**Interview Questions**

**JAVA**

* How does java program run
* What is random() class and how can we use it
* What is serializable and why random class implements serializable
* Diff between comparable and comparator
* What is static keyword. Why do we need that
* Can static variables/ methods be accessed through className
* Can non-static variables be accessed directly inside static methods
* Where static variables/methods memory is allocated
* What is static block. When is it gets executed
* About static classes. Can non-static class variable accessed by static method?...... ([Why We Use Static Class in Java - Javatpoint](https://www.javatpoint.com/why-we-use-static-class-in-java))
* Why/when do we need to give final keyword for variables, methods, classes (question asked in tcs)
* Why/ when do we use abstract keyword(abstraction). Can we use it for classes, methods, variables
* Can abstract classes have normal methods. Can normal classes have abstract methods.
* Can we create an object for abstract classes.
* Can we give final and abstract for same class or method. If not why
* What are singleton, transient, scoped in java
* What is lazy initialization and eager initialization, how to achieve both in java
* What is Singleton class in java. Why we need that.
* What is polymorphism. When do overriding of methods happen (run-time or compile-time polymorphism)
* What is inheritance. Can we extend two classes at same time (multiple inheritance). Why cant we do multiple inheritance. ([Inheritance in Java - Javatpoint](https://www.javatpoint.com/inheritance-in-java))
* What is encapsulation. Why do have to do that.
* why do we need getter and setter methods in java when we can give variables as public( not private)
* What are data structures in java
* Difference between && and & operator? Compare
* using Boolean && and Boolean &
* What is wrong with Array, why List
* Comparable vs comparator. Their uses
* What is serialization, serializable interface
* Exception handling. Why do we use.
* Diff between exception and error. Hierarchy diagram of exception [Exception Handling in Java | Java Exceptions - javatpoint](https://www.javatpoint.com/exception-handling-in-java)
* When we catch exception in catch block will it terminate the program? NO
* How to create userdefined exception?(extends runtimeException, constructor with super(message)) When we throw new user defined will it terminate the program? YES
* When we throw new unchecked exception will it terminate the process? YES
* When we throw new checked exception will it terminate the process? NO
* Diff between throw and throws keywords (question asked in tcs)
* Garbage collection in java. About Finalize and gc methods
* Garbage collection happens in which memory heap or stack
* Java memory allocation. What and all gets created in heap and stack
* Enum (Enum values or Enum instances are public, [static](http://javarevisited.blogspot.sg/2011/11/static-keyword-method-variable-java.html), and [final](http://java67.blogspot.sg/2012/09/what-is-blank-final-variable-in-java.html) in Java. you can declare methods in Java Enum and you can even [override and overload](http://javarevisited.blogspot.sg/2011/12/method-overloading-vs-method-overriding.html) them. Enum in Java doesn't have a public [constructor](http://javarevisited.blogspot.sg/2012/01/what-is-constructor-overloading-in-java.html). Enum cannot extend a class in Java. Every Enum in Java extends from java.lang.Enum implicitly. Enum in Java can implement an interface and override methods. In fact, Java Enum implicitly implements a [Comparable](http://javarevisited.blogspot.sg/2011/06/comparator-and-comparable-in-java.html) interface and overrides the [compareTo()](http://javarevisited.blogspot.sg/2011/11/how-to-override-compareto-method-in.html) method.)
* Regular expression in java
* Diff between default construc and no args constructor
* **Difference between** \*\*PATH\*\* **and** \*\*Classpath\*\* **in Java?**( PATH is used by the operating system while Classpath is used by JVM to locate Java binary)
* **Can we override static method in Java?** (No, because overriding resolves at runtime while static method call is resolved at compile time.)
* difference between Enumeration and iterator (Iterator has a remove() method while Enumeration doesn't. by using Iterator we can manipulate the objects by adding and removing the objects from collection)
* **Can a** \*\*Serializable\*\* **class contains a non-serializable field in Java?(** Yes, but you need to make it either static or transient.)
* **Difference between** \*\*java.util.Date\*\* **and** \*\*java.sql.Date\*\* **in Java? (**contains both date and time while later contains only date part.)
* Why string is immutable?(String pool, thread safe, cache hash value, class loading)
* How to create immutable class? (make class final, variables instances as private and final, No setter methods, initialize all variables in constructor, return cloned object from getter will not impact original object)
* Can we have static method in interface? (yes)
* Can u declare constructor final? (NO)
* Why default methods in interface implemented (For adding lambda expression in Java 8, JDK needs to add  methods(such as **foreach**) to List or collections Interface, but if you add this method to these interface, it will break millions lines of code as class which implements the interface, need to implement all its methods.)
* https://java2blog.com/interface-default-methods-in-java-8/
* Why multiple interface will not work with interfaces having default methods ()
* What is dynamic method dispatch in java? (Run time polymorphism. It is a mechanism by which a call to an overridden method is resolved at run time)
* How to use try with catch loop(in workspace)
* Does finally block execute always ? (No, all cases exept system.exit(0))
* Fail-fast and Fail-safe iterator?(workspace)
* <https://dev.to/javinpaul/50-core-java-interview-questions-with-answers-for-programmers-3bc>
* <https://java2blog.com/java-interview-questions-for-2-years-experience/>
* <https://javahungry.blogspot.com/2020/10/java-interview-questions-for-2-years-experience.html>
* Check these three first
* [10 Java Concepts to know before appearing for technical interview (intechgrity.com)](https://www.intechgrity.com/10-java-concepts-must-know-appearing-for-technical-interviews/)
* <https://javarevisited.blogspot.com/2015/10/133-java-interview-questions-answers-from-last-5-years.html#axzz7BX34s2Yl>
* <http://www.instanceofjava.com/p/interview-questions.html>
* <https://www.edureka.co/blog/interview-questions/java-interview-questions/>
* <https://javahungry.blogspot.com/2020/10/java-interview-questions-for-2-years-experience.html>
* <https://java2blog.com/java-interview-questions-for-2-years-experience/>
* <https://dev.to/javinpaul/50-core-java-interview-questions-with-answers-for-programmers-3bc>
* <https://www.monsterindia.com/career-advice/java-interview-questions-and-answers-for-2-to-3-years-experience-8382.html>

**Collections framework:**

* Why collections . Advantages
* When are we using extends and implements keywords and why
* Which collection extends/implements which interface/class - hieararchy diagram of collection , Map , Object
* What are the common methods we use in collections& common exceptions we can get
* What is thread safety & synchronization under collections, classes and methods
* How to create a thread and how to start and run that thread
* What is runnable interface
* What is AtomicInteger / AtomicBoolean …. and why do we need those
* What is arrayList/ Advantages.
* How to create an ArrayList. How to add and get a value from arrayList
* What are generics and why do we have to add it. Is is mandatory to give generics
* What is virtual capacity and physical capacity in ArrayList. What is the default virtualCapacity value and can we change it?
* What are wrapper classes. Can we give normal int instead of wrapper class for generics
* What is autoboxing
* In how many ways u can print ArrayList (typical for loop, foreach, stream lambda, iterator)
* What is foreach loop and how to use that
* What are streams and advantages. Why they introduced streams
* What is lambda expression. How to write that
* Can we pass another ArrayList while declaring a ArrayList. How to do that
* How to add two arrays?. Add element at nth index?. Clear the Array?
* How to do Array cloning. Use of .contains, .indexOf, .lastIndexOf, .remove, .removeIf, .retainAll
* How to create new ArrayList using sublist of another Array
* How to change ArrayList to an Array
* How to remove duplicates from ArrayList using LinkedHashSet and using streams
* What is Synchronized Array List and how to create one
* When can we use synchronized keyword (explicit synchronization) for add, remove, fetching/update should we have to use explicit synchronized.
* What is copyOnWriteArrayList (class or interface?). How to write one
* LinkedList implements from?
* What are the three types of LinkedList. What are the internal working of Singly, Doubly and Circular LinkedList?
* Can we create our own LinkedList and how to create
* Why we need LinkedList and Advantages of LinkedList
* How to iterate LinkedList and using forEach. Use of add, AddAll, addFirst, addLast, remove, removeAll, removeFirst, removeLast, clear
* How to print in reverse order ( use of descendingIterator)
* How to sort the LinkedList directly (use of Collections.sort)
* What is Map interface? How to create a Map
* Different ways of Creating Map – Using HashMap, Using static way(static HashMap),immutable map with single entry(singleton Map), Creating two dimension array of String and collecting in the form of Map
* Diff ways of Creating of Map using .MapOf (in java jdk 9)
* What is HashMap and why we need it. Duplicates, Null values, key-value, Thread safe?
* HashMap hierarchy which class it extends to. And How is it connected to Map interface
* How does HashMap works internally?
* How to create a HashMap? How to iterate it?
* Can we use direct iterator with HashMap? Use of .keySet? How to loop it and get values?
* How to get pair of key,value from iterator? Use of .entrySet? How to loop it and get values?
* Using forEach to get key value pairs?. Use of .remove?
* How can you compare two HashMaps (use of .equals method)
* How to compare HashMaps if they have same keys?
* How to find extra keys when comparing two HashMaps?
* How to compare HashMaps if they have same values? (when duplicates are allowed and not allowed)
* How to convert HashMap to ArrayList
* What is LinkedHashMap
* Why do we need LinkedHashMap and diff between LinkedHashMap and HashMap
* Is HashMap synchronized? What is synchronized HashMap? How can we create synchronized HashMap
* What is concurrentHashMap? How it works internally (working of threads)? How to create concurrentHashMap?
* What is HashTable? Advantages of HashTable? How to create HashTable?
* How HashMap Works internally?
* What is treeMap? Advantages of TreeMap? How to create TreeMap?
* How TreeMap internally works?
* How treeMap sorting happens when we have Strings in key place and when we have numbers in key place.
* How to sort in reverse order of TreeMap? (use of Comparator.reverseOrder)
* **How does**\*\*ConcurrentHashMap\*\***achieve scalability? (**by dividing the map into segments and only locking during the write operation.**)**
* **Which two methods should you override while putting the custom object as key in HashMap(**You need to override [hashcode and equals method](https://www.java2blog.com/hashcode-and-equals-method-in-java/) in custom class while putting objects of custom class in HashMap.**)**

[How HashSet Works Internally In Java? (javaconceptoftheday.com)](https://javaconceptoftheday.com/how-hashset-works-internally-in-java/#:~:text=How%20HashSet%20Works%20Internally%20In%20Java%3F%20HashSet%20uses,are%20stored%20as%20keys%20of%20this%20HashMap%20object.)

[LinkedHashSet in java basics and internal working - KSCodes](http://www.kscodes.com/java/linkedhashset-in-java/#:~:text=LinkedHashSet%20in%20java%20basics%20and%20internal%20working%201,insertion%20order%20of%20elements%204%20Is%20non%20Synchronized.)

[How TreeSet Works Internally in Java - Javatpoint](https://www.javatpoint.com/how-treeset-works-internally-in-java#:~:text=Let%27s%20understand%20the%20internal%20working%20of%20the%20TreeSet.,on%20it.%20It%27s%20working%20is%20similar%20to%20HashSet.)

* Doubts while coding
* How to add two objects as a list directly like(List.of(course,course1))
* Use of copy method. Diff between shallow and depth copy (question asked in tcs)
* Which one u prefer among collections in diff situations and why (question asked in tcs)
* Why only for set duplicates r not allowed why in list they r allowed (question asked in tcs)
* How to make a method to not be accessed by other classes (question asked in tcs)

**MultiThreading:**

* What is thread and multithreading
* Diff between process and thread, multiProcessess and multithreading (Is the application we create is a process)
* What are Thread class and runnable interface. Diff between them
* In how many ways you can create a thread. (creating thread using runnable and Thread. Which is better way)
* What is thread lifecycle. What are diff states in that.
* What are important thread methods
* Diff between run() and start() methods. Why & How to use getName(), setName(), sleep(), stop(), wait(), join(), getId(), interrupt()
* How main thread is linked to threads we created. What is work of main thread.
* What is synchronized keyword. Where do we give that. What happens when we use that.
* What is interThread communication. Check producer, consumer example
* What happens when we call wait() and notify() methods.
* What is Daemon Thread. Example of daemon thread
* How many maximum of threads can be created
* Volatile and atomic integer in java concurrency
* What is ThreadPoolExecutor and how to use it
* **Difference between Race condition and Deadlock in Java?**

**Java 8 features:**

* What Streams? How to use and write them? Why do we need them?
* Functional interfaces? Why, how and where to use them?
* Filter, reduce , predicate?
* Method references?
* Lambda expressions?
* Default keyword?

**Java 9 and higher versions:**

* Check workspace

**Spring**

* Why do we need spring framework? Advantages of Spring?
* What is dependency injection? Why do we need that?
* What is use of Spring container? Where do spring creates/ stores objects?
* Is spring container inside JVM? When does it gets created?
* What are the types of scopes for object creation?
* Explain singleton and prototype scopes? Bydefault Spring uses which Scope? how can we change the scope?
* Use of @Component? Use of @Scope?
* Use of @AutoWired? Why do we need Autowired annotation?
* When does object gets created in Spring container? (new Laptop() , @Component, @AutoWired)
* Advantages of dependency injection?
* What is maven? Why do we need it? Uses of maven?
* What is pom.xml. Can we have multiple pom.xml’s in a project?
* How can we do packaging when we have multiple pom.xml’s?
* How can we access other module classes when we have multiple modules in maven project?
* What is groupId and artifactId for project / any dependencies
* What is Scope in depencies of pom.xml? types of scopes?

**Spring MVC**

* Why Spring mvc? Advantages
* What is DIspatcherServlet? Why do we need web.xml? What and all will be there in that?
* What is @Controller and @RequestMapping annotations ? Why do we need that?
* How to redirect from controller method to another jsp page?
* How to send data from jsp to controller method? And how to fetch from java method to jsp page?
* What is expression language? How to use it in jsp?
* How can we configure java class instead of using web.xml and servlet.xml?
* What is @Configuration and @ComponentScan annotations? Why do we use them?
* What is webInitializer class? Why do we need it? How to give configure class and mapping “/” in this class file?
* How can we avoid giving .jsp ? What is internalViewResolver? How to add it in configure class?
* What is @Bean annotation in configure class
* Use of @RequestParam annotation?
* Use of @EnableWebMvc?

**Spring boot**

* What is spring?
* What is dependency injection?
* Why SpringToolSuite?
* What is Autowired annotation and relation with dependency injection?
* Where do u put index.jsp and how to display it
* Where do u put index.jsp and how to display it controller class?
* How to make tomcat to recognize jsp files (use of tomcat-jasper dependency)
* How to use application.properties to connect to database?
* How to config index.jsp file in properties file if we change its location
* How to send data to client/jsp page & get data from client(using HttpRequest and HttpResponse)
* How to send/receive data to client using ModelAndView
* How to configure H2 database (adding h2 dependency) and how to enable it
* How to open/use h2 database through browser and check
* Use of entity and id annotation
* How to make an interface as repository. Use of Crud repository
* Use of annotations like @Controller, @RequestMapping, @ResponseBody, @RestController,@PathVariable, @RequestParam
* Inbuilt methods for saving objects and getting all data through object
* How to get details based on other variables using inbuilt methods
* Using custom methods with custom queries (use of Query annotation)
* Inbuilt methods like findAll, findById etc…
* Use of postman. How to send data from postman(use of @PostMapping). Getting data to postman(use of GetMapping)
* Update data from postman (use of @PostMapping). Delete data through postman (use of @DeleteMapping)
* Use of getOne inbuilt method
* What is the default format of data returning from controller. How to return in Json format.
* How to send data in xml format also. (use of Jackson xml dependency). Will adding this make data to be sent in both json and xml fomat
* How to restrict data format if it is going in both json and xml formats/ multiple formats (use of consumes and produces)
* How to restrict in Postman to accept/send data in particular data.(use of headers in postman)\
* What is Spring Rest Api?
* Can we do the basic operations without controller. (use of JpaRepository and @RepositoryRestResource) how to use these
* How to request for all operations(get,post,put,delete) from postman when using RestApi

**Spring JPA**

* Can we
* How to

**JPA**

**Hibernate**

* What is persistant
* what is hibernate and why should we use it
* dependency needs to be added to make hibernate work
* what is the hibernate xml file used and what should we give in that
* how to add database connection through hibernate xml file
* what is hibernate hbm file and what and all we give in that file (question asked in tcs)
* what are Service Registry and addAnnotatedClass in hibernate
* how will hibernate connects to database (configuration,sessionfactory,beginTransaction,session……)
* diff between properties update and create and how to give in hibernate xml file
* can we display sql queries generated by hibernate and how
* use of entity annotation
* use of id annotation, is it mandatory
* use of Transient annotation
* annotations & properties used to change table name and column name of a table
* use of annotation embeddable
* what are hibernate relations and annotations used
* use of annotations OneToOne , OneToMany , ManyToOne, ManyToMany
* Real-time example on oneToMany (question asked in tcs)
* can we get rid of creating separate table while using nested objects
* What is default fetch type in hibernate and how can we change it (what is fetchType EAGER)
* What is hibernate caching
* What are 1st level and 2nd level cache in hibernate
* Which level cache is there bydefault in hibernate.
* What are the dependencies, properties in xml file, annotations to add for 2nd level cache in hibernate.
* Which 3rd party library supports 2nd level cache
* What is 2nd level cache with query and how to achive that. Property to add for this
* What is hql and why we need hql
* Main diff between sql query and hql query
* How to write hql query and execute it
* While fetching the data from query we use uniqueResult() and list() what is the diff between them
* What is the return type when using uniqueResult() & list()
* How to write hql select queries in hql and what is the return type when getting uniqueResult() and list(). How to display the returned data
* How to pass parameters into a hql query
* What are native queries. How to use sql queries directly in hql
* How to fetch and display the native queries in hql.
* What is addEntity and why to use it while executing native queries
* How to fetch the data when fetching few columns in a table. Why/How to use setResultTransformer
* What are hibernateObject states
* Diff between transient, persistant, detach and remove states
* When we set a value to object after save(), will it get updated into database
* Diff between get() and load() while fetching data
* Which one is best when it comes to exception handling
* What are nameQuery and namedQueries annotations. Why do we need those

**Spring boot MicroServices**

**SQL**

https://www.youtube.com/watch?v=a3ngELoA9h0&list=PLtPFQl\_0853oNyMPC2Fd0K1uW9x41KQn6

**Scala**

* What is scala. Why should we learn that and use it
* Is scala works with java. Is scala functional programming or object oriented programming
* Diff between functional programming or object oriented programming
* Diference between classes and objects in scala. Where should we write main method. Can we write main method in class.
* How to define a variable and methods in scala. Also when it is returning something
* How to write loops( for loop, forEach, while ) in scala. Use of until & to in for loop
* What is match expression and how to write that
* How to do filtering from for loop. Use of yield keyword
* Can we use x++, x--in scala
* How to create List, set , Map in scala. How to retrieve values from them
* Inheritance in scala. Can we achieve multiple inheritance in scala. If not how to achieve that
* What are abstract class, traits in scala. Diff between abstract class & traits. Diff between java Interface and scala traits.
* Can we use implements keyword in scala
* What are Higher order functions
* Function composition
* Lambda function
* Anonymous function
* Function currying
* Partially applied function
* Implicit parameters
* Nested functions
* Call by value & Call by name
* Closures
* Tuples
* Options