**Boddu Recommendated Site & Questions:**

1. **How to do in java - website**
2. **Java revisted – website**

[**http://crunchify.com/how-to-create-a-simple-in-memory-cache-in-java-lightweight-cache/**](http://crunchify.com/how-to-create-a-simple-in-memory-cache-in-java-lightweight-cache/)

[**http://www.java2novice.com/java-interview-programs/arraylist-implementation/**](http://www.java2novice.com/java-interview-programs/arraylist-implementation/)

**Executor frame work**

**Concurrent Collection API.**

**ConcurrentHashMap**

**arrayOnCopyList & Set.**

**Executor service frame:**

**Used to increase the performance.**

**e.g. if a query result 1000 records, then we can create 10 thread & each thread will get 100 records, finally all results will be merge into single concurrent list.**

**It is useful when there is no dependency & for sequential task.**

**Can’t use for dependent task &**

**Learning Content**

**Useful Link for**

1. Spring
2. Hibernate
3. Restful Web services
4. Other

**General Interview Questions**

1. Java
2. Spring
3. Hibernate
4. Restful Web services
5. Other

**Place Holder for Client Interview Question**

1. Java
2. Spring
3. Hibernate
4. Restful Web services
5. Other

**Code Quality**

1. Coding standard
2. Code coverage
3. Code review
4. Junit
5. Build
6. Code Analysis
7. Performance Tuning

**Design Patterns**

**Data Structure**

**Log File Filtering**

**Database Concepts**

# J2EE authentication and authorization

**Assignment**

**Useful Link for**

**Hibernate**

<http://www.tutorialspoint.com/hibernate/>

<http://hibernate.org/orm/documentation/>

**Webservices**

<http://it-ebooks.info/book/392/>

**Spring**

<http://www.tutorialspoint.com/spring/spring_tutorial.pdf>

**Java interview Questions**

1. Please explain the different class loader in java

<http://javapapers.com/core-java/java-class-loader/>l

http://javarevisited.blogspot.in/2012/12/how-classloader-works-in-java.html

Java class loaders can be broadly classified into below categories:

* Bootstrap Class Loader  
  Bootstrap class loader loads jdk core classes like java.lang, java.util etc. These are classes that are part of java runtime environment from rt.jar. Bootstrap class loader is native implementation and so they may differ across different JVMs.
* Extensions Class Loader  
  JAVA\_HOME/jre/lib/ext contains jar packages that are extensions of standard core java classes. Extensions class loader loads classes from this ext folder. Using the system environment propery java.ext.dirs you can add ‘ext’ folders and jar files to be loaded using extensions class loader.
* System/Application Class Loader  
  Java classes that are available in the java classpath are loaded using System class loader.

When a class name is given, class loader first locates the class and then reads a class file of that name from the native file system. Therefore this loading process is platform dependent.

Classes can also be loaded from network, constructed on runtime and loaded.

 when a class is loaded into JVM, you have an entry as (package, classname, classloader). Therefore the same class can be loaded twice by two different ClassLoader instances.

1. How the GC is working java.
2. Please give an example of runtime and compile time polymorphism.
3. Can **override the main method?**
4. int[] num = {1,2,3,4,5,1,2,3,4,2,3,1,6,9,8,7,5…….}; please find the duplicate number without using java API
5. **What is concurrency page is in java**
6. **What is atomic operation?**
7. **What is bubble sort and give the examples.**
8. **What is deep and shallow copy in java.**
9. what is meaning of final keyword in inner class?
10. what is polymorphism and method overriding works on the which access modifier ?.
11. what is serialization?
12. what is reflection in java? (used in IDE, Debugger, Testing tools) used with annotation

http://www.javatpoint.com/java-reflection

1. what is thread and which is better extends thread and implement runnable interface?
2. how is used static keyword in serialization and what will impact?
3. Servlet is single thread or multi thread by default?
4. Write a program for checking palindrome
5. What is final class
6. what is finally and where it is used
7. Whats the difference between procedural language and java?
8. what are oops concepts?
9. Difference between overloading and overriding?
10. What is marker interface and when it is used?
11. Can static variable be serialized?
12. Difference between interfaces and classes? Why do we need it, why can’t the things be done in class alone instead of interfaces?
13. Why threading is used in java?
14. What is intrinsic lock? Difference between intrinsic lock and reentrant lock.

<https://mishrabagish.wordpress.com/2012/12/17/explicit-vs-intrinsic-locks-in-java-ensuring-thread-scheduling-fairness/>

1. What is volatile variable? When it is used?

http://javarevisited.blogspot.in/2011/06/volatile-keyword-java-example-tutorial.html

<http://tutorials.jenkov.com/java-concurrency/volatile.html>

1. How wait and notify works in java?Basicaly the internals of thread synchronization with some samples to confuse us.
2. What is deadlock and how do you avoid it?
3. What are the differences between HashSet and ArrayList?
4. What are hash functions and why do we need it?
5. What is the significance of equal’s method in HashMap?
6. What does iterators remove function does?
7. Why OOPs is better compare to procedural language? Give some key benefit of OOPS over

procedural language.

1. Why exactly you need interface? Can we use abstract class in place of interface?
2. Is a variable allowed inside an interface?If not why so?
3. Why marker interface don't have any methods defined within them?
4. Can we write our own marker interface?

Basical you can check IS\_A Relationship & perform the task Like serialization/cloning.

Externalization is used when you want to customize your read/write serialized objects using readExternal() & writeExternal().

1. When do you use LinkedList?
2. What is the difference between set and List?
3. When do u use vector?Is all methods in Vector are synchronized?-Here code sample is given and

and they will check how the synchronization is done.

1. Why iterator is used? Which is better looping through list manually or using iterator?
2. When LinkedList is used and Why?
3. ArrayList holds order in which order?
4. Should read method be synchronized in a vector
5. How to make array list thread safe

There are two ways to synchronize explicitly:

1. Using Collections.synchronizedList() method
2. Using thread-safe variant of ArrayList: CopyOnWriteArrayList
3. Synchronization lock , re-entrant locks
4. Real time example for re-entrant locks
5. We have a CSV file with employee data (4-5 fields): in which collection will we store it and how will we store it
6. How will we parse the CSV file : File API used
7. In buffered reader can we go back to the previous read line and if yes how
8. What is serialization and why should we go for it when we want to save the state of an object
9. How is it different from externalization
10. One more question on RMI w.r.t serialization .
11. How will we compare 2 objects in java
12. How does comparison of objects work in array list
13. Why string is called immutable?
14. Why you should use String buffer?
15. What are different access modifiers in java?
16. What is default access?
17. what is generics ,why do we need it ?
18. If we have 2 sorted list of strings ,how to create  a sorted merged list  ?
19. We have a telephone directory. What type of collections you will use? Why?
20. What is singleton design pattern
21. Difference between Java design pattern and Spring design pattern
22. Why do we need to override hashCode and equals method
23. how to identify the LinkedList second last element with one iteration
24. Can we override the static method in child class
25. what is serialization and deserialization and how did you achieved it your application

**Hibernate :**

1. **Please explain what are the ORM levels?**
2. **Please explain Hibernate xml mapping file.**
3. **What are the important tags of hibernate.cfg.xml?**
4. **What are the Core interfaces are of Hibernate framework?**
5. **In Hibernate how to connect RDBMS.**
6. **What’s the difference between load() and get() method?**
7. **Please explain different methods in Hibernate.**
8. **Please explain different methods in Hibernate.**
9. **What is Hibernate Template?**
10. **How do you switch between relational databases without code changes?**
11. **How to make a relationship in Hibernate.**
12. **Please explain all collection types in Hibernate.**
13. **What are the ways to express joins in HQL?**
14. **What are the types of inheritance models in Hibernate?**
15. **Please explain the Hibernate Criteria API.**
16. **How to call Hibernate from spring.**
17. How to use Enum in hibernate ?
18. Fetching Strategies
19. How to specify FetchMode
20. How to specify  Mapping (one-many/many to many) in hbm ?
21. How to use subclasses in HBM.
22. How to write subselect query

# Web services:

1. Explain about how we upload a document using Rest call? Spring
2. **What is REST?**
3. **What is SOAP, WSDL?**
4. **What is difference between restful webservice VS SOAP?**
5. **How to pass JSON in Restful webservices.**
6. **What are the Tools/API are there for developing /testing web services.**
7. **Please explain different methods in Restful webservice.**
8. **In Restful web Service using JSON or xml, which one will be good and why.**
9. **How do you annotate a method to respond to HTTP GET requests**
10. **How will you handle application error scenarios in Restful web service**
11. **What are the pros and cons of using the Restful framework?**
12. **Explain different annotations used in Restful webservice.**

**Spring:**

1. **What are different modules in spring?**
2. **Explain** different types of Autowire.
3. What is default scope of bean in spring and how u can make it prototype.
4. What is view resolvers.
5. How to generate pdf in spring.
6. **What is difference between ApplicationContext and BeanFactory.**
7. **What type of transaction support spring.**
8. What is joint Point and point cut?
9. What is DelegatingVariableResolver?
10. What to call structs in spring.
11. What is DelegatingVariableResolver?
12. How do you setup LDAP Authentication using Spring Security?
13. Explain Bean-LifeCycle

**Place Holder for Client Interview Question**

1. Java
2. Spring
3. Hibernate
4. Restful Web services
5. Other

# AOS

1. Collections framework - Hash map --> Hash code and equals

2. Tree Set

3. Binary Search algorithm

4. Singleton pattern, factory pattern

5. Servlets -  how MVC dispatcher servlets work , how HttpServlet work

6. Cookies -->Questions on request/response objects

7. String immutable questions and memory heap hit during string manipulation

8. reverse string without API

9.Algorithm to write a factorial of a number

10. Tail recursion concept in phython

11. Java program to write to develop a Battleship  with n/n players. Write the code accessible for audting purpose

12. Sping framework

                 -> how lazy init works

                 -> how spring cyclic dependency works

                 -> how auto-wiring works?

                 -> Explain dependency injection

13. Hibernate

                 ->Named queries

                 -> Persistent and detached object

                 -> lazy loading

14. Explain Mutli-tier application

15. Web-services --> WSDL, SOA

# ****MyPage Questions****

**Qustions from Ashish for MyPage project**

**1) Explain your current project**

**2) Do you have experience with Webservices?**

**3) How does a web service know that the request has come from a valid/authenticated user and NOT an external hacker?**

**4) Do you have experience with Spring? In which projects of yours have you used Spring?**

**5) What all bean scopes have you used in your projects? Where did you define/use these bean scopes in your projects?**

# ****AMFT Questions from Chitra Thampi****

- **Previous project experience(type of work and project detials) : Project structure, how web pages interact when a js library(like jQuery, DOJO) wrapped over jsp**

**- Spring modules I have worked on**

**- Hibernate Joins using annotations**

**- Spring RESTful Webservice**

**- Spring Webservice(Spring-ws)**

**- Spring Core : Application context**

**- Spring web MVC : Web application context**

**- Building components required for a small web application in spring(a login/submit page)**

**Irecon**

Murali:

1. Introduce yourself and what you are doing in the current project.

2. What implementation of Jax-RS did you use? Why did you not use Spring Rest but went with a different implementation (In this case JBoss Rest Easy)

3. Can you explain how does session work? More questions were asked based on the answer

4. How comfortable are you with DB? If I have to put index on a column, what indexes do I need to create?

5. What are clustered indexes? Difference between clustered and BitMap indexes?

6. How will you analyze a query if the performance of the query is slow?

7. Did you work on the project where there is onsite-offshore model? Are you comfortable with it?

8. Questions on Java Concurrency (Latches, Callable and Future, Executors)

Sangeeta:

1. Introduce and tell me what you had been doing for last 2 years?

2. Did you work with a onsite-offshore model?

3. How many times you had been to US and what are the different roles/activities that you have played here?

4. She asked me if I had any questions and later on explained me the role that I am going to play.

Abde:

1. If the application throughput has gone down due to heavy load or any other reasons, what are the options that you can think of to solve the problems (Mainly he was expecting here how we can do horizontal scaling and what all are the steps)

2. If there is any exponential increase in data in the database what are the reasons you think can lead to the problems

3. Have you worked on any profiler? I said YES, and he has asked me to explain what I have used it for and what was the problem statement for which I have tried using a profiler.

4. If I give you a Thread Dump at 10:00 and 10:05 what are the things you will look for between these 2 thread dumps and what should be the analysis

5. Have you worked on SQL Query Tuning. After that he has asked a few questions on Unique Index Scan and Full Scan

6. Explain me a Web MVC flow for any framework that you have used. I have explained him with Spring MVC

**GNCS interview.**

1. How GC works or how to resolve OutOfMemoryError in java?

Hint: Its a wide question but what they want hear is that How the object profiling happens and how heap generation is being handled by different JVM implementation. How to adjust the memory allocation for different generation (e.g. b/w young generation, tenured generation & Perm generation)

2. How to improve performance of system?

Hint: They are checking your approach on capture the problem area and the different alternative solution you provide to solve the issue.

3. How concurrentHashMap work and what level of isolation is required to maintain the consistency of data?

Hint: They are interested in algorithmic knowledge you have related to this area.

4. How to maintain the versioning while using the same data from or to different datasource concurrently?

5. What kind system knowledge you have related to linux platform?

For example: Can write a shell command to check the CPU usage.

# Data Structure

1. Write a Singly Linked List class without using Java Collection API's

**Log File Filtering in UNIX**

[**http://www.tutorialspoint.com/unix/unix-pipes-filters.htm**](http://www.tutorialspoint.com/unix/unix-pipes-filters.htm)

**In the Log file format should be key value pair so that when Apple want to implement the logs in Splunk it is easy. Because in Splunk accept only key and value pair**

**Database Concepts**

# http://sqlzoo.net/

**Code Quality**

1. Coding standard

https://docs.oracle.com/cd/A97688\_16/generic.903/bp/java.htm

1. Code coverage

http://www.sonarqube.org/pick-your-code-coverage-tool-in-sonar-2-2/

1. Code review

https://www.atlassian.com/software/crucible/overview

1. Junit

<http://www.tutorialspoint.com/junit/>

1. Build

<http://jenkins-ci.org/>

1. Code Analysis

Checkstyle,pmd

# Design patterns

1. - <http://www.corej2eepatterns.com/>
2. Explain Factroy Design pattern with an example
3. Explain Builder Design pattern with an example
4. Singleton patterns

# J2EE authentication and authorization

# Declarative J2EE authentication and authorization with JAAS

* http://www.oracle.com/technetwork/developer-tools/jdev/index-089689.html

# J2EE

1. What is filter? Why do we need to use Filters

2. Have you used Ajax in any applications before? Explain

**Assignment**

**Develop a shopping cart web application using Spring MVC,Rest based services and Hibernate.**

**Condition**

1. **A End point, which can be used by the customer to order items to make a purchase**
2. **A End point, which can be used by the customer to return the order items within 30 days of the original order.**
3. **A data model for the shopping car web application(ER Diagram)**
4. **Code should be clean like production version (error and exception handling)**
5. **Use JSON as the input to the web service endpoint exposed to order and return items**