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](https://blog.morethancode.dev/killing-cors-preflight-requests-on-a-react-spa/) - OPTIONS call (sent during cross domain call or request has custom headers). This might add to the latency of the actual result. One option is we can cache the preflight call response but again browsers allow this only for 600 seconds. For details go through the link

HashMap, TreeMap (implements SortedMap), Hashset(implements SortedSet) 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/ Design Challenges?
6. Most challenging feature/project that you have worked on
7. 3 plus & minus points about you
8. A time when you could not deliver a task in time.
9. Most challenging task that u have worked on till now
10. An Initiative that u have taken
11. Design patterns that you have used in your application?
12. Any questions to the interviewer?
13. What is the toughest project you have ever worked on? Why was it tough? What did you do to solve the problem? How did you design it? What choices you made and why?

[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 ([applications](https://stackoverflow.com/questions/3769607/why-do-we-need-immutable-class) of immutable classes), [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](https://dzone.com/articles/when-to-use-abstract-class-and-intreface)
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

**Consistent** [**Hashing**](https://www.toptal.com/big-data/consistent-hashing)**:**

Consistent Hashing is a distributed hashing scheme that operates independently of the number of servers or objects in a distributed hash table by assigning them a position on an abstract circle, or hash ring. This allows servers and objects to scale without affecting the overall system.

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 (Cross Origin Resource Sharing)

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 (one abstract method, any number of default/object methods) - [here](https://www.geeksforgeeks.org/functional-interfaces-java/)

Static and default 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

When we extend an interface that contains a default method, we can perform following,

1. Not override the default method and will inherit the default method.
2. Override the default method similar to other methods we override in sub class..
3. Re declare default method as abstract, which force subclass to override it.

[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)

[**Spring Dependency Injection, IOC**](https://www.careercup.com/question?id=5766735711961088)**:**

Spring helps in creating loosely coupled application because of Dependency Injection.

In spring objects define their associations (dependencies) and do not worry about how to get those dependencies. Now it is the responsibility of Spring to provide the required dependencies for creating objects. Inversion of Control (IOC) and Dependency Injection (DI) are used interchangeably. IOC is achieved through DI**. DI is the process of providing the dependencies and IOC is the end result of DI. By DI the responsibility of creating objects is shifted from our application code to Spring container hence the phenomenon is called IOC**.

Dependency Injection can be done by setter injection, constructor injection.

Setter Injection vs Constructor Injection (Which to use when ??) - [here](https://javarevisited.blogspot.com/2012/11/difference-between-setter-injection-vs-constructor-injection-spring-framework.html#axzz5kbmHoHCI)

Can multithreading be implemented on a single processor system? - [here](https://stackoverflow.com/questions/16116952/can-multithreading-be-implemented-on-a-single-processor-system) [here](https://spring.io/blog/2007/07/11/setter-injection-versus-constructor-injection-and-the-use-of-required/)

No Public constructor is allowed inside a enum class, all constructors have to be private

[Aspect Oriented Programming](https://howtodoinjava.com/spring-aop-tutorial/):

Advice: Around, Before, After, AfterReturn, AfterThrowing

JoinPoint: Place of Action/Point of Execution

PointCut: Expression that matches the joint points

Objective Oriented Programming vs Functional Programming ??

Is java script an objective oriented programming language or functional programming language ?

**Interface vs Abstract Classes:**

When to use Abstract:

1. There are some common fields and more number of common methods
2. Have common partial implementations/behavior
3. Establish relation between related objects

When to use Interface:

1. Leverage the advantage of multiple inheritance
2. Establish relation between unrelated classes (cloneable, serilazable etc.,)

[OSI](https://www.youtube.com/watch?v=vv4y_uOneC0) (Open Systems Interconnection) Model - [here](https://www.youtube.com/watch?v=vv4y_uOneC0)

1. Physical Layer
2. Data Link Layer
3. Network Layer
4. Transport Layer
5. Session Layer
6. Presentation Layer
7. Application Layer

Each layer is a group of protocols

Web Application vs Enterprise Application

Web Server vs Application Server

jar vs war vs ear

Web Application: Contains only web related technologies like JSP’s, Servlets, html, css, js

Enterprise Application: Contains all related J2EE technologies

Web Server: Server used to deploy applications. Ex: Tomcat

Application Server: Server used to deploy Enterprise applications Ex: Web logic

Jar: Contains Java Class Files

War: Contain web technology related files

Ear: contains all type of files