Java :

- Steps to implement immutable Class
- Sleep vs wait() method. What will happen if wait() is called from message method [Class A { void message(){ wait();} }]
- What will be complexity of get(key) if hascode() returns 1 and equal() is implemented properly (using any IDE generator) ?
- Which method should be used if One has to use List.contain(Employee obj) and why?
- Diff between comparable and comparator
- How will you order a collection of Employee object in {name, age} orders?

Interface

- What is marker Interface and why do we use? (Searilizable interface. Cloneable interface.)
- Implementation steps for custom marker interface.
- Functional interface – why do we have functional interfaces and what they help achieve
- the difference between Abstract class and Interface.

Serialization

- What is Serialization? How serialization works. How can one do custom serialization?
- How to avoid serialization of child class
- Serialization, Serialization when non serialize reference is present in the class
- Prevention using deserialization and reflection.
- Implement Singleton Object. How to avoid duplication through reflection, serializable
- transient keyword

****

- Immutability – Advantages and Usage
- Immutability (String class functioning, string pool)
- Immutability scenario-based question

****

- Input from multiple system and validation of the input design

****

- Core Java concepts (access specifiers, abstract classes, interfaces, method overriding)
- Design patterns
- Spring framework
- Spring basics – DI and IoC
- Spring - DI, various containers
- Annotations in Spring MVC

****

- Aggregation and composition example

****

- Call by value / Call by reference

****

Data Structure

HashMap

- Collections in java like HashMap, HashSet
- HashMap, HashSet internals Working
- Concurrent HashMap functionality [Concurrent HashMap with object- Hashcode overrides]
- Working of HashMap. Java contract of equal() and hashCode()?
- Implement your own HashMap
- HashMap collision -Optimize your HashMap in case of collisions
- Hashcode equals contract.

- Concurrent Hash Map Working, equals, hash code
- HashTable vs Concurrent Hash Map
- Custom key in hash map
- String in HashMap as a key
- Stringpool and object creations of strings
- Linked List (How to find the 3rd last element of linked list in one pass)
- Identify a duplicate in an array

- Tree data structure collections in java like TreeMap, TreeSet.
- Level Order Printing of Binary Tree (pseudo code.)
- Time complexity of operations on stack and queue data structures
- Implementation of queue data structure using array
- LinkedListed vs ArrayList, Internal working, which one to prefer, time complexity for same
- difference between ArrayBlockingQueue and LinkedBlockingQueue.
- how to find a circular loop – write node – write your own linked list find a circular reference between the node.
- logic to traverselinked list once and find middle node
- Write a code to find loop in a linkedlist
- Reverse a linked list.
- Implementation of the PriorityQueue
- Collections (Queues, List, Set)
- Stack vs queue

Multithreading

- Volatile Keywords and Its uses
- AtomicInteger & its internal implementation
- synchronization mechanism
- What is difference between intrinsic synchronization and explicit locking using Lock
- Reentrant nature of the intrinsic locks.
- Threads/Multithreading (Atomic classes, Countdown latch, Volatile keyword, Executors)
- Thread life cycle
- Locks / Deadlock
- Race condition
- class & object level locks
- Can an array be declared as volatile variable. If yes then what are its implication
- Cyclic barrier vs countDownLatch

Polymorphism Scenario

- Parent and Child class having same method signature but different return types
- Parent and Child class having same method signature and same return type, but this time parent throws FileNotFound Exception and Child throws IOException
- Runtime Polymorphism
- Compile time polymorphism

- Overloading and Primitive/ Wrapper Class combination
- basic override principle for exceptions
- Explained run time polymorphism [method hiding (overriding of static methods), covariant method overriding

Java 8 concepts

- New features in JAVA 8
- Java 8 basics, streams
- Filter a Collection using lambda expressions.
- default implementation and functional interface feature from Java 8

Garbage Collections tuning fundamentals

- Different Garbage collection tuning parameters and scenarios

- Memory Model and GC
- Basics of Memory areas and collections
- JVM, Explain Classloaders

- Inheritance in Generics
- Explain Inheritance, Constructor chaining questions

****

- Upcasting, Downcasting.
- Overriding rules.
- Overloading & Overriding (Autoboxing and widening)
- Overriding. Reference Vs Intance methods.
- Implementation question on method overloading and overridding

- Covariant and Invariants.

Exception handling

- Exception hierarchy
- Try catch finally examples
- Try with resources
- Difference checked and unchecked exception.

Data types

- Shallow/copy deep copy.
- Implementation of cloning.
- Enum and its advantages

OOPS Concept

- Tight coupling loose coupling w.r.t. OOPs.
- OOPs concept with real life example
- Basic OOPs concepts. Abstraction and Encapsulation

- Volatile keyword.
- Who is responsible for runnable to running transition?

****

Database side fundamentals

- Table Indexing and Partitioning

- Database Designing of Shopping cart
- Class diagram of Shopping cart
- What is the logic behind indexes / Internal logic of index

****

- static and final
- logic to find the occurrence of character with an array and then sort it based on value.
- comparator
- logic of generating the nth Fibonacci number in series
- Open close and linskov substitution with example
- Singleton pattern (creating a perfect singleton).
- Explain the Factory Vs Abstract factory pattern with examples.

****

- BFS & DFS

- Diff b/t put and post o String, StringBuffer and StringBuilder

- String constant pool
- What are generics
- Wildcards
- Examples

Language fundamentals
Generics

Problem Solving Questions

- Create a method to accept either 1 or 0 as a value and return exact opposite of it. (Note: Not allowed to use conditional and logical operators)

Core Java Fundamental

- Given a List if I would like to print every alternate while you call iterator. Can you implement API.
- Given a Stock Price and Stock Name, Print the Stock Name and Price in decreasing Order of Price

****
Given a list of sorted integers which may have duplicates.
Find the low and high index of given number K.
****

- Given an array with positive and negative numbers – get maximum value (sum) possible from start to end where for each element either you can go to the next element or next-to-next element

****

- Find top/max 20 elements from the list of unsorted integers.

  Provided O(n log n)

****
Scenario – Given a series of functions like -
f1(Employee e){
if(something) -> process1
f2(e)
}
f2(Employee e){
if(something else) -> process2
f3(e)
}
and so on till fn.
Given list of million employees, I want to know which all the processes were actually applied -> from the main method, I call f1() for each employee in the list.
****

- Given random list of tickets, find itinerary of a Person

****

- Find all duplicates in list of Integers
- Problem Solving, Given random list of tickets, find itinerary of a Person
- Program to print odd even values with different heads.
- Program with stream API to filter and map student list.

****

Basic concurrency problem
Given 2 threads, one printing Odd number and another printing Even numbers, a design to have output in sequential order

****

Class Immutability

- Given Employee class having id, name & Date of Joining fields, need to devise this class capable for Key part of HashMap.

- Number of Squares from a Rectangle (Recursive solution, Pseudo code)

****

Programming Scenario samples

- How will you design IPL, give high level Classes and Services required?

- A person smokes 3/4 of a cigarette and leaves 1/4. He will combine 4 leftover 1/4 pieces and make one cigarette and again smoke 3/4 of it and leave 1/4. Given number of cigarettes, return the total number of cigarettes smoked by the person.

Eg.
4 is number of input then output will be 5
3 is number of input then output will be 3 and so on.

****

- Consumer – Producer Problem (pseudo code using wait and notify)
- SQL Query – Name of Departments having more than 10 employees

- Blackbox problem (pseudo code.)

- Usage of Lambdas and Streams in Java 8: (write code logic for simple problem to convert list of integers represented as strings to list of even number integers.)

- Find if one string is rotation of another

- How to sort a list of Employees by age in Java.
- Remove all duplicate characters from a string.
- Count repeated words in a file and display in descending order.
- Find the next greater element in unsorted array.
- Check if 2 strings are rotations of each other.

- Logical Question: Second largest number in LinkedList
- Logical question – Find whether given string is a palindrome or not