Equal hashcode with respect to map, set and problem			SQL query with group by
what is the underlying implementation of arraylist			Clustered non clustered index
Immutable class			Wait notify, fairness parameter, reentrant lock synchronized
Final			Etl what is it, why used, reporting via business object
Cloneable clone			When to use no SQL db
Cascade types			Decorator, factory pattern
One to many relationship			
Lazy loading if session is closed			
Code segments - Output			
Bulk loading/reading - how will you read from a file of million lines and update / insert in dB using Java			1. Get min elements in stack in O(1)
Marker interface			2. Hashmap working.
factory pattern			3. What is hashcode
main java basics - internally hashmap how does it work, iterator, implement iterator - collections, linkedhashmap - string buffer			4. Find all pairs in array of integers whose sum equals to k
cache - internally work			5. Basic java questions and which data structure to use in different scenarios.
internally implement GC			6. Data caching on startup of project.
how java works internally			

Optimize query, build plan, how many indexes are a good limit,
overloading, passing null as parameter,
Weak Reference,
Comparable with user defined object,
Wait -notify, deadlock,
Db query, find emp having no record for dept.
Create immutable class in detail.
Need of immutability, Upcasting downcasting

Equal hashcode with respect to map, set and problem			SQL query with group by
what is the underlying implementation of arraylist			Clustered non clustered index
Immutable class			Wait notify, fairness parameter, reentrant lock synchronized
Final			Etl what is it, why used, reporting via business object
Cloneable clone			When to use no SQL db
Cascade types			Decorator, factory pattern
One to many relationship			
Lazy loading if session is closed			
Code segments - Output			
Bulk loading/reading - how will you read from a file of million lines and update / insert in dB using Java			1. Get min elements in stack in O(1)
Marker interface			2. Hashmap working.
factory pattern			3. What is hashcode
main java basics - internally hashmap how does it work, iterator, implement iterator - collections, linkedhashmap - string buffer			4. Find all pairs in array of integers whose sum equals to k
cache - internally work			5. Basic java questions and which data structure to use in different scenarios.
internally implement GC			6. Data caching on startup of project.
how java works internally			

Optimize query, build plan, how many indexes are a good limit,
overloading, passing null as parameter,
Weak Reference,
Comparable with user defined object,
Wait -notify, deadlock,
Db query, find emp having no record for dept.
Create immutable class in detail.
Need of immutability, Upcasting downcasting

1. multi thread odd even problem
2. Static and non static methods with synchronised keyword
3. Build tools ci/cd
4. Transaction management
5. Volatile keyword
6. Registry of microservices
7. Microservices communication
8. Synchronised keyword
9. Hashmap scenario for equals and hashcode override
10. Hashmap working
11. And significances
12. Locks in java
13. Executors service
14. Oops concepts - Diff abstraction and encapsulation
15. How to create project for employee hierarchy management and they want to process leaves
16. State of object in hibernate
17. Difference between inheritance and abstract class
18. Internal working of hashmap
19. Final vs finally
20. Types of exception
21. Create custom exception
22. How to create threadpool
23. How to handle concurrent http request in web application
24. What is serialization
25. Need of serialization
26. Java 8 features
27. Lazy vs eager loading in hibernate
28. Session vs sessionfactory
29. Blockingqueue
30. Print odd even - multithreading question
31. Multithreading - basics revise
32. Hashmap - internal working, implementation, custom
33. Immutability
34. Abstraction
35. Interface vs Abstract
36. Multiple Inheritance
37. Process which reads data from data, transformation, generates a report
38. Spring batch
39. Queue - priority of different types with FIFO for same type
40. Cyclicbarrier vs countdownlatch
41. Print even odd using multithreading
42. -Thread interaction.

43. -Hashing
44. - Mostly on collection - Hashmap linkedhashmap. Java exception handling. Reenterant and
45. readwrite lock. Streams. Threadpoolexecutors.
46. Bean scope
47. Types of cache in hibernate
48. Stream api
49. Java 8 features
50. Indexes from db
51. Java 8 foreach vs advanced for loop
52. Executor framework
53. How to access two interfaces with same method with a class
54. Design problem with large data to be read persisted and again stored in queues, how to do so
55. Spring batch questions
56. Cached thread pool and fixed thread pool
57. Bean injection
58. Db queries on joins
59. Private variables accessing in other class where the other class is injected with autowired
60. Diamond problem
61. Functional Interface, Java Stream API, Concurrent HashMap, putifabsent(), RedBlack tree
62. internal structure, Why choose RedBlack tree?, Reject policy in Multithreading, how to
63. implement abort policy and what is the default policy, Semaphore, Blocking queue, spring
64. dependency injection, private autowired, spring transaction, db index, Design problem with large data to be read persisted and again stored in queues. how to do so,