**Core Java - Interview Inputs:**

* Memory management
* OOPS
* Design the immutable class with considering edge cases.
* Design the singleton class with coinciding all the edge cases.
* Collections API, Hashcode and equals concept w.r.t hash map, custom linked hashmap.
* Multi-threading concept and concurrent API.
* Problem solving skills.
* Advantages of Spring framework over other traditional frameworks.
* Different scopes in Spring explain them and their applications.
* Given a 4 digit no find the next digit formed from given 4 digits.
* Given two arrays find the common numbers in these arrays.
* Countdownlatch vs cyclic barrier.
* What is the deadlock on how to resolve it.
* Clone a doubly linked list pointing to arbitrary no.
* If a Web application is not responding what steps you will take.
* How garbage collector works?
* Use of static class and can we create private constructor of static class?
* Diff between Wrapper class and primitives?
* Scenario based questions on wrapper classes?
* Synchronization related scenario based questions?
* Why is string immutable how can we ensure immutable with our custom class Serialization, constructor chaining, cloning.
* Annotations for Bean Declaration.
* How garbage collection internally work.
* How many heap & stack memory in one JVM
* Where static variable store in memory that area will be garbage collected.
* Function overriding
* Immutability, Steps to implement immutable Class
* Deep and shallow cloning
* Sleep vs wait() method. What will happen if wait() is called from message method [Class A { void message(){ wait();} }]
* Working of HashMap. Java contract of equal() and hashCode()?
* What will be complexity of get(key)
* If hascode() returns 1 and equal() true
* Use case of tree map and linked hashmap
* How will you order a collection of Employee object in {name, age} orders, Scenario based?
* Diff between comparable and comparator.
* How to avoid serialization of child class
* How serialized k attributes out of N attribute
* Thread safe Singleton class design.
* How to avoid duplication through reflection, serializable
* Wait and sleep method difference?
* Start same thread twice ?
* Thread t1 and t2 called two diff method static synchronized m1 & synchronized m2. What will be the oder of execution of this methods
* How to start N threads in sequence
* How to print counting using two thread where t1 always print odd number and t2 always print even number
* Hashcode and equals methods not overridden- how the put and get method works
* Hashcode overridden but returning constant number and equals not overridden -how the put and get method works.
* Hashmap resizing in case of multithreading·
* Stack and heap memory.
* Types of generation garbage collection.
* Clustered and non-clustered index.
* Union vs union all

**Data Structure & Algorithms :**

* First question was for given array you need to find the number which has maximum occurance and how many times it is been occurred. Also if the second array is given then you need to find that if second array is subset of first array.

And find the intersection of that array

* Second question was for design the architecture for given problem

Problem was suppose we have shared cab service and fare will be shared between person for the distance in which they will share the cab

Example person p1 boarded at a station and his destination is d station. Person p2 boarded at b station and his destination is e station. Fare should be calculated in such a way that a to b Person p1 is travelled alone from a to b so he need to pay the entire fare for that distance but from b to d he has shared the cab with person p2 so the fare will be shared between two persons till station d

Person and station can be arranged in multiple ways

* Third question was related to first question that delete the near first and next to element from that array of maximum occurred number
* Find the number with highest frequency in a very big list.
* Delete the numbers adjacent to the highest frequency number.
* A cab takes n passengers in pool from a to b. at any point, passengers can alight or embark. For distance a to b, fare is equally divided between passengers. Design the solution (classes, business logic...)
* Design an algo to transverse N metres using 1,2,3,4 and 5 metres-
* Segregate an array with elements 0,1,2 but all the elements are duplicate. With time complexity O(N) and Space complexity O(1)
* Delete given node from singly list If head node is not given. With time complexity O(1)

**Concepts :**

* Immutable object, enums, exception handling, string API,String index of and chart function,Custom exception creation,oops,static/dynamic binding, constructor definition,encapsulation, polymorphism
* Given a scenario to implement a system which will have multiple readers and only one writer
* Executor framework or concurrent thread execution, class level lock, automatic integer, future, callable, count down latch. cycle barrier
* Hashmap resizing in case of multithreading
* Hashsets internal , Reverse string, finding the output array with O[N] iteration,char/word/line count.
* Hashcode and equals methods not overridden- how the put and get method works
* Hashcode overridden but returning constant number and equals not overridden -how the put and get method works
* Serialization and how to make singleton serialization safe
* Lambdas
* Databases - clustered and non-clustered index, union vs union all

| **Assessment Area** | **Interview Questions asked** |
| --- | --- |
| Memory management | 1. How garbage collection internally works.  2. Diff b/w perm gem & meta space.  3. In one JVM how may no. of heap & stack memory are possible. |
| OOPs | 1. Diff b/w abstract class & Interface.  2. Improvement inside interface in java 8.  3. Use of abstract class in java 8.  4. Method overriding problem related to exception. |
| Immutability | 1. What is immutable class.  2. How to design immutable class.  3. How to handle mutable scenario.  4. Diff in deep & shallow copy. |
| Singleton | 1. What is a singleton class.  2. How to make a thread safe singleton.  3. Use of volatile keywords.  4. How to prevent singleton object in serialisation & deserialisation.  5. How to prevent singleton object in reflections. |
| Collections | 1. How hashmap internally works.  2. In a custom class only equals method overridden gethashcode is not overridden.  In that case if we put this class object as a key in the hashmap, so hashmap will work or not.  If not then why?  3. Diff b/w tree set & linked hash set.  4. Create a custom linked hashmap with get, put, delete functionality with O(1) complexity.  5. Diff b/w concurrent hashmap & hash table.  6. Why concurrent hashmap not allowed null key or value. |
| Multi-threading | 1. Possible ways to create thread in java.  2. Use of callable interface.  3. Diff b/w wait and sleep method.  4. Print counting from 1 to 100 in sequence using two thread t1 & t2. Where T1 print odd numbers & T2 prints even numbers.  5. How thread pool internally works.  6. How blocking queue internally work. |
| RDBMS | 1. Diff b/w primary & unique key.  2. Diff b/w clustered & non clustered Indexing. |
| Spring | Good experience |
| Data Structure / Problem solving | Segregate 0, 1 & 2 in sorted order of an unsorted array elements. which contains 0,1,2 withrepetition. time complexity O(n) space complexity O(1). |
| Communication | Should be good |