**PoojaPBQuestion-**

**Pending**

1. What is difference between hibernate and jdbc
2. SessionFactory vs Session
3. What is persistence context
4. get() vs load()
5. Traisient state, persistent state, detached state
6. What is connection pool why we use it
7. Fetch types in hibernate- lazy and eager
8. @GenerationType
9. @NamedQuery
10. (Round 1)1-Explain mvn clean install
11. 2-Future class in Java
12. 3-What in Transient
13. 4-Primary key specification in hibernate(Explain Generation Strategies)
14. 5-Difference between CRUD and JPA repo
15. 6-Can we write service and controller in single file, what will happen?
16. 7- How do we send status and data together to request
17. 8-Explain @Configuration and @Component
18. 9-Difference between Eager loading and lazy loading type of fetchsize
19. 10-How do we manage security in spring
20. 11-How to increase efficiency of an API to responding to any request
21. 12-Java 8 features
22. 13- Collection classes
23. 14-How to iterate over the hashmap without using for loop
24. 15-Differece between static and volatile variable
25. 16- Why @Autowired ? Can't we use non parameterized constructor to get the object of any class
26. 17-How we achieve serialisation in hibernate models
27. What is functional interface
28. Interface which have only single abstract method.
29. Can have multiple default methods and only single abstract method
30. Exception hierarchy
31. Cyclic Barrier
32. Countdown latch
33. Singleton class
34. Why and how to synchronise singleton class
35. Garbage collection

* How many ways to collect garbage objects in java-Serial Garbage Collector.
* Parallel Garbage Collector.
* CMS Garbage Collector.
* G1 Garbage Collector.
* Z Garbage Collector.

1. Permgenspaces and metaspaces
2. Difference between spring boot and spring mvc
3. Difference between war and jar
4. What is indexing
5. Difference between procedure and function
6. What is normalisation
7. When will you prefer abstract class over interface
8. When do we use joins in SQL. What are options instead of joins
9. Http headers contain ?
10. Session based authentication or token based authentication?
11. Ans = https://www.google.com/amp/s/www.geeksforgeeks.org/session-vs-token-based-authentication/amp/
12. Cross cutting concerns
13. Custom Validation
14. How to create annotations in java
15. Hashmap internal working
16. Save, saveOrUpdate, persist
17. Hibernate conf file
18. Sorted collection vs ordered collection
19. How to implement joins in hibernate
20. Cognizant technical interview questions
21. Spring framework modules-aop,context,bean,security,spring web
22. Bean scopes in spring
23. How to handle database transactions-acid
24. Insert into tabel using hibernate
25. How to test rest api - postman,swagger,Junit-add in spring dependcy,docketapi,
26. Innerjoin, outerjoin, left outer join
27. Functional interface-single method to use lambda expression
28. Consumer supplier predicate-learn after lambda expression
29. Advantages of stream api-stream api
30. What is the use of serialisation-implement serialible
31. How to clone an object-
32. How do you create rest service in spring boot and how to integrate it-
33. Create your own arraylist class-
34. One rest service depends on other service which is taking long time, how do you analyse and fix that issue-nesting,time complexity,optimise conditions which reduce time complexity,heavy data by query-batch processing
35. Batch processing
36. How to call soap service from a rest service-
37. How does caching work in your application-hibernate caching ,cookies
38. What is dependency injection-
39. Different types of scope supported by spring
40. While writing code what will you give importance to performance or memory management-use array instead of arrayList
41. Difference between mock(@mock) and injectmock(@injectmock)-Junit mocking service ,fake call for behavirous check
42. What's the purpose of default methods in functional interface-java 8 functional interface,defualt method pi example
43. Difference between set and list-
44. How to use an api into another api-
45. Finalize keyword-
46. Reflection API-
47. How to create immutable class in java-
48. Can we use main method recursively and is it a good practice?
49. Primary Key vs Unique key
50. What is the contract between equals and hashcode ?-
51. What is the importance of string pool-
52. Design immutable class if class has mutable attributes-search how to create mutable class
53. Permgen vs Metaspace-
54. Async call in rest api-asynchronization
55. Spring Security (jwt, oAuth2)-
56. Spring bean scope
57. Spring bean lifecycle(filter request interceptor)
58. Why @ComponentScan ? And @springbootapplication uses it internally so what is the suggested way ?? To use it externally or internally ?
59. How to use jetty in place of tomcat ?
60. @Qulifier
61. Design Patterns(observer)
62. Object level locking-object specific resources locking
63. Class level locking-
64. Deadlock-
65. Explain system.out
66. All database commands
67. Treemap use cases-
68. Maven life cycles-
69. Command to create a jar
70. Difference between error and exception-
71. Read data from file and perform operations-
72. Command to create jar file
73. <https://www.freecodecamp.org/news/database-indexing-at-a-glance-bb50809d48bd/amp/>
74. lambda expression ,stream api,serealization,deserialzation-
75. why we use @autoCOnfiguration
76. Java 8,singlelton class,market interface,

**Revise-**

select

------------------------------------------------------------------------------------------------------------------------------------------------------------

**YouTube-**

**Pending-**

Difference between hash table and hashmap-nvin automation labs

**Revise-**

Oops,JitCompliler,Exception,Main Method[],

------------------------------------------------------------------------------------------------------------------------------------------------------------

**Logic Based-**

**Pending-**

Sorting Algorithm-bubble sort,selection sort,merge sort,String Occurrence

**Revise-**

Select sort,Second-third max

------------------------------------------------------------------------------------------------------------------------------------------------------------

**JavaEE-**

**Pending-**

Marker Interface,comparable-comparator,java Bean cycle,Internal working of HashMap,DataStructures-que,stack etc

Revise-

Spring fundamentals

------------------------------------------------------------------------------------------------------------------------------------------------------------

**SpringBoot-**

**Pending-**

Actuator,LoggerModel, splunk-forwarder,application.property Vs application.yml,MicroServiecs,API GateWay,Zool,

Services communication,(Service Discovery-Eureka|| Consool),Histrics,LooseCOupling in services,

**Revise**-

Thyme leaf,annotations