1. Java 8 features
2. Usage of for each and stream api in java 8
3. Generics in java, compile time or runtime
4. Sorting comparable & comparator with conditions
5. Singleton class and safeguard against Serialization
6. Time complexity big O notation i.e. O(n) or O(1) operations
7. JVM architecture
8. Class loaders
9. Volatile array
10. Atomic operations
11. String pool and heap in java
12. Association , Composition, Aggregation
13. Difference in usage of Thread class & Runnable interface
14. Spring annotations.
15. Difference between JAR, WAR and EAR?
16. Map implementation.
17. Thread Safe collections. Performance impact and how to make the collection as thread safe.
18. Difference between StringBuffer and String Builder. Why we use it
19. Multithreading. How we have used in our project.
20. Difference between runnable and callable.
21. ClassNotFoundException vs NoClassDefFoundError
22. What is future object?
23. Difference between WAIT and SLEEP
24. Can we call start method twice? What will happen if we do so (THREAD)
25. What is the use private constructor?
26. In a try-catch-finally block if there is a return statement inside the try block will the finally block will still get executed?
27. What is Garbage Collector and why is it used?
28. What is Java Executor Framework?
29. What is the difference between comparable and comparator?
30. What is a Deadlock and how to prevent it?
31. Which unit testing framework you have used?
32. How do you write a unit test case for throwing an Exception?
33. If inserting new key same as existing key in Hashmap , what will happen
34. How to generate PDF/Excel from java
35. hashmap vs hash table
36. serialization
37. final String a = a.get()//some operation
38. override a method in interface
39. JSP directives
40. JSP vs JSF
41. Java Servlet lifecycle.
42. Difference between get and Post and how it is handled in server side.
43. How JDBC works.
44. How we create a DB connection in Production environment. Where and how do we specify user name and password?
45. Logging mechanism. I said it's log4j we have used.
46. What are the different logging specifiers and where we specify the same?
47. Maven...Why we use maven and what is the difference between Maven and Ant.
48. Can we call ant scripts from Maven and how do we used maven in our project.
49. Where we configure the Maven setting and commands to use the maven.
50. Clean install and install difference
51. Where do we get jars in Maven(.m2 repo)
52. Benefits of Spring framework.
53. What is IOC
54. How it is implemented generally and project specific?
55. Will ask for Spring Transaction Management: - Since I have not worked on this, so I said I don't have any idea on the same.
56. Spring Security: - Since I have not worked on this, so I said I don't have any idea on the same.
57. What is Dependency Injection?
58. Between Setter Injection and Constructor Injection which one would you prefer and why?
59. What is the difference between Singleton and Prototype bean scope?
60. What is JDBCTemplate?
61. How do you establish connection from Spring to Hibernate?
62. Dependency Injection
63. What is difference between Views and Tables?
64. Why we use group by?
65. What are the features I have used in DB and then few questions on those like why we create index and all?
66. How can we call java code from Stored Proc...? I was also not aware of. There is a concept called as Java Stored Procedure using which we can invoke it.
67. DDL vs DML?
68. What is benefit of JNDI lookup over JDBC?
69. Difference between Inner Join and Outer Join?
70. What is self-join?
71. Fetch first 10 records from a TABLE < SQL statement >?
72. Difference between UNION and UNION ALL?
73. What are the DB features used in your application ?<SP /Package/trigger>
74. Truncate is DDL or DML?
75. Create trigger DDL or DML
76. Difference between Rest and SOAP
77. How we have used in our Project
78. What is WSDL?
79. How to create a Web Service...Step wise
80. SOAP Security.
81. In and Out of web services they will ask.
82. REST framework u implemented Jersey?
83. How client created for REST WS?
84. PDF format supported by REST?
85. Top down and bottom up approach
86. How to send userid and password to authenticate via SOAP?
87. Jaxb?
88. Jaxws vs axis?
89. What is the difference between a top-down web service and a bottom-up web service?
90. How do you manage performance issue of your web application in production environment?
91. Design pattern used in your application? Implementation of singleton design pattern? Factory pattern?
92. Where to put Business Logic? App layer or data layer explain?
93. UI Frameworks
94. JSON
95. How in spring web service JSON is passed as input
96. How do you get an attribute in xslt?
97. How Dom parser works?
98. dom vs sax
99. Design patterns. Which all you know
100. Singleton, MVC, TemplateMethodPattern
101. Junit, Used mocking?
102. Mockito and Easy mock related
103. Jdbc connection configurations and calling through java code
104. Jdbc driver initialization
105. Why in the form of Object in transforming xml from machine A to B
106. DDL & DML
107. Diff in DELETE & TRUNCATE & DROP
108. Diff in Stateless & Statefull
109. Call a store procedure through java which methods in jdbc api.
110. Encapsulation & Abstraction
111. JMS , MQ, Session beans, Queue & Topic
112. Spring Jdbc template
113. Jsp, servlet, thread, spring lifecycle.
114. difference between static synchronized and synchronized methods
115. map iterate
116. list.remove(new Integer(300))
117. Java references
118. Indexing in database
119. How to make a list immutable?
120. Volatile vs atomic
121. SOAP WS security using https. How to build?
122. SOAP https port no?
123. JBoss standalone.xml or any server.xml why it is required?
124. SOAP ws customizing with some methods to prepare a wsdl file
125. Rest web service allowing only 2 types of data say json & xml. How?
126. Media types in Rest ws
127. How try with resources works internally?
128. Multiple and multilevel exceptions
129. Java forEach iterate list inside list
130. Are generic types allowed in 1.8? How does it work.
131. Arraylist traversing both sides. Forward and reverse.
132. String reverse with recursion
133. Same bean class but diff id in spring declared singleton. How many instances?
134. Spring singleton vs java singleton
135. Find middle element in linked list
136. Stackoverflowerror vs Outofmemoryerror
137. How parallel stream works internally
138. Difference between *this* and *ClassName.class* inside a synchronized block
139. Why Dao class used as a singleton?
140. Functional Interfaces

Function - <R> apply (T parameter)

Predicate – boolean test (T t)

Supplier – () -> new Integer(10)

Consumer – (value) Syso (value)

1. this vs this()
2. Why can’t we override start method or directly call run method in Runnable interface
3. Why variables inside Interface are implicitly final
4. How to create a custom Array list which will behave as a set i.e. won’t allow duplicates

What happens if a Map’s key is a custom object and we don’t override the hashcode method. Will it allow duplicate objects then?

1. Changes in hashmap implementation in java 8. Why BST is used in bucketing instead of normal Linked list which was used before?

Hash map implementation. Example – equals method comes into picture in both .put and .get

Arraylist. Linkedlist example – internal structure, adding element is better in arraylist coz it ads at last

Interface abstract class when to use. Example -

Why spring? Loosely coupled example – A has a B, later on B1 extends B so existing code might get hampered. Instead in spring we can introduce a bean id and refer it anywhere.

Why web services? Platform independent example – if both platform are same say java still also it will be independent and loosely coupled since it doesn’t need to create a jar for every project, instead create a service and host it.

Why comparator, comparable diff? Example – Instead of hampering the existing code already developed when there is a sorting requirement we can add/write a comparator and adhere it to the existing app.

RequestBody

Log4j implementations

Database

Web server, app server

Clustering, Scalability, High Availability in web apps

Why base64 decoder is used as encryption & decryption

Any challenge you faced and how did you overcome

Any sorting other than Comparator, Comparable

Advantages of Spring jdbc template

DI, types of DI, which one you would prefer

Request Param (In spring -- @RequestParam ("id") int id)

–

Query param (In rest API)

Path variable (In spring – @RequestMapping (value = " /order/{orderId}", method=RequestMethod.GET) public String getOrder (@PathVariable String orderId))

–

Path param (In rest API)

@GET

@Path("/employee/{dept}")

Patient getEmployee (@PathParam ("dept") Long dept, @QueryParam("id") Long id)

