Java Theory

[**Priority Queue**](https://www.journaldev.com/1642/java-priority-queue-priorityqueue-example)**:**

Priority Queue: An unbounded priority queue based on a priority heap. The elements of the priority queue are ordered according to their natural ordering, or by a Comparator provided at queue construction time, depending on which constructor is used. A priority queue does not permit null elements. A priority queue relying on natural ordering also does not permit insertion of non-comparable objects (doing so may result in ClassCastException).

X-Frame-Options: DENY

preflight - OPTIONS call

HashMap, TreeMap, Hashset Implementations, [ConcurrentHashMap](https://medium.com/@itsromiljain/curious-case-of-concurrenthashmap-90249632d335) (Imp)

Links to Refer (Programming Questions):

<https://www.geeksforgeeks.org/matrix-chain-multiplication-dp-8/>

Links to Refer (General)

<https://www.youtube.com/watch?v=deG25y_r6OY> (RabbitMq)

<https://www.differencebetween.com/difference-between-physical-and-vs-virtual-memory/>

<https://www.quora.com/What-is-the-difference-between-Program-Files-and-Program-Files-x86>

<https://scotthelme.co.uk/a-new-security-header-referrer-policy/>

<https://stackoverflow.com/questions/10636611/how-does-access-control-allow-origin-header-work>

<https://stackify.com/java-performance-tuning/>

<https://www.baeldung.com/>

<https://www.geeksforgeeks.org/object-level-class-level-lock-java/> (class lock is required when we required synchronized static block)

<https://www.geeksforgeeks.org/comparison-yield-join-sleep-java/>

<https://www.geeksforgeeks.org/differences-between-wait-and-join-methods-in-java/>

Design Links:

<https://www.vertabelo.com/blog/technical-articles/a-database-model-for-a-movie-theater-reservation-system>

<https://prismoskills.appspot.com/lessons/System_Design_and_Big_Data/Chapter_07_-_Designing_Google_Maps.jsp>

<http://massivetechinterview.blogspot.com/2015/07/design-chess-game-using-oo-principles.html>

Spring Projects:

<https://spring.io/guides/gs/testing-restdocs/>

<https://spring.io/guides/gs/gateway/>

<https://spring.io/guides/gs/rest-service-cors/>

<https://spring.io/guides/gs/caching/>

<https://spring.io/understanding/CORS>

<https://spring.io/guides/gs/securing-web/> - spring security securing web application

<https://spring.io/guides/gs/circuit-breaker/>

<https://docs.spring.io/spring/docs/current/spring-framework-reference/integration.html#cache>

Things to do before Interview:

1. Revise all these practice questions
2. Revise all the design questions
3. Revise Java Theory
4. Revise SQL queries
5. Revise Project related technology questions (SpringBoot, RabbitMq, Microservices, Mongo)
6. Revise the features that I have worked in all my previous projects
7. Go through some of the behavioral questions

Things to do on weekends:

1. Practice design questions
2. Revise all the things mentioned above

Things to do on Weekdays:

1. Solve as many questions as possible
2. Read theory
3. If don’t feel like doing 1/2, try solving some design questions

Behavioral Questions:

1. Why are planning to leave the company
2. What did u like the most in the present company
3. What did u not like the most in the present company
4. Any situation where you had an argument with your manager on designing something and how did you convince him
5. Feature/Project that you feel you should not have done the way that it was done
6. Most challenging feature/project that you have worked on
7. 3 plus & minus points about you

[Java](https://www.journaldev.com/java-interview-questions) [Theory](https://www.javatpoint.com/corejava-interview-questions): [Link1](https://intellipaat.com/interview-question/java-interview-questions/), [Link2](https://www.guru99.com/java-interview-questions-answers.html)

1. Class Loader, JVM, Java Memory [Model](https://www.quora.com/How-many-types-of-memory-areas-are-allocated-by-JVM), Garbage Collection and related algorithms, String Immutability, Immutable class, [Singleton](https://www.geeksforgeeks.org/singleton-design-pattern/) class, Error vs [Exception](https://www.geeksforgeeks.org/errors-v-s-exceptions-in-java/)
2. Set, HashMap, TreeSet implementations
3. Collections Framework, Sorting complexities, when to use what, interface vs abstract
4. Java8, Java9 updates, streams, default methods in interface
5. Thread concepts (volatile, sleep, yield, join, notify, synchronization, thread life cycle, deadlock, lock types), semaphore, mutex vs semaphore, Executor Service
6. Immutable class, SingleTon,
7. Design Patterns, consistent hashing

10 Common Design Questions ([here](https://www.evernote.com/client/web?login=true&paywallUrlName=WEB-INF%2Fjsp%2Fredesign%2Fglobal%2Flayout.jsp#?anb=true&n=1fb605f7-d935-4d66-a97e-050555b997e2&s=s703&)):

1. Vending Machine (State Pattern)

<https://www.youtube.com/watch?v=VIzWKPRG9lo>

<https://www.youtube.com/watch?v=MGEx35FjBuo>

1. Score Card (Subscriber/Publisher Pattern)
2. Online Gaming (Chess, Snakes&Ladder etc)
3. Elevator/Parking System, Scheduler design
4. Messaging FrameWork (Whatsapp, Messenger)
5. Social Networking Site (Facebook)
6. Ticket Booking App (Oyo, Redbus, IRCTC)
7. Storage Application (Instagram)
8. Tax booking Apps (Ola, Uber)
9. Snakes And [Ladders](http://www.cvc.uab.es/shared/teach/a21291/temes/object_oriented_design/materials_adicionals/ladders_and_snakes/ladders_and_snakes.pdf)

String Immutability: [here](https://stackoverflow.com/questions/22397861/why-is-string-immutable-in-java)

**String Constant Pool** ... If string is mutable, changing the string with one reference will lead to the wrong value for the other references.

**Security**

String is widely used as parameter for many java classes, e.g. network connection, opening files, etc. Were String not immutable, a connection or file would be changed and lead to serious security threat. ...

Other Links:

<https://www.educative.io>

<https://www.linkedin.com/pulse/average-googler-four-weeks-study-plan-milad-naseri/>

<https://www.baeldung.com/java-microbenchmark-harness> - Java Bench Marking

<https://sites.google.com/site/steveyegge2/five-essential-phone-screen-questions>

http://steve-yegge.blogspot.com/2008/03/get-that-job-at-google.html?m=1

Design Pattern Concepts with Examples:

<https://springframework.guru/gang-of-four-design-patterns/>

Open Questions:

1. How does java support multi-threading ??
2. What are L1, L2 caches
3. Pinterest Engineering [Blocks](https://medium.com/@Pinterest_Engineering/sharding-pinterest-how-we-scaled-our-mysql-fleet-3f341e96ca6f)

Trivial Concepts:

1. The local variables are not initialized to any default value, neither primitives nor object references.
2. super() is implicitly invoked by the compiler if no super() or this() is included explicitly within the derived class constructor. Therefore, in this case, The Person class constructor is called first and then the Employee class constructor is called.
3. this() and super() must be the first statement in the class constructor, hence we cannot call both inside the constructor
4. Yes, we can change the scope of the overridden method in the subclass. However, we must notice that we cannot decrease the accessibility of the method. The following point must be taken care of while changing the accessibility of the method.
5. A final variable, not initialized at the time of declaration, is known as the final blank variable. We can't initialize the final blank variable directly. Instead, we have to initialize it by using the class constructor. It is useful in the case when the user has some data which must not be changed by others

6. IOC container is responsible to:

* create the instance
* configure the instance, and
* assemble the dependencies
* Makes the Collections unmodifiable
* Collections.unmodifiableCollection(Collection c)Collections.unmodifiableMap(Map m)
* Collections.unmodifiableList(List l)
* Collections.unmodifiableSet(Set s)

1. Java 8. Lambda Expressions, Interface Default and Static Methods, Method Reference, Parameters Name, Optional Streams, Concurrency.
2. Path – path to run time executiuon, classpath – path to .class files
3. Class vs Object

JMS Implementation - [here](https://dzone.com/articles/java-message-service-jms)

XSS (Cross site scripting) - [here](https://pentest-tools.com/blog/xss-attacks-practical-scenarios/)

CSRF (Cross Site Request Forgery)

CORS

Fority Security:

Top Questions to solve for Interviews:

<https://www.interviewbit.com/coding-problems/>

<https://www.geeksforgeeks.org/must-do-coding-questions-for-companies-like-amazon-microsoft-adobe/>

[**Java 8**](https://www.javatpoint.com/java-8-features)**:**

Lambda’s

Streams

Optional Class

Method Reference

Functional Interface

Static methods in interface

LocalDateTime, LocalDate

forEach - [here](https://www.mkyong.com/java8/java-8-foreach-examples/) (difference between collection.stream().forEach() vs collection.forEach() - [here](https://stackoverflow.com/questions/23218874/what-is-difference-between-collection-stream-foreach-and-collection-foreach))

Collectors class

[Java 9:](https://www.javatpoint.com/java-9-features)

Interface with private methods

Try with resource enhancements

JShell

Stream API Improvement (takeWhile, dropWhile, ofNullable, iterate)

Go Through [Questions](https://www.wisdomjobs.com/e-university/mockito-interview-questions.html) on [Mockito](https://dzone.com/articles/java-unit-testing-interview) as well (Mockito vs Spy, [Mocks vs Stubs](https://www.martinfowler.com/articles/mocksArentStubs.html), MockitoRunner vs SpringRunner)

Orchestration pattern vs Choreography Patter - [here](https://stackoverflow.com/questions/4127241/orchestration-vs-choreography)