**Interview Questions**

Difference between Spring and Spring Boot?﻿The primary diff by Spring and Spring Boot is in its purpose and functionality each provides. Spring framework provides wide range of﻿ tools and libraries to create complex enterprise level application. Spring requires a lot of manual configurations to be performed in﻿ the project as we need to manually configure external server, and dependencies manually and manage version conflicts manually. In﻿ case of Spring Boot it is a project build on top of Spring framework that simplifies the need of manual configurations of external﻿servers and other dependencies. Spring Boot comes with embedded server and starter dependencies that provides pre-configured﻿ dependencies. Use of spring boot reduces development time significantly; more developer can focus on business logic more than spending time﻿ in manual configurations.﻿

What is Dependency Injection?﻿DI is a design patter in Spring boot that allows an object to receive it's depended-on from an external source rather than creating﻿ itself. That helps to achieve loose coupling. There are diff types of injection like constructor injection, field injection and﻿ setter injection.﻿

What is Inversion of Control?﻿In traditional programming object itself manages its dependencies by creating or fetching them. With introduction of dependency﻿ injection this control is inverted and the dependencies are injected by container in spring boot. This process is called Inversion of﻿control.﻿

Inversion of Control vs Dependency Injection?﻿Inversion of control (IOC) is a principle that inverts the control of object creation whereas Dependency Injection is a pattern that﻿ implements IOC by injecting the﻿dependencies.﻿

What happen when we start spring boot application?﻿When the Springboot app starts off the main method in the application is executed, which calls SpringApplication.run(). Spring﻿boot auto configures based of the class path, dependency and spring boot stater is initialized with all the beans and it will start the﻿embedded server and start accepting the﻿ requests.﻿

What is Bean? Explain about Bean lifecycle? Explain about different types of Bean?﻿Spring bean is an object managed, instantiated and assembled by Spring IOC container. A bean life cycle starts by reading bean﻿ definition i.e. xml or annotations to define all the beans and their properties. Later bean is instantiated using one of its constructor﻿. After instantiation spring performs dependency injection on the bean as it will add the dependencies that the bean depends on.﻿, we have defined a bean post processor which will help in customization of beans either before initialization that will be executed﻿ once the IOC container is shutdown all the beans will be destroyed which will be the last stage in the life cycle of a bean. The different﻿ types of beans are singleton, prototype, session, global and request. These are considered different scopes on the scope of the﻿ bean.﻿

How to change port number in spring boot application?﻿server.port=value in [application.pro](application.pro#application.pro)perties﻿file﻿

Which server spring boot uses internally? How to change that?﻿Spring boot uses Tomcat as its default internal server and it also supports Jetty, Undertow and undertow as its embedded servers. To﻿ change the default server we have to exclude the tomcat starter dependency from spring boot started web dependency and include﻿start using it.﻿

How to implement paging and sorting in Spring Boot?﻿To use paging and sorting we need our repositories to implement PagingAndSortingRepository and we need to pass﻿ paging information from our controller, we would require details on what value the data should be sorted, page size and page﻿number. With these values we need to create a Pageable object which will be passed in our repository methods to fetch the﻿ details.﻿

What is @SpringBootApplication?﻿It is one of the main annotations in spring boot, it comprises of 3 annotations @Configuration, @ComponentScan, and @EnableAutoConfiguration.﻿

Explain about different annotations like @Service, @Controller, @RestController, @Entity, @Component, @Repository, @Autowired, @Required﻿, @Primary, @Async, @Transactional, @SpringBootApplication.﻿@Service- Used to define service layer classes.﻿@RestController- Marks a class as rest controller which marks methods which return json responses.

@Controller- used to mark a class as﻿ entity which will be a representation of a table in DB. @Component- Marks a class as a spring bean.

@Repository- Helps in defining a﻿ class as repository which will help to connect to the database queries.

@Autowired- used to inject a dependency to a class.

@Required-﻿used to mark a dependency as required if it’s not present during startup will fail. @Primary- If there are 2 beans which are created﻿ implementing an interface we can use this annotation to pick one of the as primary as it will be used during @Autowired.

@Async- used at﻿ method level which helps in executing the method in non blocking manner. @Transactional- to make a specific field in entity class not﻿be saved to db.

@Component-This means the package where the annotation is specified to create the﻿ beans.﻿

Difference between @Controller and @RestController?﻿@Controller is used in Spring MVC project to return view and @RestController is a combination of @Controller and @ResponseBody﻿annotation where it is used to return restful web service responses.﻿

What is @Configuration?﻿Annotation added at class level that creates beans for spring boot﻿ application.﻿

What does autowiring mean? Types of autowiring?﻿Adding dependency to a class is called autowiring, there are constructor, setter and field based﻿ injections.﻿

What is ApplicationContext?﻿Application context helps spring to store all its bean created. It helps in achieving inversion of control in﻿ spring.﻿

Difference between @Bean vs @Component?﻿@Bean annotation is used at method level that returns a bean and @Component is used at class level to make the class as a﻿ bean.﻿

What is @Qualifier annotation?﻿@Qualifier helps in avoiding ambiguity when there are multiple beans available from a same﻿type.﻿

What are uses of Filter and ControllerAdvice?﻿Filters: advice is used in exception handling, when we want to commonly apply exception handling across all the available﻿ controllers we use controller﻿ advice﻿

Explain about different HTTP Methods?﻿get- to get the resource,

put- does insert/update on the availability of resource,

patch- does partial update of resource,

delete to﻿ delete resource,

post - creates the﻿ resource.﻿

What is Idempotency? Explain about Idempotent methods?﻿When same api methods calls doesn't change the state of resource further its called idempotent, get patch and put are idempotent, except﻿ post that creates resource per﻿call.﻿

What is @Transactional annotation? Understand its uses?﻿@Transactional annotation is used to achieve atomicity, where the process either completes successfully or all the actions are reversed﻿ in case of any exception.﻿

What is actuator? Explain about few endpoints of actuator?﻿actuator provides a set of production ready features for monitoring and managing your application. Some of the end points are health,﻿info, beans﻿What is Profiling in Spring boot?﻿It we want diff properties to be set as part of environment we use profiling in active profile in application properties change from environment﻿ to environment. We can configure diff properties for each env by enabling﻿ profiling.﻿

Difference between CrudRepository vs JpaRepository?﻿Crud repository is basic repo in spring data jpa, its extended by paging and sorting repo which in turn extended by jpa repository﻿

Explain about different starter dependencies of Spring boot?﻿Spring boot comes with diff starters that helps in adding all the dependencies related to that functionality to our project. ex﻿ spring-boot-starter-web dependency comes with all the other related dependency that helps in creating a web project. spring-boot-jpa﻿dependency comes with all the dependencies that helps in connecting to﻿DB.﻿

How to connect multiple databases?﻿

If I want to disable configuration class in spring boot application, how to do?﻿We can use exclude either by properties file to exclude a configuration class or we can use @ConditionOnProperty whose value is false to disable a﻿ configuration﻿

Difference between @RequestParam and @PathVariable?﻿@RequestParam helps to get request params as part of request url in rest controller and @PathVariable helps to get specific value out at﻿URI.﻿

Process of Global error handling / Exception Handler and @ControllerAdvice?﻿@ControllerAdvice is used to define a global exception handler that catches exceptions thrown by any controller. Each method is﻿annotated with @ExceptionHandler specifying the type of exception to﻿handle.﻿"Entity Class Definition having One-to-Many and Many-To-One mapping.﻿1. Emp, Employee and Department.﻿2. ""All annotations used inside this may ask further of any related annotations."""﻿[https://sathishkumaranjayaraman.com/conversational-learning-with-spring-boot-jpa-and-hibernate-onetomanytoone/](https://sathishkumaranjayaraman.com/conversational-learning-with-spring-boot-jpa-and-hibernate-onetomanytoone/#https://sathishkumaranjayaraman.com/conversational-learning-with-spring-boot-jpa-and-hibernate-onetomanytoone/)﻿"Create a crud Rest API service for fetching the attendance report based on the role\_type and input\_date.﻿1. Input date - pass as path variable﻿2. role\_type - pass as request parameter﻿3. If the input date is null return report of previous day"﻿[https://geekyhacker.com/spring-boot-crud-operations/](https://geekyhacker.com/spring-boot-crud-operations/#https://geekyhacker.com/spring-boot-crud-operations/)﻿Using of JPA inside repository (examples in above question) same link as﻿above.﻿Using of @Query in above question﻿example:﻿If we have another table named Employee how to write a query to fetch the data using native query? Explain using @Query or @NativeQuery? If have﻿ideal we have to use native query in spring boot to create a custom query. This will be enabled by @Query annotation and sql query and﻿parameters values from﻿How to fetch values from [application.pro](application.pro#application.pro)perties ? List few use case ? value annotation or @ConfigurationProperties for more details refer﻿[https://www.baeldung.com/properties-with-spring](https://www.baeldung.com/properties-with-spring#https://www.baeldung.com/properties-with-spring)﻿Where to we add db connection properties in spring boot app? After Adding db we need to add any other logic to process that or automation﻿added?﻿In properties file we need to add db connection details and after that we need to create a repository to connect to﻿db.﻿Tool used for compiling spring boot project (maven) what happen in maven compile command. Steps involved. maven and gradle are the﻿tools generally used in maven compile target folder is removed and new target folder with compiled class files are﻿created.﻿Native Query Usage example helps in writing db specific﻿query﻿Explain the Spring MVC Architecture?﻿Difference between @Controller and @RestController?﻿List all annotations used in your Spring MVC Project?﻿Explain usage of @RequestParam and @PathVariable in Spring MVC?﻿What is @Qualifier Annotation used for in Spring MVC?﻿What is @SpringBootApplication annotation used for in Spring MVC?﻿Explain the Stereotype annotations used in Spring MVC?﻿How do you create a custom annotation in Spring?" Already answered in previous questions and for creating custom annotation please follow steps here﻿[https://medium.com/the-fresh-writes/create-your-own-custom-annotation-in-spring-boot-b66cf4527688](https://medium.com/the-fresh-writes/create-your-own-custom-annotation-in-spring-boot-b66cf4527688#https://medium.com/the-fresh-writes/create-your-own-custom-annotation-in-spring-boot-b66cf4527688)﻿"What is difference between JPA repository and CRUD repository?﻿Explain the difference between Hibernate and Spring JPA ?﻿What is Spring Batch ? Explain the Use case of it?﻿What is Spring Security ?﻿How Spring JPA handle transactions ?﻿Explain the commonly used HTTP methods (and like GET and POST) " already covered in previous﻿questions.﻿"Need to generate attend report (daily, dev, test) and report table contains team name, employee, presentation, date﻿1. write a complete spring boot application controller and repository.﻿2. report will generate everyday 11pm using scheduler." already covered in previous questions.﻿Microservices﻿==================

What is Microservice Architecture and why it is used?﻿Difference between Monolithic Architecture vs Microservice Architecture? Advantages and disadvantages of both?﻿In which scenario we should use Microservice or Monolithic architecture?﻿Basic understanding of Docker and K8s?﻿What is Distributed Transaction?﻿What is Service Registry and service discovery?﻿What is API Gateway? What are benefits of api gateway?﻿How to make synchronous vs async communication between them?﻿What is Circuit Breaker pattern?﻿What is Distributed Tracing and Distributed Logging?﻿What is availability , performance and reliability?﻿Understand about messaging and caching, how we can use them in microservices?﻿Understand about different libraries like Eureka, Hystrix, Ribbon, Feign, RestTemplate, Spring Cloud.﻿What are Microservices and how they differ from monolithic architecture?﻿How microservices communicate with each other?﻿What is role of Apache Kafka and its core components?﻿What are advantages of Microservices?﻿What are the limitations of Microservices?﻿List the 12 factor App of Microservices Application?﻿

JAVA﻿====﻿Why we have static keyword in Java? "Static keyword is used for memory management. It belongs to the class rather than object of the﻿class.﻿Static keyword can be used with variables, methods and classes."﻿Why we create static methods and what are benefits of that? Method with static keyword is called static method. It can be accessed with﻿the class name. Static method can access the static variables without using the object of the class.﻿Explain about checked and unchecked exceptions with examples? "Compile time exceptions are called checked Exception. Example - IO﻿Exception﻿RunTime Exceptions are called Unchecked Exception. Example - NullPointerException,ArrayIndexOutOfBoundsException"﻿Can we have try with multiple catch and how actually exception handling works, explain with code snippet? "Yes we can have try with﻿multiple catch blocks.﻿try{﻿int num[]=new int[5];﻿System.out.println(num[10]);﻿}﻿catch(ArrayIndexOutOfBoundsException e){﻿System.out.println(e);﻿}﻿catch(Exception e){﻿System.out.println();﻿}"﻿How HashMap internally identify whether object is unique or not? HashMap identifies an object as unique if its hashCode() is different. If its﻿hashCode() is same its equals() returns false.﻿How to create a synchronized arraylist in java? "Collections.synchronizedList() is used to synchronize the list. List should be passed as﻿an argument to synchronizedList(). To iterate the synchronizedList, we need to manually synchronize the﻿list using synchronized(list){}"﻿CopyOnWriteArrayList is also used to create thread safe variant of ArrayList.﻿Why interfaces are there in java explain about different types of interfaces? "Interfaces are mainly introduced to perform multiple﻿inheritance. It is used to achieve Abstraction.﻿Marker interface - Interface with single abstract method. Eg - Predicate, UnaryOperator﻿Functional Interface - Interface with no abstract methods. Eg - Serializable, Cloneable﻿Normal interface - Interface with one or more abstract methods."﻿Difference between Callable and Runnable interface? "Both runnable and callable are interface.﻿Runnable - It contains run() method does'nt return any value. It is mainly used with Thread/Executor Service﻿Callable - It contains call() which throws Exception. It is commonly used with ExecutorService/Future"﻿What is the use of yield() method in java? yield() is a static method from Thread class. It signals the task scheduler that current﻿running thread is willing to pause and let other threads to execute.﻿How to read a file from a given directory using java? To read a file from given directory classes like File, FileReader,﻿BufferedReader are used.﻿What is concurrentHashMap? Explain about internal working of concurrentHashMap? ConcurrentHashMap is the thread safe version of HashMap﻿which allows multiple threads to read and write without locking the entire map.﻿Explain about internal working of HashMap? HashMap uses hashing to store key and value pairs in bucket. resolving collisions using linked﻿list or tree (introducing haster 1.8 onwards). insertions and deletions.﻿How to create immutable objects and class in java? "To make class to be immutable class should be declared as final. All the data members﻿of the class should be declared as private and should not contain setter methods.﻿The objects should not be modified once created (shallow immutable objects). The object created for immutable class is called﻿immutable objects. Immutable objects are thread safe."﻿Explain about Singleton Design Pattern with code snippet? How to break and overcome? "Singleton Design pattern ensures the class has only﻿one instance and provides global access to it. It is achieved by making constructor private, and having static method﻿getInstance().﻿How many types of final class in java? "When a class is declared with final keyword, it is final class. Final class cannot be﻿extended/inherited."﻿String is immutable, explain? "In String, value changes cannot be extended.﻿final class String{}"﻿Explain about fail-fast and fail-safe iterators? "Fail-fast iterator - It throws ConcurrentModificationException whenever there is a﻿structural change (adding/removing elements) from the list/map/set collection while iterating. Ex - Iterators of ArrayList, LinkedList,﻿HashMap﻿Fail-safe iterator - It will not throw ConcurrentModificationException whenever there is a structural change in the collection while﻿iterating. It is because it iterates over the copy of original collection. Ex - Iterators of CopyOnWriteArrayList, ConcurrentHashMap"﻿How object class methods internally works? "All the JAVA classes are either directly/indirectly derived from Object class. Object class﻿has few methods like toString(),hashCode(),equals(),getClass()"﻿Object class methods can be used in all the java classes?﻿What is difference between OOPS and Functional Programming? "In Functional Programming, we build the program using functions. To﻿encapsulate, Haskell language﻿In Object Oriented Programming Building using classes and objects.it has many concepts like encapsulation, abstraction, polymorphism etc.﻿Ex - Java"﻿Explain about JVM? "Java Virtual Machine provides the environment to run the Java application.﻿.java files created by developer is converted in to .class file after compilation. JVM then reads the bytecode(.class files) and﻿converts it into native machine code using interpreter or JIT(Just In Time) compiler."﻿JVM main feature is memory management, ensures efficient use of resources.﻿What is Inheritance, Association and Composition? "Inheritance is a concept in OOPS which allows one class to inherit the﻿features(methods and variables ) from other class. It represents ""IS-A"" relationship. Sub/Inheritance allows us to have code﻿reusability, Method Overriding."﻿Association - Association is a concept in OOPS which describes the relationship between two independent classes. Association is of two﻿types.﻿Unidirectional - One class is aware of the other class and interact with it. Reverse is not possible.﻿Bidirectional - Both the classes are aware of each other and interact with each other.﻿Composition - Composition is a type of association where one class contains another class. The contained object lifecycle depends on the﻿container object.﻿Contained object cannot exist independently of the container. If container object is destroyed contained object is also destroyed.﻿What is Aggregation?﻿Aggregation - In Aggregation, one class contains the reference of other class. It represents "HAS-A" relationship. Here the﻿contained object can exist independently of container object.﻿What is abstract class? How to create abstract class? abstract class with abstract keyword is called abstract class. Abstract class cannot﻿be instantiated. Abstract class can have both abstract methods and concrete methods.﻿Abstract class will have only method declaration for abstract methods, its implementation will be provided in the subclasses. Abstract class﻿can have constructors and fields.﻿Abstract methods can be called using the subclass objects as its definition will be given in the subclasses.﻿Difference between abstract class and interface? "Abstract class - can have abstract and concrete methods.﻿It can have non-final variables.﻿It can have constructors﻿Interface - It can have only abstract methods, post java 8 it can have default methods.﻿It can have only static final variables﻿It will not have constructors"﻿Why String is immutable in java and what are benefits of that? "String is immutable, because the values cannot be changed once they are﻿created.﻿Benefits:﻿Strings are stored in string pool to save memory. If multiple variables hold the same string literal, it points to same object.﻿Since strings are immutable, it is thread safe."﻿Difference between String Vs StringBuffer Vs StringBuilder? "String - String is immutable, content cannot be changed once created.﻿Threadsafe﻿StringBuffer - StringBuffer is mutable and threadsafe - performance will be slower compared to stringbuilder due to synchronization﻿StringBuilder - It is mutable and not threadsafe"﻿ Difference between final, finally and finalize? "final" keyword can be used with variable(value cannot be changed), method(cannot be overridden) and class(cannot be inherited).﻿finally - The code inside finally block will always run. It is mainly used to close the DB connection.﻿finalize - It is used for garbage collection.﻿throw vs throws in java? "throw" is used within the block to throw an exception in exception handling.﻿throws - It is used in the method signature to throw multiple exceptions.﻿Explain about Java Collection Framework? "Java collection framework is a set of interfaces and classes in java.util package that provides the algorithm and data structure for storing and manipulating the collection of objects.﻿diff between the Hashtable and Concurrent HashMap? HashTable - HashMap is a class under traditional collection. It is not thread safe. If one thread is iterating, that time if another thread tries to modify a HashMap, then other thread is already iterating it, HashMap can have one null key and multiple null values. It is used for single threaded applications.﻿Concurrent HashMap - Concurrent HashMap is a class under concurrent collection. It is Thread Safe. It will not throw any ConcurrentModificationException as it is synchronized. It will not allow any null key and null value.﻿ArrayList vs LinkedList in Java? "ArrayList - It uses dynamic array to store the element. ArrayList is used whenever we want to access the element using index. Low memory usage.﻿LinkedList - It used doubly linked list to store the element. LinkedList is use when we want to perform insertion/deletion in the middle of the list. High memory usage. Each element in LinkedList is stored in a Node so it contains actual data and a reference to previous node and next node.﻿How TreeMap internally works? "TreeMap uses the data structure of TreeMap. It contains NavigableMap, and SortedMap Interface which keeps the element sorted and in ascending order.﻿ It is generally used when we need sorted and unique data. It is slower than hashset as it is sorted"﻿Comparator vs Comparable in Java? "Comparable - It belongs to java.lang package. It comparator actual class is modified. It compareTo() method.It provides single sorting sequence﻿Comparator - It belongs to java.util package.In comparable actual class is not modified.It has compare() method.It provides multiple sorting sequence﻿ HashMap vs Hashtable? "HashMap is not synchronized, not thread safe. Whereas Hashtable is synchronized and thread safe. HashMap allows one null key and multiple null values. Whereas Hashtable will not allow null key and null values. HashMap is faster as it allow multiple threads to access it."﻿Explain about OOPs Concept?﻿ Class: "Class is a blueprint that defines the attributes and methods that the object created from the class will have."﻿Object: "Object is the instance of the class. They store the actual values of the attributes and call the methods."﻿ Inheritance: "Inheritance is a concept in OOPS which allows one class to inherit the features(methods and variables) from another class. It represents "IS-A" relationship."﻿Abstraction: "It is a concept of hiding the implementation details and showing only the essential features to the user. It is achieved through abstract classes or interfaces."﻿ Encapsulation: "Encapsulation is the concept of hiding the internal data details and restricting the direct access to the object. It is achieved through making the data members of the class as private and providing getter and setter to access them."﻿Polymorphism: "Polymorphism is multiple forms. It can be achieved through method overloading(Compile time Polymorphism) and method overriding (Runtime Polymorphism)."﻿Comparator interface in Java with example? "It belongs to java.util package. In comparator actual class is not modified. It has compare() method. It provides multiple sorting sequence"﻿HashMap remove() method in java? "HashMap.remove() is used to remove the remove the key - value mapping in the map.It takes either key as an argument. If key does not exist it throws nullpointer"﻿What is difference between synchronized and concurrent collection? "Synchronized Collection - It is a part of java.util package. It is thread safe. It allows only one thread to access/modify the collection at a time. Performance is low as synchronization blocks all threads except one. It requires explicit synchronization for iteration to avoid ConcurrentModificationException.Ex: Vector, HashTable, Collections.synchronizedList(), Collections.synchronizedSet(), Collections.synchronizedMap()﻿Concurrent Collection - It is a part of java.util.concurrent package. It is thread safe provide better performance compared to synchronized collection."﻿Ex: ConcurrentHashMap, CopyOnWriteArrayList, CopyOnWriteArraySet﻿How to remove duplicate elements from arrayList? "To remove duplicates from arrayList, we can use HashSet. HashSet does'nt preserve the order of insertion"﻿Convert List to array in Java? Using stream API or toArray()﻿What is Marker Interface explain with example? "Interface that does not contain any methods/fields. Eg: Serializable, Cloneable"﻿ Can we overload/override static method? "Static methods can be overloaded, but cannot be overridden as static methods belong to the class rather than object. If the derived class has same method as base class, it is considered method hiding and not method overriding."﻿ Exception Handling: "Exception handling in java is used to handle the runtime errors for the normal flow of the program. It prevents the program from crashing."﻿ Thread, Thread creation, Thread States, Synchronization, Deadlocks, Volatile Variable, Synchronized Block and Synchronized Method, Thread Dump: "Thread - Thread is a light weight process which can run concurrently with other threads sharing the same memory.﻿ Thread states - New, Runnable, Blocked, Waiting, Timed\_Waiting, Terminated﻿ Thread Creation - Thread can be created either by extending Thread class/implementing Runnable interface.﻿ Synchronization - Only one thread can access the shared resource at a time.﻿ Deadlocks - Deadlock occurs in multithreading when two or more threads are waiting for each other to release resources.﻿ Volatile Variable - Volatile is mainly used in multithreading. If volatile keyword is used threads will read the data from main memory rather than the local copy of the changes made by one thread will be visible to all the threads immediately.﻿ Synchronized Block - Locks only specific block of code﻿ Synchronized Method - Locks the entire method﻿ Thread Dump - Thread dump is a snapshot of all the threads in Java application which shows their current status, stack traces, lock information and used for identifying deadlocks."﻿ How to Thread Implement﻿ Write a Java code that shows Singleton Design Pattern﻿ What is Thread Pool? Explain briefly and notifyAll() "Thread Pool is a collection of pre created threads that can be reused to perform multiple tasks. It is maintained by a Scheduler framework.﻿ notify() and notifyAll() methods used in interthread communication. notify() is used to wake up only one thread whereas notifyAll() is used to wakeup all the waiting threads."﻿ Briefly difference between Thread.start() and Thread.run() "Thread.start() is used to start a new thread and invoke the run() in that thread. start() method directly calls the run() in the current thread."﻿ Executor Framework: "It is an API in Java that simplifies the creation and management of threads by decoupling tasks.﻿ Callable - It is similar to Runnable, but it can return result and throw exception.﻿ Future - It represents the result of computation. It allows to retrieve the result once the task completes.﻿ ScheduledThreadPool Executor - It is used to schedule a task to run after a delay / run periodically after fixed interval.﻿ ThreadLocal variable - It provides the thread local storage"﻿ Difference between ArrayList and LinkedList in Java? "ArrayList - It uses Dynamic array to store the element. ArrayList is used whenever we want to access the element using index. Low memory usage.﻿ LinkedList - It uses doubly linked list to store the elements. LinkedList is use when we want to perform insertion/deletion in the middle of the list. High memory usage. Each element in LinkedList is stored in a Node so it contains actual data and a reference to previous node and next node."﻿ Explain the hierarchy of Collection Framework in Java? "Collection -> List, Set, Queue﻿ Map -> HashMap, TreeMap, HashTable"﻿ What is fail-safe n fail-fast iterators? "FailFast Iterator - It throws ConcurrentModificationException whenever there is a structural change(adding/removing a element from the list) in the collection while iterating. Ex - Iterators of ArrayList, LinkedList, HashMap﻿ FailSafe Iterator - It will not throw ConcurrentModificationException whenever there is a structural change in the collection while iterating. It is because it iterates over the copy of original collection. Ex - Iterators of CopyOnWriteArrayList, ConcurrentHashMap"﻿ Explain the internal working of HashMap in Java? "HashMap uses hashing to store key and value pairs in bucket. Resolving collisions using linked lists or trees ensuring faster lookups, insertions and deletions"﻿ What updates where on HashMap from Java 8? "Before Java 8 HashMap uses only linked list to store the entries. After Java 8 it uses either linked list or tree to store the entries. Also after Java8, HashMap has functional methods like forEach(), compute(), merge() etc."﻿ Difference between HashSet and TreeSet? "HashSet is unordered and faster while TreeSet is sorted and slower"﻿ What is difference between List, Set and Map in Java? "List - List is collection of ordered elements. It allows duplicate. Ex -ArrayList, LinkedList﻿ Set - Set is a collection of unique elements. It does not allow duplicates. Ex - HashSet, TreeSet﻿ Map - Map is a collection of key value pairs. Ex - HashMap, TreeMap"﻿ Explain the get() operation in HashMap and how will you avoid collision in HashMap ? "The get() operation in HashMap hashes the key, locates the corresponding bucket and searches the key to return the value. While collision can be minimized by a good hash function"﻿ Difference between Array and ArrayList? "Array - Arrays are of fixed size and it can store elements of any type. It has no built in methods.﻿ ArrayList - ArrayList is of dynamic size can grow/shrink as needed. It can store objects of any type. It has built in methods like add(), remove()"﻿ How to synchronize a collection in Java? "Collections can be synchronized using Collections.synchronizedList(), Collections.synchronizedSet(), Collections.synchronizedMap(), ConcurrentCollection"﻿

Explain Proxy Pattern?﻿

What are the different design pattern used in your project?﻿

What is factory design pattern?﻿

What is dependency injection in Java? "Proxy Pattern - A structural design pattern is way to use a placeholder object to control access to another object. Instead of directly interacting with the main object, client talks to the proxy which then manages the interaction and provides additional functionality such as controlling the access, lazy initialization etc.﻿ Factory Method Design Pattern - Define an interface for creating an object, but let subclasses decide which class to instantiate.﻿ Dependency Injection is a design pattern that allows a class to receive its dependencies from external source(Eg - Spring reducing coupling."﻿SOLID Principles in Java:﻿ Single Responsibility Principle(SRP) - It makes the code cleaner and easier to maintain﻿ Open-Closed Principle (OCP) - Enables adding new functionality with minimal changes﻿ Liskov Substitution Principle (LSP) - Ensures the integrity of inheritance hierarchies﻿ Interface Segregation Principle (ISP) - Reduces unnecessary dependencies﻿ Dependency Inversion Principle (DIP) - Decouples high level and low level modules for better scalability"

CODING:﻿

There is a list, contain duplicate numbers, to find out duplicate numbers and count how many times these duplicate numbers are occurring using stream API.﻿

"There is an employee class having fields (id-int, name string, gender-string, yearofjoining-int, salary double). Take 3 records in the list using the values for employee objects.﻿

Need to find highest paid employee name,﻿

who have joined after 2018, how many male and female employees using "java stream API"."﻿

Write a java program repeated values by using java 8 "Sachin", "Virat", "Rohit", "Saurav"﻿

Write a program to reverse a String.﻿

Write a program to count the frequency of each character in a string?﻿

Write a program to remove duplicate word from a list of Strings?﻿

Write a program to find even numbers and double them using lambda in java 8?﻿

Write program to find first non-repeated character in a string?﻿

Write Program to filter Employees with containing specific character for e.g. 'a'?﻿

Write Program to filter Employees with Salary greater than 50000?﻿

Write Program to extract domain names from e-mail addresses?﻿

Write program to convert list of strings to their lengths?﻿

Write Program to square each number in a list?﻿

"From the list of string find the count of the occurrence using stream api﻿

e.g. - [a,b,c,a,b,a,d,e,f]﻿

output - {a=3, b=2, c=1, d=1, e=1, f=1} "﻿

"Have a class student with﻿

int Student\_Id﻿

String Student\_Name﻿

int Joining\_Year﻿

String Gender﻿

List<Int> marks.﻿

Find the name of the student with highest mark joined after 2018 using stream API"﻿

"Student contains (name id, name, standard, grade (occurs need to write code in java8)"﻿

a). Name of student as list from 8th standard whose grade is 'A'﻿

b). When count of student from 10th standard who have same same grade 'B'."﻿

"Write the java program to get the employees whose department having IT.﻿

Find the odd number which contains the 1230 and make out of the list." - JAVA7/JAVA﻿

String reverse program using recursion ﻿

Find second largest element in array ﻿

Java8:﻿

Explain latest features of Java 8? Lambda expressions, Stream API, Functional Interfaces, Default methods, forEach(), Method references﻿

What is the use of Optional in Java? "Optional class is used to avoid NullPointerException"﻿

What changes have been made in HashMap in Java8? "Before Java 8 HashMap uses only linked list to store the entries. After Java 8 it uses either linked list or tree to store the entries. Also after Java8, HashMap has functional methods like forEach(), compute(), merge() etc."﻿

What is Stream API? "Stream API is a feature introduced in Java 8. It is used for performing multiple operations like filtering, mapping and sorting."﻿

What is the benefit of using Stream API? "Stream does not change the original data.﻿

It provides parallel processing.﻿

Stream provides lot of methods to perform intermediate and terminate operation."﻿

Example:﻿

Intermediate operation - filter(), map(),sorted()﻿

Terminate Operation - collect(), reduce()﻿

What improvements have been made in Date Time API? After Java 8, Date Time API is thread safe and immutable. It directly supports the addition of certain period/duration.﻿

What is default method why it has been introduced in Java8? Before Java 8, interfaces have only abstract methods and the classes which are implementing the interface need to provide the method definition. So if the new method is to be added in the interface its definition need to be provided in the class which implements the interface. To overcome this default methods was introduced in Java 8 where the interface can have default methods and it will have the method definition.﻿

What is Functional Interface? "Functional Interface can have only one abstract method.It can have any number of default methods. Explain about different built in Functional Interfaces? Supplier<T>(supplies values on demand. get()﻿

Consumer <T>(performs action like printing/processing elements - accept()﻿

Predicate<T>(performs operations like filtering elements based on condition. test()﻿

Function<T,R>(performs operations for transforming data from one type to another. apply()"﻿

What is Lambda Expression? What are its usage? "Lambda expression is a short block of code which takes the parameters and returns the value. Lambda expression can be used only with Functional Interfaces.﻿

Lambda expression reduces the number of lines of code"﻿

What is UnaryOperator, Supplier, Consumer, Predicate? "UnaryOperator - takes input and output type will be same. It is used to perform operations like incrementing/squaring a number.﻿

Supplier<T> - Generates/supplies values on demand. get()﻿

Consumer<T> - Used for operations like printing/processing elements - accept()﻿

Predicate<T> - Filters elements based on condition. test()"﻿

Does lambda contains return statement? "If the lambda body has single expression, return is not mandatory. If the lambda body is a block and enclosed within {}, return is used."﻿

Java 8 Stream findFirst() vs findAny()? "findFirst() - It always returns the first element from the stream. Performance will be slow for parallel streams. Use findAny when the order of element matters.﻿

findAny() - It can return any element from the stream. Optimized for parallel streams. It is unordered."﻿

What is removeIf() method in Java? "removeIf(Predicate predicate) method is used to remove a element from the list remove elements that matches the given predicate."﻿

What are lambda expressions? "Lambda expression is a short block of code which takes the parameter and returns the value. Lambda expression can be used only with Functional Interfaces.﻿

Lambda expression reduces the number of lines of code."﻿

What is functional interface in Java 8? "Functional Interface can have only one abstract method.It can have any number of default methods."﻿

What is predicate interface and how can we use lambda expressions in Java 8? "Predicate is a built in functional interface. It is used to test a condition and returns true/false﻿

test()"﻿

Predicate<Integer> number -> n>10;﻿

System.out.println(number.test(10));﻿

How does method reference different from lambda expressions? "Both method reference and lambda expression is used to reduce the no of lines of code﻿

Lambda expression - (parameters) -> expression. It is used to define custom logic﻿

Method reference - ClassName::methodName. It relies the existing methods"﻿

What is difference between filter, reduce and map method of Stream API in Java 8? "filter() - it is used to select/filter the elements of a stream based on a condition.﻿

map() - It is used to transform elements of a stream by applying a function.﻿

reduce() - It is used to reduce the elements of a stream to a single value.﻿

forEach() - It is used to traverse the elements of a stream to perform some functional. It takes an input and produces an output"﻿

How does reduce difference from collect in Java 8 streams? "reduce() - It is used to perform reduction operation and reduce the stream elements to a single value. It is mainly used to find maximum, minimum, sum, average.﻿

collect() - It is used to transform the stream to a collection like List,Set,Map."﻿

What is difference between Iterable, Iterator, and Stream? "Iterable - represents a sequence of elements that can be iterated over. It does not have methods for filtering or processing.﻿

Iterator - used for iterating over a collection. It provides methods like hasNext() and next().﻿

Stream - built on top of Iterable. It can be used for both sequential and parallel processing. It has many intermediate functions like filter(), map()."﻿

What is difference between Comparable and Comparator? "Comparable - It has compareTo(Object o) method. It is implemented in the class itself.﻿

Comparator - It has compare(Object o1, Object o2) method. It is implemented in a separate class."﻿

What is the difference between Map and FlatMap in Java 8? "Map - It produces one output for one input. It produces a stream of elements.﻿

FlatMap - It produces a flattened stream by applying one-to-many transformation and then flattening the resulting streams into one stream."﻿

What is the difference between regular interface and functional interface in Java 8? "Regular interface can have multiple abstract methods and default methods which are introduced in Java 8.﻿

Functional interface can have only one abstract method. It supports lambda expressions"﻿

KAFKA﻿

What is Kafka, Why do we use it? "Kafka is primarily used to build real-time streaming data pipelines and applications that adapt to the data streams. It combines messaging, storage, and stream processing to allow storage and analysis of both historical and real-time data."﻿

Explain some key features of Kafka? The key attribute of Kafka is its ability to deliver a large number of messages with minimal latency and without fault. Some of the scalable features of Kafka helps us achieve these key features.﻿

[https://www.scaler.com/topics/kafka/features-of-kafka/](https://www.scaler.com/topics/kafka/features-of-kafka/#https://www.scaler.com/topics/kafka/features-of-kafka/)﻿

What are the Topics and Partitions in Kafka? Topics are logical groupings of streams of data within Kafka. They act as message queues where producers publish data and consumers retrieve it. Partitions are the basic unit of data storage and distribution within Kafka topics.﻿

What is Broker or Kafka Broker? A Kafka broker is a single instance or node in the Kafka system. It is in charge of receiving incoming messages, storing them, and serving them to consumers.﻿

What is meant by fault tolerance? How does Kafka handle failures? Kafka streams (build on fault-tolerant capabilities integrated natively in Apache Kafka. Kafka partitions are highly available and replicated so when stream data is persisted to Kafka it is available even if some of the application fails and needs to re-process it.﻿

Explain some use cases of Kafka? "Used in applications where data retrieval in near to real time. Like tracking applications, google map etc"﻿

What is Offset? Kafka offset represents the order of messages inside a partition from the beginning of that partition. This numerical value helps Kafka keep track of progress within a partition. It's crucial for Kafka to work horizontally while staying fault-tolerant.﻿

What is a zookeeper and how does it work? Zookeeper is used in distributed system for service synchronization and as a naming registry. When working with Apache Kafka, Zookeeper is primarily used to track the status of nodes in the Kafka cluster and maintain a list of Kafka topics and messages.﻿

How does Kafka help in developing microservice-based applications? Kafka enables the production, consumption, and storage of streams of records in real time. These streams can be used to build event-driven microservices that can react to changes in the system in a decoupled and asynchronous manner.﻿

SQL﻿

What is indexing? Indexes are special lookup tables that can be used by the database search engine to speed up data retrieval. An index is simply a reference to data in a table. A database index is similar to the index in the back of a journal. It cannot be viewed by the users and they are just used to speed up the database access﻿

Writing SQL Query﻿

What is view? A database view in SQL is like having virtual tables containing a single query and its result. The result is usually a virtual table with rows and columns, just like the actual tables. We can either pass a query to store all the rows and columns of a table or part of it.﻿

How to delete and truncate delete operation performs the delete action row by row where truncate clears all the data in one go. Which is better in performing﻿

How to call stored proc from java application﻿

[https://www.geekforgeeks.org/how-to-call-stored-procedure-and-stored-procedures-using-jdbc/](https://www.geekforgeeks.org/how-to-call-stored-procedure-and-stored-procedures-using-jdbc/#https://www.geekforgeeks.org/how-to-call-stored-procedure-and-stored-procedures-using-jdbc/)﻿

How join query you can ask, second highest salary etc.