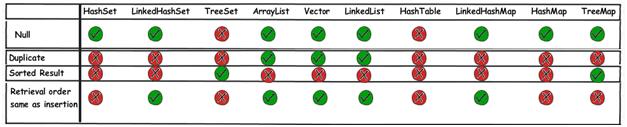
1. Explain about your current project with all technologies.
2. Which API you have used for SOAP web services.
   1. **JAX WS – Java API in Java EE5**
   2. **Apache Axis2**
3. In which version of java JAX-WS API introduced. **- Java EE5**
4. Explain about spring scopes (singleton, prototype)?
   1. **Singleton – Single instance of bean per Spring IOC Container.**
   2. **Prototype – Bean definition have any number of object instances.**
5. Have you used JavaScript/JQuery in your project?
6. Which API used for Restful web services? – **JAX RS or Apache CXF**
7. How do you make Ajax calls in JQuery? **$.ajax()**
8. Give me a solution for below scenario.

When multiple users are accessing/updating/saving data on your application simultaneously? How do you manage all requests? How do you provide consistency to your project in this situation? - **Synchronized block/ Multithreading**

1. What are the different HttpMethods for restful web services? Explain
   1. **GET**
   2. **PUT**
   3. **POST**
   4. **DELETE**
   5. **HEAD**
   6. **OPTIONS**
2. Difference between put and post.
   1. **PUT and POST operation are nearly the same with the difference lying only in the result where the PUT operation is idempotent and POST operation can cause a different result.**
3. How to make secured restful web services?
4. Difference between soap and rest.
   1. **SOAP uses SOAP protocol SOAP uses only XML (WSDL representation)**
   2. **REST uses underlying HTTP protocol, simpler and permits different data formats like JSON etc.**
5. How to improve performance of a java application.
6. What are different scopes for spring beans?
   1. **Singleton**
   2. **Prototype**
   3. **Session**
   4. **Request**
   5. **Global Session**
7. Servlet life cycle – **init, service, destroy**
8. Core Java: Collections Hierarchy, and significance of various Collections implementation and their differences.



1. Core Java: Static and Dynamic polymorphism.
   1. **Static Polymorphism – Function Overloading**
   2. **Dynamic Polymorphism – Function Overriding**
2. Core Java: Methods available in Object class and their usage in Multithreading.
   1. **notify()**
   2. **nofifyAll()**
   3. **wait()**
3. Design Pattern- Singleton, Factory.
   1. **Singleton – returns the same instance**
   2. **Factory – returns the new instance for different types which has the same base class**
4. Spring: Spring MVC request flow.
   1. **Dispatcher Servlet with the help of HandlerMapping invokes the Controller which returns the ModelAndView and will be forwarded to the view page with the help of ViewResolver**
5. Spring: DispatcherServlet and its configuration.

**<servlet>**

**<servlet-name>HelloWeb</servlet-name>**

**<servlet-class>**

**org.springframework.web.servlet.DispatcherServlet**

**</servlet-class>**

**<load-on-startup>1</load-on-startup>**

**</servlet>**

**<servlet-mapping>**

**<servlet-name>HelloWeb</servlet-name>**

**<url-pattern>\*.jsp</url-pattern>**

**</servlet-mapping>**

1. Spring: ViewResolver and its configuration.

**<bean class="org.springframework.web.servlet.view.InternalResourceViewResolver">**

**<property name="prefix" value="/WEB-INF/jsp/" />**

**<property name="suffix" value=".jsp" />**

**</bean>**

1. Spring: Bean Scopes available in Spring MVC.
2. Spring: Auto wiring and types of Dependency Injection.
   1. **Types of DI – Setter and Constructor**
   2. **Auto Wiring Types – byType, byname, constructor and autodetect**
3. Hibernate: Advantages of Hibernate over JDBC.
   1. **Supports Caching**
   2. **Provides Dialect and not database dependent**
   3. **Supports Collections and relationships like one-one, many-many etc.**
4. Hibernate: First Level and Second Level cache.
   1. **First Level – Associated with Session objects**
   2. **Second Level – Supported by APIs like EHCache, JBossCache etc.**
5. Hibernate: Difference between load and Get methods.
   1. **Load – returns the proxy object**
   2. **Get – returns the object by hitting the database**
6. Hibernate: Difference between eager and lazy loading.
   1. **Lazy Fetch – Default fetch type and hibernate will never load the related objects unless explicit request has come**
   2. **Eager Fetch – Loads all the objects related to the particular object request from hibernate**
7. SQL: Select the 3rd highest salary of the employee.
   1. **Select MIN(salary) from (select distinct top 3 salary from employee order by salary desc)**
8. SQL: Different type of joins in SQL.
9. JavaScript: How to find if a checkbox is checked in JS or JQuery.
10. JavaScript: Difference between JQuery bind and live.
11. Which collections have you used?
12. Do you know about cyclomatic complexity?
13. Have you used check style of code config?
14. How does hashing work?
15. How to create singleton object? Mention different ways?
    1. **Using Private Constructor**
    2. **Using Enum Type**
16. How many ways threads can be created?
    1. **Implementing runnable**
    2. **Extending Thread**
17. Which option would you prefer implement Runnable or extending Thread? – **Implement Runnable since java allows only single inheritance**
18. Why do we have extends Thread option when implementing Runnable is preferred?
19. What is inheritance and polymorphism? Explain with example
20. Difference between abstract class and interface?
21. What is loose coupling and tight coupling?
22. Explain thread lifecycle?
    1. **New**
    2. **Runnable**
    3. **Running**
    4. **Blocked**
    5. **Terminated**
23. Why wait (), notify () and notify All () are present in the Object class?
    1. **Wait and notify works at the Monitor level and the monitor is assigned to the Object hence they are defined in the object class.**
24. Difference between arraylist and Linkedlist?
    1. **Arraylist uses array implementation and good for search operations**
    2. **LinkedList uses doubly linked list and good for insertion and deletion operation**
25. In which scenario is arraylist faster?
    1. **Arraylist is suitable for Search operation**
26. Is Arraylist or Linkedlist thread safe? – **Both are non synchronized Class and can be made thread safe using Collections.synchronizedList**
27. How can we achieve thread safety using collections?
28. Aptitude Question: We have 3 boxes labelled Red, Green and Red Green. They have to be labelled based on the color of balls that they contain but have been labelled incorrectly now. The Red box can contain either Green or Red Green balls, Green box can contain Red or Red Green balls and Red Green box would contain Red or Green balls. Provided that we can only pick one ball at a time from one of the boxes, how many iterations would you need to label the boxes correctly?
29. How would you cut a birthday cake into 8 equal pieces using 3 cuts?
30. How do you create bean in spring?
31. Why do you go for spring over Java? What are the benefits? – **Loosely Coupled**
32. How do you inject a bean into another? Provide the syntax?
33. What is dependency injection?
    1. **Dependency Injection (DI) is a design pattern that removes the dependency from the programming code so that it can be easy to manage and test the application.**
34. What is the lifecycle of a bean? How would you use it?
35. Method A makes 3 db calls, how would you design the application such that if one of the db call fails the method should fail?
36. How do you secure a web service?
37. How do you create a certificate in Java? – **using Keytool command**
38. How do you establish a jdbc connection?
39. Which is the best option for persistence? Hibernate VS JDBC
40. How do you create a DAO bean in a multi-threaded application which is defined as Singleton?
41. Design an application in which we have 1 thread writing 10 lines to a file and another thread writing 5 lines to the same file alternatively? – **using notify inside a synchronized block having a counter variable to notify after every 5 lines**
42. Why do we need to have a try catch block for the wait method of thread? Will the code compile without a try catch block?
43. What are the different types of exceptions?
    1. **Checked Exceptions/Compile Type Exceptions**
    2. **Unchecked Exceptions/ Runtime Exceptions**
    3. **Error**
44. What is a good design for application? Would you throw an exception or throw an error? – **Throw an exception**
45. The general websites do not throw exception back to the user and they throw error codes? Would you still throw an exception?
46. How do you transform XML to CSV?
47. Have you used Quartz scheduler?
48. How does Quartz scheduler work?
49. How do you manage fallback in case of different quartz schedulers running on individual servers and one of the server fails?
50. How do you design logging mechanism for an application where we have method A, B, C & D calling each other such that we are able to get the details of the method calls, user who called and other information from the log file? This is a critical application and not even a single message should be lost? – **using Spring AOP**
51. How do you get access to the spring application context in a web application?
    1. **Using ClassPathXMLApplicationContext**
52. How do you get access to a non-singleton bean from the context?
53. Difference between assertion and composition?
54. What is inheritance?
55. What is the difference between association, aggregation and composition?
    1. **Composition, when one class owns other class and other class cannot meaningfully exist, when it's owner destroyed**
    2. **Aggregation even the owned class is destroyed the child class exists meaningfully**
    3. **Association - relationship between two objects**
56. In which of the relationship mentioned above if object A is garbage collected then the associated object B is also garbage collected? - **composition**
57. Auto wire in spring, is it aggregation or association?
58. Print even an odd number simultaneously using 2 threads? -– **using notify inside a synchronized block having a counter variable to notify**
59. Do you know about callable interface?
    1. **Callable interface can return a value and throw checked exception**
60. Have you heard of countdown latch, barrier or semaphore
61. Where do you use factory and abstract factory pattern?
62. Spring application context used which pattern?
63. Calendar.getInstance (), which pattern does it implement?
64. 3 different ways of implementing Singleton pattern?
65. Why do we need web service as compared to servlets?
66. In case of null pointer exception what kind of response would the front end user receive?
67. REST web service is returning response in XML format, how do you get JSON from it?
68. While testing a spring application, how to you supply the context to the mokito framework?
69. Is mock object created in Mokito a map or a collection?
70. How do we test a private method? – **Using Java Reflection API**
71. How do you test db related methods?
72. How do we exclude test related classes from Maven? How can we do it without touching the code?
73. What has been your experience with performance testing?
74. How do you fine tune a long running DB query?
75. What is the criteria for creating an index? -**When the particular column is used for join operations**
76. How do you check if XML is compliant with the schema?
77. Do we need to use encoders in schema validation in encoder? Which encoder did you use?
78. How do we analyze performance problem in a threading application?
79. How do we integrate Hibernate with spring? – **Spring manages the hibernate sessions**
80. What are your core competencies? --> Here, please start with Core Java and so on….don’t start with some narrow areas like JUnit
81. How do you write unit test for private method? – **Using Java Reflection API**
82. How do you convert a jar file in to war in Tortoise SVN?
83. How do you secure a web service? Can you provide the design?
84. What are the boiler plate annotations provided by spring?
85. Describe each annotation in spring and mention the situation in which they would be used?
86. I have an abstract class and another class which extends this class. How would you define the beans in Beans xml? How many objects would be created?
87. What would be the scope of the bean created in above situation?
88. I have to create a DAO bean, should this be marked Singleton or Prototype?
89. How multiple updates and insert can be performed if DAO is marked Singleton? How threads will manage DAO resource sharing?
90. What is synchronization? How do you stop/terminate a thread which is executing? **Synchronization ensures thread safety in an application**
91. Which logging framework have you used?
92. Are you aware of AOP concepts in spring?
93. Which is a better way of defining beans through xml or through annotations?
94. How do you tell spring to use annotation in your application? – **using <mvc:annotation-driven> or <context:annotation-driven> tags**
95. What collections have you used?
96. Difference between .equals and ==?
    1. **Equals – compares both the values and the reference**
    2. **== - compares only the reference not the values**
97. What order of elements does arraylist have post insertion?
98. Which list is preferred if we do a lot of insertion and retrievals?
99. How does Hashing work?
100. Can HashMap contain null key and null values? – **Can have one null key**
101. I have a student class with attributes name and roll no. I have inserted couple of student objects in a list. I would like to sort the list by roll no. How can that be achieved?
102. A 10 sec to fetch 1 lakh records from the db, I want to use Threading to bring it down to 1 sec how would you do that?
103. Is there any design pattern for this?
104. How different thread communications happen? If a Thread has to inform the parent thread that it has completed, should it call notify or notifyAll?
105. How to print numbers from 1 to 10 in the same order using threads in which one of them prints odd numbers and other one even numbers? **using notify inside a synchronized block having a counter variable to notify**
106. What all spring features have you used?
107. What is a spring lifecycle?
108. Where does the context xml reside in the application?
109. How do you let the application know where to look for context xml? – **using ContextLoaderListener**
110. How do you design a web service application which supports both SOAP and REST?
111. If we need to put a transaction control, which layer would we put the transaction on?
112. What is the difference between Factory, AbstractFactory and FactoryMethod design patterns?
113. 3 different ways of implementing the singleton design pattern?
114. What all libraries would you require for a test driven development using JUnit?
115. How do you measure the performance of an application?
116. What components impact the performance of an application?
117. How do we control/gauge the heap performance?
118. How do we check the performance of DAO layer alone? How to identify the time taken by the DAO layer to run?
119. Have you heard of Spring AOP? How can the above be achieved through that?
120. Technology & project details
121. Java 5 features
122. Generics - wild card syntax - why do we go for it
123. Auto boxing
124. Collections - HashMap get() method implementation, Fail fast and Fail safe iterators, ConcurrentHashMap
125. Spring - dependency injection, bean life cycle, bean scopes
126. Scenario based question on bean scopes
127. How do you secure your application? (Spring Security)
128. Spring MVC rest based annotations - @ResponseBody
129. How does your application communicate with the UI? Output format - JSON/XML?
130. How does your application communicate with the database? (Any ORM technology)
131. What is ThreadLocal? Where it is used?
132. What is CountDownLatch? In which scenario it is used?
133. Does ConcurrentHashSet available in Java concurrent package?
134. How ConcurrentHashMap internally works?
135. What is Lock(), UnLock(), ReentrantLock() and TryLock()? How it is different from synchronized Block in Java?
136. SOAP supports other than HTTP protocol?
137. REST supports other than HTTP protocol?