TECHNICAL ROUNDS

Core Java:

Core Concepts:

OOPS concepts
Diff between aggregation & inheritance
Optional
functional interface
marker interface
Super class of all other classes in java
Go thru all the methods present in Object.class
Garbage Collection
Aggregation
Wrapper class
Covariant return type
Serialization
Streams,Terminal and intermediate methods
Flatmap usage
Default method
Comparator,Comparable implementations
Default size of arraylist when it is created
what happens when adding one more element when arraylist default size exceeds
Query optimization technique
Create custom exception
Different types of object instaniation
Different constrain used in sql
SQL for 3rd highest salary of employee
Static keyword
How will u say a class is immutable class
What happens in finally block?
Join types
Throw throws difference
how to convert a list to set, map to list, etc

Strings:

Why string is immutable
String pool
Ways to initalize a string
Diff btwn string str; and string ob = new string();

Threads & Sync:

What is a thread
Thread pool and types of thread pool
Diff btwn scheduled thread pool and cached thread pool

- multithreading
- deadlock
- lifecycle of thread
- thread lock
- notify,notifyall,join
- How to use a thread ? -Runnable & Thread.When to use them?
- Callable & future interface
- Atomic Integer
- Convert non sycn collections -> sync
- When we iterate a list and we use remove operation during iteration what will happen
- What is concurrentModificationException - how to overcome?
- concurrentHashMap, copyOnWriteArrayList
- Synchronized

DS:

Singly,Doubly,Circular LL internal working -Program to insert,calc length,delete.
Reverse a Singly linked list
Find if a string is palindrome using Linked list
Find middle element of Linked List
Internal working of :

- hashmap(hash collision, load factor, rehashing, put() & get() working)
- hashset
- Stack using array & linked list -Code
- Queue

Just Theory:

- Vector
- PriorityQueue
- ArrayDequeue
- Deque
- Blocking queue

Worst case Ques:

- Binary Tree
- Binary Search Tree
- Tree traversals-
  - BFS : Level Order
  - DFS : In,Pre,Post Traversals
- AVL tree
- When do you say a tree is balanced?How to balance a tree? - Check AVL insertion
- RBT
- Why Binary tree -> Why Binary Search Tree -> Why AVL -> Why Red Black Tree?

Graph traversals
Spanning tree -Theory
Minimun Spanning tree - Prims and Kruskals Algorithm

Algorithm of :

Search - 2 Linear,Binary
Sort- 5 Selection,Bubble, Inserting,Merge,Quick
Best sort?
Collections.sort, Arrays.sort uses which algorithm?
Collections.sort, Arrays.sort uses comparable/comparator?
[Note the Time complexity & Space Complexity for all search & sort]

Spring & MVC:

Why Spring Why spring boot -features and advantages
@requestParam,@requestmapping
@PathVariable
@component vs @service/@repository/@controller
@requestParam vs @PathVariable
@Controller vs @Restcontroller
What is the use of @SpringBootApplication?
Front controller
View Resolver
Dependency Injection
Types of dependency injection
Jus go thru some common annotations-Qualifier,Autowire,SpringBootApplication etc
Diff between GET,POST,PUT,DELETE
Http responses
Bean scopes
Worst case:
Aop
Caching in hibernate
Hibernate
Types of table relationships
Different state in hibernate

Differences:

BFS and DFS
Collection and collections
Comparotor and comparable

Failfast iterator and fail safe iterator
String builder and string buffer
@component and @ service
Interface & Abstract class
Iterator vs Enumerator
Map vs Flatmap
Map vs Set
Arraylist vs hashmap
arraylist vs linkedlist
hashmap vs hashtable
hashmap vs concurrentHashMap
hashmap vs linkedhashmap vs treemap (similarly for set)
Must hv idead abt Diff. between all the collections

Some Random Ques:
What are the variables used in js
Diff between ==, === in js
Data types used in js
How will u connect js and xml
What is elastic search?
Diff between NoSql vs SQL DBs
Difference between rest and soap webservice
Microservice questions, architecture, advantages, etc
How to analyze time complexity for a program
Junit Mockito working
String based programs from Hackerrank,leetcode

Gartner Syllabus:

About This Role

Software Engineer position responsible for design, implementation, and support of cloud-based web applications to help fulfill our Research Content strategy.

What You’ll Do

Implement and unit/integration test feature stories
Contribute to the review and analysis of business requirements, partnering closely with the business
Collaborate with team on architecture and technical design discussions
Perform and participate in code reviews, peer inspections and technical design/specifications
Actively participate in all Scrum ceremonies
Learn technologies quickly and apply effectively

What You'll Need

Bachelor's degree in Computer Science or equivalent experience in the field of software development.
2 - 5 years' experience in web application development.
Experience in Node.js is preferred (otherwise, Java or .NET experience is required)
Relational database knowledge including SQL
Experience with software development in the Cloud is preferred
Experience using JavaScript frameworks: ReactJS (preferred), Angular.
Knowledge of development methodologies, standards and object-oriented design patterns
HTML5, JavaScript, JQuery, AJAX, XML and JSON
Experience with rule engine (Drools) is preferred
Self-motivated developer who takes ownership of work
Strong problem-solving skills
Excellent verbal and written communication skills
Experience in Agile/Scrum is preferred

Who You Are

Effective time management skills and ability to meet deadlines
Excellent communications skills interacting with technical and business audience's
Excellent organization, multitasking, and prioritization skills
Must possess a willingness and aptitude to embrace new technologies/ideas and master concepts rapidly.
Intellectual curiosity, passion for technology and keeping up with new trends
Delivering project work on-time within budget with high quality