1. How to implement Application security

<https://docs.oracle.com/javaee/7/tutorial/security-webtier002.htm#BNCBM>

1. Spring IOC-Explain

Check spring document

3. Have you worked on Design patterns? if so, how did you implement?

4. Give us some example on prototype scope, when can we use that?

<http://www.mkyong.com/spring/spring-bean-scopes-examples/>

5. Multithreading-->what is better? single threading or multithreading?

Why would we use a multi threaded application vs. a single threaded application? First we must define multithreading. Multithreading is a feature of an operating system that allows programs to run subcomponents or threads in parallel. Typically most applications only need to use one thread because they do not perform time consuming tasks. The use of multiple threads allows an application to distribute long running tasks so that they can be executed in parallel. This gives the user the perceived appearance that the application is working faster due to the fact that while one thread is waiting on an IO process the remaining tasks can make use of the available CPU. The allows working threads to execute in tandem so that they can be competed sooner.

**Multithreading Benefits**

* Improved responsiveness — Users usually report improved responsiveness compared to single thread applications.
* Faster applications — Multiple threads can lead to improved application performance.
* Prioritization — Threads can be assigned a priority which would allow higher priority tasks to take precedence over lower priority tasks.

**Single Threading Benefits**

* Programming and debugging —These activities are easier compared to multithreaded applications due to the reduced complexity
* Less Overhead — Threads add overhead to an application

**When developing multi-threaded applications, the following must be considered.**

* Deadlocks occur when two threads hold a monitor that the other one requires. In essence each task is blocking the other and both tasks are waiting for the other monitor to be released. This forces an application to hang or deadlock.
* Resource allocation is used to prevent deadlocks because the system determines if approving the resource request will render the system in an unsafe state. An unsafe state could result in a deadlock. The system only approves requests that will lead to safe states.
* Thread Synchronization is used when multiple threads use the same instance of an object. The threads accessing the object can then be locked and then synchronized so that each task can interact with the static object on at a time.

6. Have you used SOAP UI if so, how to use that?

7. Explain struts validation

Check struts document

8. Tell us about xml parsing and types of XML parsing which you used

<http://www.tutorialspoint.com/java_xml/java_xml_parsers.htm>

9. Have you worked on SQL queries?

<http://www.w3schools.com/sql/>

10.Do you know AspectJ

Using ordinary Java, it can be difficult to modularize design concerns such as

* system-wide error-checking strategies
* design patterns
* synchronization policies
* resource sharing
* distribution concerns
* performance optimizations

AspectJ is a simple and practical extension to the Java programming language that adds to Java aspect-oriented programming (AOP) capabilities. AOP allows developers to reap the benefits of modularity for concerns that cut across the natural units of modularity. In object-oriented programs like Java, the natural unit of modularity is the class. In AspectJ, aspects are concerns that affect more than one class.

More information can be found at:  
[**http://aspectj.org/**](http://aspectj.org/)

11. What is Spring AOP

12.Have you worked on batch scripts?

13.What all IDEs have you worked on?

12.In spring have you used Annotations? or XMLs?

13.If Error is there in config file, when will it show the error is it run time or compile time?

14.How will you deploy an application?

15.Where is the installed application visible?

F:\apache tomcat\apache-tomcat-7.0.55\webapps

16. Do we need to Start the server to view changes of class?

Yes

17. Do we need to start the server to view changes of jsp?

Because when Tomcat is asked to execute a JSP, is compares the modification date of the JSP file with the modification time of the compiled class corresponding to this JSP, and if more recent, it recompiles on the fly before executing it.

This is BTW an option that should be turned off in production, because it takes time to perform this check.

18. what is war file and ear file?

19. Have you worked on Hibernate?

20. When spring beans are initiated? What exceptions are thrown when class is missing?

21. How maven is configured in eclipse? How eclipse knows about .m2?

22. What components are present in spring.xml?

23. Can a bean mapped to an interface?

24. How will you know if one node is up and another one is down in a cluster?

25. What module in spring you have used?

26. How to know if application is correctly installed?

27. Write code to reverse a sentence?

28. Write code to reverse each word in a sentence?

29. Write code to count occurrence of a string in a sentence?

**30. Why we need singleton class?**

**31. How we can stop multiple thread manipulating singleton state?**

32. What thread is Garbage collection?

33. What do you mean by daemon thread?

34. Who and how it runs?

35. What is package? Why we need package?

36. What is checked exception?

37. How runtime is not checked exception even though it extends exception?

38. How to code own exception?

39. What design patterns I worked on?

40. Why we use factory pattern?

41. How you write singleton class?

42. What are things to keep in mind for performance improvement?

43. What is memory leak? Why it happens?

44. What is marker interface?

45. What methods are in Serializable interface?

46. What are the scopes available in spring?

47. What is dependency injection?

48. What are the different types of Dis in spring?

49. How you can eliminate junk characters from incoming JSON before it is converted into object?

50. What happens when we setup jdbc connection?

51. How connection pools are managed and how much memory is allocated to it?

52. How filters work?

53. Common exceptions in Spring?

**Persistent Systems**

What are GC algorithms?

How to find out null pointer exceptions? What tools to use?

What is immutable class? What are things to take care to make a class immutable?

<http://javarevisited.blogspot.in/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html>

How to make a class for which only 3 instances are allowed?

Tell me about bean life cycle and why we need so many stages?

What is difference between SOAP and RESTful?

How you write a LinkedList program?

**How does get(Key key) method works internally in HashMap, and Hashtable in Java?**  
  
<http://java67.blogspot.com/2013/06/how-get-method-of-hashmap-or-hashtable-works-internally.html#ixzz45ioG9Rsd>

<https://tekmarathon.com/2013/03/11/creating-our-own-hashmap-in-java/>

**MYSIS round -2**

What are the top 5 things I check during code review?

What do you mean memory leak?

<https://dzone.com/articles/what-memory-leak-java>

To develop a new application how will you decide how many classes required?

Why we need multiple classes? Why can’t we put all code in same class under different methods?

Why we need interface? Why can’t we use multiple classes?

Why handle exceptions with try catch instead of declare it and let JVM handle it?

How I check performance of the application?

What is normalization? Why we need multiple tables? Why can’t we use single table?

How I check GC logs in WAS server?

How to write an immutable class?

Now show how can you create multiple objects?

What will happen if you assign new object to old ref variable?

When newly created object is lost how can we get it back?

What is encapsulation?

If we do not put setter and getter then is code not encapsulated?

What is abstraction?

Is only abstract class is abstraction?

SAP interview

1. What is classloader?
2. How classloader works?
3. How many types of classloader?
4. Give some design patterns in JDK?

<http://stackoverflow.com/questions/1673841/examples-of-gof-design-patterns-in-javas-core-libraries>

1. What is blocked queue in jdk 5?
2. Can we write class inside interface?
3. What is index in DB?
4. How it is internally works?
5. How to select top 10th salary in table?
6. SELECT   empno,
7. 2           ename,
8. 3           sal,
9. 4           deptno
10. 5    FROM   (SELECT   e1.\*, DENSE\_RANK () OVER (ORDER BY sal DESC) rnk
11. 6              FROM   emp e1)
12. 7   WHERE   rnk = 3
13. What is singleton?
14. How you write singleton?
15. How are different ways of creating singleton class?
16. How a singleton class cannot be singleton?
17. How deadlock happens?
18. How to get rid of deadlock?
19. Is servlet multithreaded?
20. How then objects are shared between threads?
21. What difference is between create statement and prepare statement?
22. How transactions are handled in spring?
23. How put works in hashmap?
24. What is concurenthashmap?
25. How to write own concurrenthash map?
26. What are copytowrite\* collection class?
27. How consumer producer works?