# Thread

1. What is better? Single threading or multithreading?

**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.

# Spring

1. What is Spring IOC?

Check spring document

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.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?

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

22. What components are present in spring.xml?

23. Can a bean mapped to an interface?

25. What module in spring you have used?

47. What is dependency injection?

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

53. Common exceptions in Spring?

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

1. How transactions are handled in spring?
2. How spring comes to know about the spring config xml?
3. What happens before ApplicationContext is created?
4. 46. What are the scopes available in spring?

# Collections

How you write a LinkedList program?

<http://crunchify.com/how-to-implement-a-linkedlist-class-from-scratch-in-java/>

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/>

1. What is blocked queue in jdk 5?
2. How put works in hashmap?
3. What is concurenthashmap?

<http://crunchify.com/hashmap-vs-concurrenthashmap-vs-synchronizedmap-how-a-hashmap-can-be-synchronized-in-java/>

1. How to write own concurrenthash map?
2. What are copytowrite\* collection class?
3. What is ConcurrentModificationException?
4. How can you write a code which will create ConcurrentModificationException?

<https://examples.javacodegeeks.com/java-basics/exceptions/java-util-concurrentmodificationexception-how-to-handle-concurrent-modification-exception/>

1. What is difference between collection.remove() and iteration.remove()?
2. What are differences between hashmap, concurrenthashmap and synchronizedhashmap?
3. How concurrent collections works? How to write own concurrent collections?

J2EE

1. How to implement Application security

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

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

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

13.What all IDEs have you worked on?

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.

Design Pattern

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

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

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

40. Why we use factory pattern?

41. How you write singleton class?

39. What design patterns I worked on?

1. What is singleton?
2. How you write singleton?
3. How are different ways of creating singleton class?
4. How a singleton class cannot be singleton?

<http://www.journaldev.com/1377/java-singleton-design-pattern-best-practices-with-examples>

1. Write your own decorative pattern?
2. What is scope of singleton?
3. How can you break singleton?

# GC

1. **What thread is Garbage collection?**
2. Daemon
3. **What is daemon thread?**
4. Daemon thread is a low priority thread, which runs intermittently in the back ground doing the garbage collection operation for the java runtime system.
5. **Which method is used to create the daemon thread?**
6. setDaemon method is used to create a daemon thread.

* You can make any java thread as daemon thread. Daemon threads acts like service providers for other threads running in the same process.
* Daemon threads will be terminated by the JVM when there are none of the other threads running, it includs main thread of execution as well.
* To specify that a thread is a daemon thread, call the setDaemon method with the argument true.
* To determine if a thread is a daemon thread, use the accessor method isDaemon.

package com.myjava.threads;

public class DaemonThread extends Thread{

    public DaemonThread(){

        setDaemon(true);

    }

    public void run(){

        System.out.println("Is this thread Daemon? - "+isDaemon());

    }

    public static void main(String a[]){

        DaemonThread dt = new DaemonThread();

        // even you can set daemon constrain here also

        // it is like dt.setDeamon(true)

        dt.start();

    }

}

<http://www.java2novice.com/java_thread_examples/daemon_threads/#sthash.gwM2ZwYK.dpuf>

1. **What is memory leak? Why it happens?**

There actually four categories of memory issues with similar and overlapping symptoms, but varied causes and solutions:

* **Performance**: usually associated with excessive object creation and deletion, long delays in garbage collection, excessive operating system page swapping, and more.
* **Resource constraints**: occurs when there’s either to little memory available or your memory is too[fragmented](https://en.wikipedia.org/wiki/Fragmentation_(computing)) to allocate a large object—this can be native or, more commonly, Java heap-related.
* **Java heap leaks**: the classic memory leak, in which Java objects are continuously created without being released. This is usually caused by latent object references.
* **Native memory leaks**: associated with any continuously growing memory utilization that is outside the Java heap, such as allocations made by JNI code, drivers or even JVM allocations.

<https://www.toptal.com/java/hunting-memory-leaks-in-java>

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

<http://coderevisited.com/memory-leaks-in-java/>

<http://www.pointsoftware.ch/en/under-the-hood-runtime-data-areas-javas-memory-model/>

<http://cmk-java.blogspot.in/2014/08/java-virtual-machine-internals.html>

<http://blog.ragozin.info/2011/06/understanding-gc-pauses-in-jvm-hotspots.html>

<https://plumbr.eu/blog/garbage-collection/minor-gc-vs-major-gc-vs-full-gc>

<http://www.oracle.com/webfolder/technetwork/tutorials/obe/java/gc01/index.html>

<https://plumbr.eu/handbook/garbage-collection-algorithms>

<http://www.ibm.com/developerworks/library/j-jtp11253/>

1. **What are GC algorithms?**

<https://plumbr.eu/handbook/garbage-collection-algorithms>

<https://www.linkedin.com/pulse/java-memory-architecture-garbage-collection-issues-kumar-puri>

1. **What are strong reference and weak reference?**

<https://community.oracle.com/blogs/enicholas/2006/05/04/understanding-weak-references>

How I check GC logs in WAS server?

1. How to analyze outofmemory exception?
2. How to know the code causing outofmemory exception?
3. How mark and sweep works?
4. How to code memory leak?

OOPs

WebServices

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

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

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

Struts

7. Explain struts validation

Check struts document

SQL

9. Have you worked on SQL queries?

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

Write Code

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?

12.Have you worked on batch scripts?

18. what is war file and ear file?

19. Have you worked on Hibernate?

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

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

26. How to know if application is correctly installed?

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?

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

44. What is marker interface?

45. What methods are in Serializable interface?

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?

**Persistent Systems**

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?

What is difference between SOAP and RESTful?

**MYSIS round -2**

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

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?

<http://www.studytonight.com/dbms/database-normalization.php>

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. Can we write class inside interface?

<http://way2java.com/java-lang/inner-classes-java-lang/class-inside-interface/>

<http://stackoverflow.com/questions/10553475/what-is-the-use-of-writing-a-class-inside-an-interface>

* Nested interface must be public if it is declared inside the interface but it can have any access modifier if declared within the class.
* Nested interfaces are declared static implicitely.

public interface Employee {

class Role{

public String rollname;

public int roleId;

public Object person;

}

Role getRole();

}

The class Role is used by the employee interface in getRole() method.

The designer of the interface decided that this class is so tightly coupled with the interface that it is worth to define it inside that interface to emphasize how important that class is for the interface.

Also it provides semantic namespace for the class: employee.Role.

It limits the scope of the class to where it belongs.

1. What is index in DB?
2. How it is internally works?

<http://www.programmerinterview.com/index.php/database-sql/what-is-an-index/>

1. How to select top 10th salary in table?

SELECT   empno,

  2           ename,

  3           sal,

  4           deptno

  5    FROM   (SELECT   e1.\*, DENSE\_RANK () OVER (ORDER BY sal DESC) rnk

  6              FROM   emp e1)

  7   WHERE   rnk = 3

1. How deadlock happens?
2. How to get rid of deadlock?

<http://www.javacreed.com/what-is-deadlock-and-how-to-prevent-it/>

1. Is servlet multithreaded?
2. How then objects are shared between threads?
3. What difference is between create statement and prepare statement?

<http://www.journaldev.com/2489/jdbc-statement-vs-preparedstatement-sql-injection-example>

1. How consumer producer works?
2. How many types of scans avaialable for DB?

Netcracker:

1. How to write factory pattern?
2. What is difference between String s = “abc”; and String s = new String(“abc”)?
3. What is immutable?
4. How to write Immutable class with date?

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

<http://howtodoinjava.com/core-java/related-concepts/how-to-make-a-java-class-immutable/>

1. Where does string objects and sting values reside?
2. What is enum?
3. What is difference between enum and enumeration?
4. What is scope of singleton?
5. How can you break singleton?
6. How server sets up session?
7. How to save session?
8. What is hierarchical query?
9. How to use hints in sql?
10. Write select query to get employee\_name and manager\_name?

|  |  |  |
| --- | --- | --- |
| emp\_id | emp\_name | manager\_emp\_id |
| 1 | Ram | 3 |
| 2 | Shyam | 3 |
| 3 | Ravi | 7 |
| 4 | Pritam | 7 |
| 5 | Laxman | 4 |
| 6 | Pradeep | 4 |
| 7 | Sunil | 7 |
| 8 | Raghu | 3 |

1. Write a program consists of two threads printing as mentioned below.

1-t1

2-t2

3-t1

4-t2

5-t1

1. What is explain?
2. How to run explain in dev?
3. How you analyze explain?