
15. Spring Actuator
16. Post mapping /request body
17. Types of annotation
18. How we create jar files
19. Docker file
20. Convert from string to long without using inbuild features like wrapper class.

**Ameripise Interview**

1. Tell me about your project
2. Solid principles
3. Polymorphish
4. Parents also static and child also static what will happen
5. List and set
6. List and treeset
7. Check and uncheck exception
8. Try and catch
9. How to create a immutable class --> how to make student class as immutable
10. Spring boot flow
11. Microservice comunicate

**Synechron Interview**

1. Project
2. Reverse a int[]
3. List and array list
4. Java 8 /lambda exprssion
5. Functional interface

Coding questions

1. I/P -> I LOVE MY COUNTRY

O/P -> COUNTRY MY LOVE I

1. Count occurenec of each duplicate character/all char
2. I/p -> my name is ashish

O/p -> ym eman si shihsa

1. Create a list and add number
2. Use prd to check if prime no. Or not
3. Write a Java program to find the missing number in an array of consecutive integers from 1 to N, where one number is missing.
4. Create a index
5. Find 2nd max sal
6. Create a post end point
7. Create a entity class
8. Reverse a int[] using 1 loop
9. Create a custom exception
10. Java 8 code to find number starts with 1
11. Code to get age>45 and for that increase the sal by 40%

list.stream().filter(e->e.getAge()>=45).map(e->e.setSal(e.getSalary()+(e.getSalary()\*0.4)).collect(Collectors.toList());

1. Find the employe with max salary

Optional<Employee> emp = list.stream().collect(Collectors.maxBy(Compartor.comparing(Employee::getSalary));

sop(emp.get());

1. student class -> roll no./name/marks

Get name whose marks>60

List<String> list = list.stream().filter(emp->emp.getMarks()>60).map(emp->emp.getName()).collect(Collectors.toList());

1. Create a bean with rest template (get for object)
2. Create a optional

List<Integer> list = Arrays.asList(1,2,3,6,8,-1);

Optional.ofNullable(list);

1. Create a functional interface and use a main method