**JAVA INTERVIEW**

* Threads, Executor Service, Scheduler Service, Multi Thread
* Collections: Array, List, Compactor, Linked List, Queue, Stack
* Java Exception Handling: Check Exception, Unchecked Exception

3. OOPS

4. Design Pattern

5. Spring boot, Annotation, hibernate

6. SQL

7. Sort Algorithm

8. Insert Algorithm

9. [https://javarevisited.blogspot.com/2011/04/top-10-java-serialization-interview.html#axzz7RaOLA1mV](https://javarevisited.blogspot.com/2011/04/top-10-java-serialization-interview.html" \l "axzz7RaOLA1mV)

10. [https://javarevisited.blogspot.com/2014/07/top-50-java-multithreading-interview-questions-answers.html#axzz7RaOLA1mV](https://javarevisited.blogspot.com/2014/07/top-50-java-multithreading-interview-questions-answers.html" \l "axzz7RaOLA1mV)

11. [https://javarevisited.blogspot.com/2013/12/inter-thread-communication-in-java-wait-notify-example.html#axzz7RaOLA1mV](https://javarevisited.blogspot.com/2013/12/inter-thread-communication-in-java-wait-notify-example.html" \l "axzz7RaOLA1mV)

**12. Why wait, notify, and notifyAll are not inside the thread class?**

What is the difference between Comparator and Comparable in Java?

**What is a race condition in Java?**

IllegalMonitorStateException

**ReentrantLock**

**[https://javarevisited.blogspot.com/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html#axzz7RaOLA1mV](https://javarevisited.blogspot.com/2013/03/how-to-create-immutable-class-object-java-example-tutorial.html" \l "axzz7RaOLA1mV)**

**volatile and atomic variables in Java**

## **What is a Wrapper Class in Java?**

Write code to remove elements from Array List while iterating?

1. Rest API
2. Rest API vs Kafka
3. Rest API protocol
4. System Design
5. String s1 = “a” + “b” + “c”  
   s1+= “abc”; How many objects will be created ?
6. Immutable class share a List Object ?

Solution: Either share a copy of List or Collection.imutable(List)

1. Java Stop the World Garbage Collection?
2. Component vs Service?
3. @SpringBootApplication (@Configuration, @ComponentScan, @EnableAutoScan)
4. Singleton Pattern
5. Hashmap
6. Swap a string without third variable

**19-06-2022**

1. Methods in Object Class?
2. @inject in Springboot
3. Finalize method use?
4. Class uses the clonable and serialble interface?
5. What are functional interface?
6. After What size hashmap linked list convert into binary tree?
7. Yield?
8. Array get 2nd greater number?
9. Default and protected access ?
10. Why Operational Interface?
11. Wait and sleep ?

**Mindtree:**

1. **Spring boot related questions like types of bean.**
2. **spring boot life cycle**
3. **Bean life cycle in Java Spring**
4. **Do you know how many types of scope we can give for bean.**
5. **Life Of Beans**
6. **What is the default scope of the bean.**
7. core java questions with focus on hash map.
8. SQL Queries, Spring framework, REST API
9. Architecture of your project
10. Static in Java
11. Exception handling
12. private in Java
13. AOP in Spring
14. transient, persistent in Hibernate
15. finally, try, catch
16. map, set, queue
17. hashset
18. comparable and comparator
19. How do you triage/trace OutOfMemoryException.
20. Why do we need to go with Spring.
21. How can we inject beans in Spring.
22. Difference b/w BeanFactory and ApplicationContext.
23. Can you explain about the design patterns used in your Spring Project.

**Threads**

1.Init by the following way:

1. Extending thread

2. Implement Runnable

2. Life Cycle Of Thread.

3. Java Thread Priorities

4. The difference between wait and sleep and yield in Java

5. Java Executor Service

6. Semaphore

7. LinkedBlockingQueue

8. ConcurrentMap

9. Java Lock

**OOPS**

1. Coupling
2. Cohesion
3. Association
4. Aggregation
5. Composition

**Can we change the return type of method to subclass while overriding?** (answer)  
Yes, you can, but only from Java 5 onward. This feature is known as covariant method overriding and it was introduced in JDK 5 release. This is immensely helpful if original method return super-class e.g. clone() method return java.lang.Object. By using this, you can directly return the actual type, preventing client-side type casting of the result.

**Can we override a method which throws runtime exception without throws clause?** ([answer](http://javarevisited.blogspot.sg/2011/12/method-overloading-vs-method-overriding.html))  
Yes, there is no restriction on unchecked exception while overriding. On the other hand, in the case of checked exception, an overriding exception cannot throw a checked exception which comes higher in type hierarchy e.g. if original method is throwing IOException than overriding method cannot throw java.lang.Exception or java.lang.Throwable.

**Can we override a non-static method as static in Java?** (answer)  
Yes, you can override the non-static method in Java, no problem on them but it should not be private or final

**What is the default method of Java 8?**

**abstract class**

**What is an interface**

**Can we make a class both final and abstract at the same time?**