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?