***Technical Concepts Handbook***

***- Siddhant Patni***

***Mobile: +91-7588811796***

***Email: siddhant4patni@gmail.com***

Contents

[***Java*** 24](#_Toc195453780)

[**Java Fundamentals** 24](#_Toc195453781)

[ **Explain how to install java? Explain directories in it?** 24](#_Toc195453782)

[ **How to configure java in windows?** 25](#_Toc195453783)

[ **Explain Java Program execution?** 25](#_Toc195453784)

[ **Explain the byte stream in java?** 27](#_Toc195453785)

[ **What is Classloader? Different types of it?** 27](#_Toc195453786)

[ **Why is java more secure?** 27](#_Toc195453787)

[ **Explain public static void main (String[] args) method?** 27](#_Toc195453788)

[ **What is the marker interface?** 27](#_Toc195453789)

[ **What is serialization and deserialization?** 27](#_Toc195453790)

[ **What is the use of the transient keyword in serialization?** 27](#_Toc195453791)

[ **What is the final class?** 27](#_Toc195453792)

[ **Why is string immutable in java?** 27](#_Toc195453793)

[ **What is StringBuffer?** 28](#_Toc195453794)

[ **What is StringBuilder?** 28](#_Toc195453795)

[ **Difference between String, StringBuffer and StringBuilder?** 28](#_Toc195453796)

[ **What is the final keyword in java?** 28](#_Toc195453797)

[ **Where can we use the final keyword?** 28](#_Toc195453798)

[ **What will happen if I create the final class?** 28](#_Toc195453799)

[ **Difference between comparable and comparator?** 28](#_Toc195453800)

[ **What is static in java?** 28](#_Toc195453801)

[ **Explain Generics? What is the use of Generics?** 29](#_Toc195453802)

[ **How to find the second largest element from the list?** 29](#_Toc195453803)

[ **What is Singleton in java?** 29](#_Toc195453804)

[ **What will happen if I assign null to some object then will that be garbage collected?** 29](#_Toc195453805)

[ **When does the finalize method get called?** 29](#_Toc195453806)

[ **What is JDBC?** 29](#_Toc195453807)

[ **Consider we have Strings as String s1 = “Welcome”, String s2 = “Welcome” and String s3 = “WelcomeOne”. What does s1==s2 represent?** 29](#_Toc195453808)

[ **Explain String constant pool?** 30](#_Toc195453809)

[ **What is the difference between equals() and hashCode() method?** 30](#_Toc195453810)

[ **Does Java support Pointers?** 30](#_Toc195453811)

[ **Why doesn't java support pointers?** 30](#_Toc195453812)

[ **Explain which language you prefer in terms of security among C, C++, Java and Python?** 30](#_Toc195453813)

[ **Is Java 100% object oriented?** 30](#_Toc195453814)

[ **How can we make java 100% object oriented?** 30](#_Toc195453815)

[ **Can we create objects of static class?** 30](#_Toc195453816)

[ **What is the difference between ArrayList and LinkedList? Which is better?** 30](#_Toc195453817)

[ **What is Enumeration in java?** 30](#_Toc195453818)

[ **Which compiler is used by Java?** 30](#_Toc195453819)

[ **Where are the hashCode() and equals() methods defined in java?** 30](#_Toc195453820)

[ **How many design patterns are present in java?** 30](#_Toc195453821)

[ **Can we override the static method?** 30](#_Toc195453822)

[ **Explain access modifiers in java?** 30](#_Toc195453823)

[ **What is volatile in java? Where can we use it?** 30](#_Toc195453824)

[ **What is an idempotent method in java?** 31](#_Toc195453825)

[ **What is final, static and non-static (instance) in java?** 31](#_Toc195453826)

[ **How will I use two threads using a singleton design pattern?** 31](#_Toc195453827)

[ **How to implement thread safety in java using singleton?** 31](#_Toc195453828)

[ **Write an immutable class in Java?** 31](#_Toc195453829)

[ **Can we override the protected method?** 31](#_Toc195453830)

[ **What is marker interface in java? What is the purpose of the marker interface?** 31](#_Toc195453831)

[ **How can we ignore the finally block?** 31](#_Toc195453832)

[ **What is synchronous and asynchronous call?** 31](#_Toc195453833)

[ **Can we write the main method as private?** 31](#_Toc195453834)

[ **What is the use of static keywords in java?** 31](#_Toc195453835)

[ **What is inner class? Have you used inner class in your project?** 31](#_Toc195453836)

[ **What is the purpose of inner class?** 31](#_Toc195453837)

[ **What is Enum? Write a syntax for enum?** 31](#_Toc195453838)

[ **Difference between static and non-static in java?** 31](#_Toc195453839)

[ **How can we break the singleton pattern?** 31](#_Toc195453840)

[ **Difference between default and protected access modifier?** 31](#_Toc195453841)

[ **Why is string immutable?** 32](#_Toc195453842)

[ **What is an immutable class?** 32](#_Toc195453843)

[ **What is atomic integer in java?** 32](#_Toc195453844)

[ **How can you make a singleton class as thread safe?** 32](#_Toc195453845)

[ **Difference between compareTo and compare method?** 32](#_Toc195453846)

[ **What is the return type of compare method?** 32](#_Toc195453847)

[ **What is contract in equals and hashcode method?** 32](#_Toc195453848)

[ **Difference between Marker interface and Functional interface?** 32](#_Toc195453849)

[ **Difference between Abstract class and Interface? Explain with scenarios which one you will it be used with conditions?** 32](#_Toc195453850)

[ **What is the use of static and instance variables in java?** 32](#_Toc195453851)

[ **How will you access JNDI in java code?** 32](#_Toc195453852)

[ **Why is the main method in java static?** 32](#_Toc195453853)

[ **What is the use of the static keyword?** 32](#_Toc195453854)

[ **How many ways can we create beans in java?** 32](#_Toc195453855)

[ **Explain different types of calling beans?** 32](#_Toc195453856)

[ **What are the default methods?** 32](#_Toc195453857)

[ **Can I add one more new default method in my child class? Does it give any problems with my existing classes?** 33](#_Toc195453858)

[ **Difference between Comparable and Comparator interface?** 33](#_Toc195453859)

[ **In Java, which class is the base class of all the classes?** 33](#_Toc195453860)

[ **Why is the equals method required?** 33](#_Toc195453861)

[ **Is it possible to have the same hashcode for multiple objects?** 33](#_Toc195453862)

[ **Enlist a few predefined interfaces from Java 8?** 33](#_Toc195453863)

[ **If I have a 200 MB excel file so can java read that file?** 33](#_Toc195453864)

[ **What is immutability and mutability in java?** 33](#_Toc195453865)

[ **Custom implementation of Singleton beans?** 33](#_Toc195453866)

[ **What is the default scope of beans?** 33](#_Toc195453867)

[ **Difference between static and final?** 33](#_Toc195453868)

[ **Explain about String class in java?** 33](#_Toc195453869)

[ **What is a special feature of String?** 33](#_Toc195453870)

[ **Create an immutable class in java?** 33](#_Toc195453871)

[ **How to access Singleton class from another class?** 33](#_Toc195453872)

[ **Where will the Singleton class be stored in memory?** 33](#_Toc195453873)

[ **Difference between bucket level and segment level locking?** 34](#_Toc195453874)

[ **How is the static method bound with an object?** 34](#_Toc195453875)

[ **What is the use of the intern() method in String class?** 34](#_Toc195453876)

[**Java Memory Management** 35](#_Toc195453877)

[ **Explain JDK, JRE?** 35](#_Toc195453878)

[ **Explain JVM?** 35](#_Toc195453879)

[ **What are the different loaders in JVM?** 35](#_Toc195453880)

[ **Explain Java memory management?** 35](#_Toc195453881)

[ **What is Stack / Heap memory?** 35](#_Toc195453882)

[ **Explain memory management below? Int and Integer?** 35](#_Toc195453883)

[ **Where local variables are stored in memory?** 35](#_Toc195453884)

[ **Where objects will be stored in memory?** 35](#_Toc195453885)

[ **Explain Heap memory in java?** 35](#_Toc195453886)

[ **Difference between Stack and Heap?** 35](#_Toc195453887)

[ **What is Garbage Collection in java?** 35](#_Toc195453888)

[**OOPs** 36](#_Toc195453889)

[ **Explain OOPs concept?** 36](#_Toc195453890)

[ **What is an object?** 36](#_Toc195453891)

[ **Explain method in Object class? Explain 9 methods in Object class?** 36](#_Toc195453892)

[ **How many ways can we create objects of class in java?** 36](#_Toc195453893)

[ **What is Encapsulation? Explain with a real time example** 36](#_Toc195453894)

[ **How do we achieve encapsulation in Java?** 37](#_Toc195453895)

[ **What is abstract in java?** 38](#_Toc195453896)

[ **Explain Composition in Oops?** 39](#_Toc195453897)

[ **What is composition and aggregation?** 39](#_Toc195453898)

[ **Parent and child class methods and how to access those?** 39](#_Toc195453899)

[ **Difference between interface and abstract class?** 39](#_Toc195453900)

[ **Where will you use interface and abstract class with real time examples?** 39](#_Toc195453901)

[ **What is encapsulation? How are you currently using it?** 39](#_Toc195453902)

[ **Explain Compile time and Run time polymorphism?** 39](#_Toc195453903)

[ **What is method overriding?** 39](#_Toc195453904)

[ **How to achieve method overriding?** 39](#_Toc195453905)

[ **What is method overloading?** 39](#_Toc195453906)

[ **What is Polymorphism? Explain with an example?** 39](#_Toc195453907)

[ **What is Inheritance?** 39](#_Toc195453908)

[ **What is the interface in java?** 39](#_Toc195453909)

[ **Can we have a static method inside the interface?** 39](#_Toc195453910)

[ **Can we override the static method?** 39](#_Toc195453911)

[ **We have Class A and Class B which contains m1() method in both the class, and class B extends parent class A. Asked which method will get called on which object?** 40](#_Toc195453912)

[**Collection Framework** 41](#_Toc195453913)

[ **Difference between List and Set?** 41](#_Toc195453914)

[ **Difference between ArrayList and LinkedList?** 41](#_Toc195453915)

[ **Difference between Array and Collections?** 41](#_Toc195453916)

[ **Difference between HashMap and LinkedHashMap?** 41](#_Toc195453917)

[ **Difference between HashMap and HashTable?** 41](#_Toc195453918)

[ **Difference between Fail-Fast and Fail-Safe iterator?** 41](#_Toc195453919)

[ **Which is more efficient among the ArrayList and LinkedList?** 41](#_Toc195453920)

[ **Difference between Stream and Collections?** 41](#_Toc195453921)

[ **Write a syntax for integer ArrayList?** 41](#_Toc195453922)

[ **How can Set not store duplicate values? Explain internal working?** 41](#_Toc195453923)

[ **Which collection you will use in order to remove duplicate elements and preserve the insertion order?** 41](#_Toc195453924)

[ **Which Collection will you avoid duplicating and store data in a sorted manner?** 41](#_Toc195453925)

[ **In which scenarios Linked list comes into fixture in HashMap?** 41](#_Toc195453926)

[ **Which among ArrayList and LinkedList will be more efficient for random access of data?** 41](#_Toc195453927)

[ **Difference between Array, ArrayList and LinkedList? When you will use ArrayList and LinkedList?** 41](#_Toc195453928)

[ **How to sort the ArrayList?** 42](#_Toc195453929)

[ **How to sort the ArrayList?** 42](#_Toc195453930)

[ **What are the practical use cases of LinkedList?** 42](#_Toc195453931)

[ **Compare ArrayList objects from the employee and return the highest employee age?** 42](#_Toc195453932)

[ **How will you find the 3rd last element from the linked list in a single iteration?** 42](#_Toc195453933)

[ **How does ArrayList internally work?** 42](#_Toc195453934)

[ **What is the default size of ArrayList?** 42](#_Toc195453935)

[ **How to increase the size of ArrayList? By how much?** 42](#_Toc195453936)

[ **Difference between Map and Set?** 42](#_Toc195453937)

[ **How HashMap calls hashCode and equals method internally?** 42](#_Toc195453938)

[ **Explain internal working of ArrayList?** 42](#_Toc195453939)

[ **How does ArrayList increase its size dynamically?** 42](#_Toc195453940)

[ **How does ArrayList grow?** 42](#_Toc195453941)

[ **Explain internal working of ensureCapacity method?** 42](#_Toc195453942)

[ **How do you write a custom ArrayList which will take the input capacity?** 42](#_Toc195453943)

[ **Which data structure is used by ArrayList?** 43](#_Toc195453944)

[ **Difference between ArrayList and Vector?** 43](#_Toc195453945)

[ **Best time complexity of Arraylist and Linkedlist? Which is better?** 43](#_Toc195453946)

[ **What is HashMap?** 43](#_Toc195453947)

[ **What is the use of a ConcurrentHashMap?** 43](#_Toc195453948)

[ **Explain internal working of HashMap?** 43](#_Toc195453949)

[ **What is hash collision in HashMap?** 43](#_Toc195453950)

[ **What are the types of HashMap?** 43](#_Toc195453951)

[ **Can we store null value in HashMap?** 43](#_Toc195453952)

[ **Can we store infinite data in a HashMap?** 43](#_Toc195453953)

[ **Can we access null value in HashMap?** 43](#_Toc195453954)

[ **How hashcode calculations happened on the Employee object?** 43](#_Toc195453955)

[ **Can we have duplicate keys in HashMap?** 43](#_Toc195453956)

[ **How does HashMap come to know that two objects are the same or different?** 43](#_Toc195453957)

[ **How does ConcurrentHashMap works?** 43](#_Toc195453958)

[ **Difference between HashMap and ConcurrentHashMap?** 43](#_Toc195453959)

[ **Difference between HashMap and HashSet** 43](#_Toc195453960)

[ **What is the default size of ArrayList?** 44](#_Toc195453961)

[ **How to increase the size of ArrayList?** 44](#_Toc195453962)

[ **Can we add infinite elements to ArrayList?** 44](#_Toc195453963)

[ **Can we store infinite elements in a list?** 44](#_Toc195453964)

[ **How can we add data while iterating ArrayList?** 44](#_Toc195453965)

[ **What is the difference between List and Map?** 44](#_Toc195453966)

[ **Which data structure used by HashSet?** 44](#_Toc195453967)

[ **Explain sorted collections in java?** 44](#_Toc195453968)

[ **Which data structure is used by HashMap?** 44](#_Toc195453969)

[ **How can Set can’t store duplicate data?** 44](#_Toc195453970)

[ **Explain HashSet?** 44](#_Toc195453971)

[ **If we have added the same key in HashMap with the same value and different key with the same hash with the same value, then how will it be stored? At what position data will store and how it will store? What will happen when we use ConcurrentHashMap?** 44](#_Toc195453972)

[ **What is the default capacity of HashMap?** 44](#_Toc195453973)

[ **Where will you use ConcurrentHashMap?** 44](#_Toc195453974)

[ **Explain HashMap vs ConcurrentHashMap?** 44](#_Toc195453975)

[ **How is HashMap faster as compared to ConcurrentHashMap?** 45](#_Toc195453976)

[ **Suppose I have multiple Employee objects with different names, so how does HashMap differentiate between different objects?** 45](#_Toc195453977)

[ **What is a Blocking Queue?** 45](#_Toc195453978)

[**Exception Handling** 46](#_Toc195453979)

[ **What is Exception Handling?** 46](#_Toc195453980)

[ **Explain the hierarchy of Exception?** 46](#_Toc195453981)

[ **How to create Custom Exceptions?** 46](#_Toc195453982)

[ **How to handle IOException?** 46](#_Toc195453983)

[ **Difference between Checked Exception and Unchecked Exception?** 46](#_Toc195453984)

[ **How to handle unchecked exceptions?** 46](#_Toc195453985)

[ **What will happen if I use Exception and custom exceptions together? Priority?** 46](#_Toc195453986)

[ **Have you created a custom exception? What is the purpose of that?** 46](#_Toc195453987)

[ **What is the use of finally block in exception handling?** 46](#_Toc195453988)

[ **What is a try with resources?** 46](#_Toc195453989)

[ **Have you come across the OutOfMemory exception? How will you handle it? Explain with standalone code?** 46](#_Toc195453990)

[ **If you have try, catch and finally block and if I have written another exception inside catch block so how control will flow?** 46](#_Toc195453991)

[ **Have you ever got outOfMemory exceptions? Can you tell me any scenarios where you have faced this exception?** 46](#_Toc195453992)

[**Java 8 Features** 47](#_Toc195453993)

[ **Explain Java8 features?** 47](#_Toc195453994)

[ **What is Functional Interface?** 48](#_Toc195453995)

[ **What is the default method in functional interface?** 48](#_Toc195453996)

[ **What is Lambda Expression?** 48](#_Toc195453997)

[ **What is the prerequisite for Lambda Expression?** 48](#_Toc195453998)

[ **What is the map function in stream?** 48](#_Toc195453999)

[ **What is the filter in the stream?** 48](#_Toc195454000)

[ **What is Stream API?** 48](#_Toc195454001)

[ **What is the use of @FunctionalInterface annotation? Will it be checked manually or by the compiler?** 48](#_Toc195454002)

[ **Difference between Map and FlatMap in stream?** 49](#_Toc195454003)

[ **Difference between Map and Filter in java 8?** 49](#_Toc195454004)

[ **What is the intermediate and terminal operator in Java stream API?** 49](#_Toc195454005)

[ **Advantages of Stream over for loop?** 49](#_Toc195454006)

[ **What is the lambda expression in java 8?** 49](#_Toc195454007)

[ **How to use lambda expressions with one example?** 49](#_Toc195454008)

[ **What is Stream and Parallel stream?** 49](#_Toc195454009)

[ **Explain Optional class in java 8?** 49](#_Toc195454010)

[ **What is the purpose of Optional class?** 49](#_Toc195454011)

[ **Explain Java 8 Functional interface?** 49](#_Toc195454012)

[ **Where can you use FlatMap in a stream?** 49](#_Toc195454013)

[ **What is the lambda expression?** 49](#_Toc195454014)

[ **What is the use map method in java 8 stream?** 49](#_Toc195454015)

[ **How to sort the list of objects in java using Stream API?** 49](#_Toc195454016)

[ **What is the inbuilt functional interface in Java 8?** 49](#_Toc195454017)

[ **What is the Consumer interface in java?** 49](#_Toc195454018)

[ **What is Predicate and Consumer in java 8?** 49](#_Toc195454019)

[ **Difference between Predicate and Function?** 50](#_Toc195454020)

[ **What is the use of Map?** 50](#_Toc195454021)

[ **What is Optional class? What are the advantages of using Optional class?** 50](#_Toc195454022)

[ **What are all functional interfaces present in java?** 50](#_Toc195454023)

[ **Optional case returning null and String, how to handle it? What are the advantages of using it?** 50](#_Toc195454024)

[ **Give an example of Functional Interface in java?** 50](#_Toc195454025)

[ **Suppose you are using lambda expressions and other than lambda expressions then are we getting any performance benefits?** 50](#_Toc195454026)

[ **Can you elaborate on the Stream API, purpose and use of it?** 50](#_Toc195454027)

[ **Can we have more than one abstract method in a functional interface?** 50](#_Toc195454028)

[ **Have you worked on NullPointerException? How to handle NullPointerException in java 8?** 50](#_Toc195454029)

[**Java 11 Features** 51](#_Toc195454030)

[ **Explain Java 11 features?** 51](#_Toc195454031)

[**Java 17 Features** 53](#_Toc195454032)

[ **Explain Java 17 features?** 53](#_Toc195454033)

[**Java 21 Features** 55](#_Toc195454034)

[ **Explain Java 21 features?** 55](#_Toc195454035)

[ **What is the added version after Java 17 to Java 21?** 56](#_Toc195454036)

[**Multithreading** 57](#_Toc195454037)

[ **What is multithreading in java?** 57](#_Toc195454038)

[ **Difference between Process and Thread?** 57](#_Toc195454039)

[ **Explain the Thread Life Cycle?** 57](#_Toc195454040)

[ **What is a deadlock? Explain with an example?** 57](#_Toc195454041)

[ **What is a Synchronized Thread?** 57](#_Toc195454042)

[ **Difference between synchronized and volatile keyword?** 57](#_Toc195454043)

[ **What happens if a thread calls notify() instead of notifyAll()?** 57](#_Toc195454044)

[ **How to create threads using java 1.7 and java 1.8?** 57](#_Toc195454045)

[ **Difference between wait and sleep?** 57](#_Toc195454046)

[ **What is Thread.sleep() method, and what does it do internally?** 57](#_Toc195454047)

[ **Difference between Thread and Runnable in Java?** 57](#_Toc195454048)

[ **What is significance of Executors in Java?** 57](#_Toc195454049)

[ **Explain the Producer Consumer problem and how would you implement it using wait(), notify() or BlockingQueue?** 57](#_Toc195454050)

[ **Difference between Callable and Runnable. When should you use each?** 57](#_Toc195454051)

[ **What is ForkJoinPool in Java and how does it differ from normal ExecutorService?** 57](#_Toc195454052)

[ **Explain Deadlock and how do you avoid it in Java Multithreading?** 58](#_Toc195454053)

[ **What are the ways of blocking the threads in java?** 58](#_Toc195454054)

[ **What is thread synchronization in java?** 58](#_Toc195454055)

[ **What are the methods of thread?** 58](#_Toc195454056)

[ **What is thread pool? What are the types of thread pool?** 58](#_Toc195454057)

[ **What are thread pool methods?** 58](#_Toc195454058)

[ **How to create threads in java?** 58](#_Toc195454059)

[ **How to start a thread in java?** 58](#_Toc195454060)

[ **How to create threads in two different ways?** 58](#_Toc195454061)

[ **What is the necessity of thread synchronization?** 58](#_Toc195454062)

[ **How to create a Thread pool in java?** 58](#_Toc195454063)

[ **Which method you will override in order to create a thread?** 58](#_Toc195454064)

[ **What are all methods present in Thread class?** 58](#_Toc195454065)

[ **How do notify methods work in multithreading?** 58](#_Toc195454066)

[ **What is synchronized in multithreading?** 58](#_Toc195454067)

[ **Thread safety with Singleton class?** 58](#_Toc195454068)

[ **Explain Thread executors?** 58](#_Toc195454069)

[ **How to create three threads so that it will execute simultaneously?** 59](#_Toc195454070)

[ **What can we replace instead of this keyword in the statement? synchronized(this) {?** 59](#_Toc195454071)

[ **What is Reentrant lock in java?** 59](#_Toc195454072)

[ **What is Callable interface?** 59](#_Toc195454073)

[ **What is Runnable interface?** 59](#_Toc195454074)

[**Scenario based Questions** 61](#_Toc195454075)

[ **Scenario based question - 1?** 61](#_Toc195454076)

[ **Scenario based question - 2?** 61](#_Toc195454077)

[ **Scenario based question - 3?** 61](#_Toc195454078)

[ **Scenario based question - 4?** 61](#_Toc195454079)

[ **Scenario based question - 5?** 62](#_Toc195454080)

[***Java Programs*** 64](#_Toc195454081)

[ **Program – 1** 64](#_Toc195454082)

[ **Program – 2** 64](#_Toc195454083)

[ **Program – 3** 64](#_Toc195454084)

[ **Program – 4** 64](#_Toc195454085)

[ **Program – 5** 64](#_Toc195454086)

[ **Program – 6** 64](#_Toc195454087)

[ **Program – 7** 65](#_Toc195454088)

[ **Program – 8** 65](#_Toc195454089)

[ **Program – 9** 66](#_Toc195454090)

[ **Program – 10** 66](#_Toc195454091)

[ **Program – 11** 66](#_Toc195454092)

[ **Program – 12** 66](#_Toc195454093)

[ **Program – 13** 66](#_Toc195454094)

[ **Program – 14** 66](#_Toc195454095)

[ **Program – 15** 66](#_Toc195454096)

[ **Program – 16** 66](#_Toc195454097)

[ **Program – 17** 66](#_Toc195454098)

[ **Program – 18** 66](#_Toc195454099)

[ **Program – 19** 67](#_Toc195454100)

[ **Program – 20** 67](#_Toc195454101)

[ **Program – 21** 67](#_Toc195454102)

[ **Program – 22** 67](#_Toc195454103)

[ **Program – 23** 67](#_Toc195454104)

[ **Program – 24** 67](#_Toc195454105)

[ **Program – 25** 67](#_Toc195454106)

[ **Program – 26** 67](#_Toc195454107)

[ **Program – 27** 67](#_Toc195454108)

[ **Program – 28** 67](#_Toc195454109)

[ **Program – 29** 68](#_Toc195454110)

[ **Program – 30** 68](#_Toc195454111)

[ **Program – 31** 68](#_Toc195454112)

[ **Program – 32** 69](#_Toc195454113)

[ **Program – 33** 69](#_Toc195454114)

[ **Program – 34** 69](#_Toc195454115)

[ **Program – 35** 69](#_Toc195454116)

[ **Program – 36** 69](#_Toc195454117)

[ **Program – 37** 69](#_Toc195454118)

[ **Program – 38** 69](#_Toc195454119)

[ **Program – 39** 69](#_Toc195454120)

[ **Program – 40** 69](#_Toc195454121)

[ **Program – 41** 70](#_Toc195454122)

[ **Program – 42** 70](#_Toc195454123)

[ **Program – 43** 70](#_Toc195454124)

[ **Program – 44** 70](#_Toc195454125)

[ **Program – 45** 70](#_Toc195454126)

[ **Program – 46** 70](#_Toc195454127)

[ **Program – 47** 70](#_Toc195454128)

[ **Program – 48** 70](#_Toc195454129)

[ **Program – 49** 70](#_Toc195454130)

[ **Program – 50** 71](#_Toc195454131)

[ **Program – 51** 71](#_Toc195454132)

[ **Program – 52** 71](#_Toc195454133)

[ **Program – 53** 72](#_Toc195454134)

[ **Program – 54** 72](#_Toc195454135)

[ **Program – 55** 72](#_Toc195454136)

[ **Program – 56** 72](#_Toc195454137)

[ **Program – 57** 72](#_Toc195454138)

[ **Program – 58** 72](#_Toc195454139)

[ **Program – 59** 73](#_Toc195454140)

[ **Program – 60** 73](#_Toc195454141)

[ **Program – 61** 73](#_Toc195454142)

[ **Program – 62** 73](#_Toc195454143)

[ **Program – 63** 73](#_Toc195454144)

[ **Program – 64** 74](#_Toc195454145)

[ **Program – 65** 74](#_Toc195454146)

[ **Program – 66** 74](#_Toc195454147)

[ **Program – 67** 75](#_Toc195454148)

[ **Program – 68** 75](#_Toc195454149)

[ **Program – 69** 75](#_Toc195454150)

[ **Program – 70** 75](#_Toc195454151)

[ **Program – 71** 75](#_Toc195454152)

[ **Program – 72** 76](#_Toc195454153)

[ **Program – 73** 76](#_Toc195454154)

[ **Program – 74** 76](#_Toc195454155)

[ **Program – 75** 76](#_Toc195454156)

[ **Program – 76** 76](#_Toc195454157)

[Advance Java / J2EE 77](#_Toc195454158)

[ **What is servlet?** 77](#_Toc195454159)

[ **Explain servlet lifecycle?** 77](#_Toc195454160)

[ **What is a dispatcher servlet?** 77](#_Toc195454161)

[ **What is JSP?** 77](#_Toc195454162)

[ **What is the difference between servlet and JSP?** 77](#_Toc195454163)

[ **What is the JSP lifecycle?** 77](#_Toc195454164)

[ **What is Spring MVC? Explain architecture of spring MVC?** 77](#_Toc195454165)

[ **What is the singleton design pattern?** 77](#_Toc195454166)

[ **What is the REST API?** 77](#_Toc195454167)

[ **Difference between API and REST?** 77](#_Toc195454168)

[ **Difference between SOAP and REST?** 77](#_Toc195454169)

[ **Which among SOAP and REST is easy?** 77](#_Toc195454170)

[ **Explain REST architecture?** 77](#_Toc195454171)

[ **What do you know about DTD and XSD?** 77](#_Toc195454172)

[ **Explain SOAP Web Service?** 77](#_Toc195454173)

[ **What is Path in XML?** 78](#_Toc195454174)

[ **Write a path for a given xml file?** 78](#_Toc195454175)

[ **What is the use of XSLT?** 78](#_Toc195454176)

[ **How does SOAP work?** 78](#_Toc195454177)

[ **What are the action types of http?** 78](#_Toc195454178)

[ **Explain how you will create a SOAP web service?** 78](#_Toc195454179)

[Data Structure 79](#_Toc195454180)

[ **Find the middle element from the singly linked list without finding the length?** 79](#_Toc195454181)

[ **Complexity to find length of singly linked list?** 79](#_Toc195454182)

[ **Do you know how LinkedList works? Can you create a data structure for LinkedList?** 79](#_Toc195454183)

[ **How will you traverse LinkedList? Your List will contains multiple nodes so how will you traverse over it? Can you write a pseudo code? You have your head with you so how will you move to the next node? How will the current node be null?** 79](#_Toc195454184)

[ **Explain Merge Sort??** 79](#_Toc195454185)

[Design Patterns 80](#_Toc195454186)

[ **What is the singleton design pattern? Explain it?** 80](#_Toc195454187)

[ **Explain Factory design pattern?** 80](#_Toc195454188)

[ **What is the Observer design pattern?** 80](#_Toc195454189)

[ **Except for the creational design pattern, can you explain other design patterns?** 80](#_Toc195454190)

[ **Explain Saga design pattern?** 80](#_Toc195454191)

[Spring 81](#_Toc195454192)

[ **Explain Spring MVC architecture?** 81](#_Toc195454193)

[ **Explain Spring IOC?** 81](#_Toc195454194)

[ **Explain Spring Dependency Injection?** 81](#_Toc195454195)

[ **What are the spring bean scopes?** 81](#_Toc195454196)

[ **Explain any scope of spring bean?** 81](#_Toc195454197)

[ **How to add scope in bean configuration?** 81](#_Toc195454198)

[ **Difference between session and prototype scope?** 81](#_Toc195454199)

[ **What is the Spring batch? Explain where you have used the spring batch in your project?** 81](#_Toc195454200)

[ **How spring batch works?** 81](#_Toc195454201)

[ **How to implement the spring batch?** 81](#_Toc195454202)

[ **What is the difference between SOAP and REST? Which is more efficient?** 81](#_Toc195454203)

[ **How REST API works?** 81](#_Toc195454204)

[ **Difference between Spring and Spring Boot?** 81](#_Toc195454205)

[ **Difference between application context and beanfactory in spring?** 81](#_Toc195454206)

[ **Explain Spring MVC architecture?** 81](#_Toc195454207)

[ **Difference between Spring and Spring Boot?** 82](#_Toc195454208)

[ **What is the use of @Transactional annotation in spring?** 82](#_Toc195454209)

[ **What are different stereotypes annotations in spring?** 82](#_Toc195454210)

[ **What is the front controller?** 82](#_Toc195454211)

[ **Difference between applicationcontext and beanfactory?** 82](#_Toc195454212)

[ **Explain spring bean life cycle?** 82](#_Toc195454213)

[ **If I have prototype bean and singleton bean, then how can we call prototype beans inside singleton bean?** 82](#_Toc195454214)

[ **What are containers present in the spring?** 82](#_Toc195454215)

[ **How transactions work in spring?** 82](#_Toc195454216)

[ **Do you know the web.xml file? What is the significance of a web.xml file?** 82](#_Toc195454217)

[ **What is the dependency injection?** 82](#_Toc195454218)

[ **What are the different ways of injection?** 82](#_Toc195454219)

[ **Who will take care of objects in Spring?** 82](#_Toc195454220)

[ **What is ApplicationContext and its types in Spring?** 82](#_Toc195454221)

[ **Difference between Filter and Interceptor in Spring MVC?** 82](#_Toc195454222)

[ **Which one has access to spring beans, interceptors or filters?** 82](#_Toc195454223)

[ **Why do interceptors have access to Spring beans?** 83](#_Toc195454224)

[ **Difference between Spring data JPA, Hibernate, JDBC, My batis?** 83](#_Toc195454225)

[ **Why do we use Spring data JPA?** 83](#_Toc195454226)

[Hibernate 84](#_Toc195454227)

[ How data is persisted in database by using hibernate**?** 84](#_Toc195454228)

[ **How to retrieve select query using Hibernate?** 86](#_Toc195454229)

[ **How to retrieve data through hibernate?** 86](#_Toc195454230)

[ **Hibernate caching?** 86](#_Toc195454231)

[ **Explain L1 and L2 cache in Hibernate?** 86](#_Toc195454232)

[ **Explain the hibernate sessions?** 87](#_Toc195454233)

[ **Difference between JDBC and Hibernate?** 87](#_Toc195454234)

[ **What is the difference between get and load method in hibernate?** 87](#_Toc195454235)

[ **Explain bean scopes?** 87](#_Toc195454236)

[ **Can we have beans without bean id?** 87](#_Toc195454237)

[ **Difference between get and load method?** 87](#_Toc195454238)

[ **Explain Performance tuning and indexing in hibernate?** 87](#_Toc195454239)

[ **Explain different classes and annotations in hibernate?** 87](#_Toc195454240)

[ **How to call stored procedures using hibernate?** 87](#_Toc195454241)

[ **What is Many to many associations in hibernate?** 87](#_Toc195454242)

[ **What is JPA?** 87](#_Toc195454243)

[ **What is the many to many associations in hibernate?** 87](#_Toc195454244)

[ **What is a Session factory in hibernate?** 87](#_Toc195454245)

[ **Explain Many to many associations in hibernate?** 87](#_Toc195454246)

[ **If I want to retrieve a few entries from an employee table, how to retrieve employee details through JPA?** 87](#_Toc195454247)

[ **Explain a built-in function in JPA?** 87](#_Toc195454248)

[ **What is lazy loading in hibernate?** 88](#_Toc195454249)

[ **What is circular dependency?** 88](#_Toc195454250)

[ **What is the second level cache in hibernate?** 88](#_Toc195454251)

[ **Can we use Hibernate without JPA?** 88](#_Toc195454252)

[ **What are the advantages of ORM?** 88](#_Toc195454253)

[Spring Boot 89](#_Toc195454254)

[ **Difference between Spring and Spring Boot?** 89](#_Toc195454255)

[ **What are the advantages of SprigBoot?** 89](#_Toc195454256)

[ **What is a yaml file in SpringBoot?** 89](#_Toc195454257)

[ **What is what is @transient in spring boot?** 89](#_Toc195454258)

[ **What are the types of propagation?** 89](#_Toc195454259)

[ **What @Qualifier in spring? What is the use of @Qualifier?** 89](#_Toc195454260)

[ **What are the advantages of Spring boot?** 89](#_Toc195454261)

[ **How to create multiple modules using spring boot?** 89](#_Toc195454262)

[ **What are all methods present in the controller?** 89](#_Toc195454263)

[ **Explain annotations in Spring boot?** 89](#_Toc195454264)

[ **What is the query parameter?** 89](#_Toc195454265)

[ **How to change the server in Spring boot?** 89](#_Toc195454266)

[ **Difference between @Controller and @RestController?** 89](#_Toc195454267)

[ **What is the use of @Responsebody annotation?** 89](#_Toc195454268)

[ **Explain @RestController annotation in Spring boot?** 89](#_Toc195454269)

[ **Which two annotations are used in built by @RestController?** 89](#_Toc195454270)

[ **How to access application.properties properties in java code?** 90](#_Toc195454271)

[ **Explain Spring boot application? How does it work?** 90](#_Toc195454272)

[ **What is the use of @Service annotation in spring?** 90](#_Toc195454273)

[ **What is the use of @Component annotation in spring?** 90](#_Toc195454274)

[ **What is the use of @Repository annotation in spring?** 90](#_Toc195454275)

[ **What is the difference between @Service and @Controller annotation?** 90](#_Toc195454276)

[ **What is @ComponentScan?** 90](#_Toc195454277)

[ **What is the use of @Autowired annotation?** 90](#_Toc195454278)

[ **Write a REST API to accept user name as input, return the response with message as "Hello, user" and status as "success" and Endpoint - /api/message?** 90](#_Toc195454279)

[ **How to handle Exceptions in Spring boot?** 90](#_Toc195454280)

[ **How to provide security to spring boot applications?** 90](#_Toc195454281)

[ **What is @ControllerAdvice in Spring boot?** 90](#_Toc195454282)

[ **How to retrieve query parameters in Spring boot?** 90](#_Toc195454283)

[ **How is the data persisted in DB with the REST API?** 90](#_Toc195454284)

[ **Explain GET and POST methods?** 90](#_Toc195454285)

[ **Write a SQL or JPQL to retrieve the data from DB?** 90](#_Toc195454286)

[ **How will you add Employee object data in the DB using POST call?** 91](#_Toc195454287)

[ **What is the size of Spring in your project? Who will decide that?** 91](#_Toc195454288)

[ **Explain about spring security?** 91](#_Toc195454289)

[ **Explain the flow of REST api?** 91](#_Toc195454290)

[ **What will happen when we replace @Service instead of @Repository?** 91](#_Toc195454291)

[ **Explain uses of SpringBoot application?** 91](#_Toc195454292)

[ **How is https used in SpringBoot applications?** 91](#_Toc195454293)

[ **Difference between RequestMapping and GetMapping?** 91](#_Toc195454294)

[ **If you have to create a Spring boot application? What steps will you follow?** 91](#_Toc195454295)

[ **What is a starter kit in spring boot application?** 91](#_Toc195454296)

[ **What are the starters in Spring boot application?** 91](#_Toc195454297)

[ **What are Actuators in spring boot applications?** 91](#_Toc195454298)

[ **What are custom exceptions? How do you define your own custom exception?** 91](#_Toc195454299)

[ **Have you worked on Unit testing? Which framework have you used in your project?** 91](#_Toc195454300)

[ **What is the difference between lazy loading and eager loading?** 91](#_Toc195454301)

[ **Difference between put and patch?** 91](#_Toc195454302)

[ **Can we use @RequestMapping at class level and method level both?** 91](#_Toc195454303)

[ **What will happen if I use @RequestMapping at method level?** 92](#_Toc195454304)

[ **How to map Service class to RestController?** 92](#_Toc195454305)

[ **What is Spring security? Have you used Spring security in your project?** 92](#_Toc195454306)

[ **What are Profiles in Spring boot?** 92](#_Toc195454307)

[ **How will you do CRUD operations using spring boot applications?** 92](#_Toc195454308)

[ **How to change from JAR to WAR in Spring boot?** 92](#_Toc195454309)

[ **Where will you give database details in the spring boot application?** 92](#_Toc195454310)

[ **How will you give database details for the individual environment?** 92](#_Toc195454311)

[ **What is a JPA repository?** 92](#_Toc195454312)

[ **What is the use of @SpringBootApplication annotation in Spring boot?** 92](#_Toc195454313)

[ **How will the Spring boot application know whether its JAR or WAR type?** 92](#_Toc195454314)

[ **How are you providing security to your services?** 92](#_Toc195454315)

[ **As an end user how can I consume your services?** 92](#_Toc195454316)

[ **How to validate a token?** 92](#_Toc195454317)

[ **Explain a few security algorithms?** 92](#_Toc195454318)

[ **What is native queries in Spring boot?** 92](#_Toc195454319)

[ **Where do you use native queries?** 92](#_Toc195454320)

[ **Suppose I am having different microservices, if I want to invoke different microservice so how can we interact?** 93](#_Toc195454321)

[ **What kind of information are you passing to the GET method?** 93](#_Toc195454322)

[ **Suppose I want to start a spring boot application, so what are all the steps?** 93](#_Toc195454323)

[ **How to configure a custom server in a Spring boot application?** 93](#_Toc195454324)

[ **How to configure multiple databases in a Spring boot application?** 93](#_Toc195454325)

[ **How does your application know I am using a particular database?** 93](#_Toc195454326)

[ **What is the use of @Qalifier annotation?** 93](#_Toc195454327)

[ **How are REST calls intercepted in our project? How is the call happening?** 93](#_Toc195454328)

[ **When I hit a URL how data will be invoked with your application?** 93](#_Toc195454329)

[ **How is the DTO getting invoked with REST endpoints?** 93](#_Toc195454330)

[ **I have a URL and I hit the URL, so which is the first class which is invoked in our product?** 93](#_Toc195454331)

[ **If you have to expose a new endpoint, what steps will you follow?** 93](#_Toc195454332)

[ **What is the difference between @Bean and @Configuration?** 93](#_Toc195454333)

[ **What is the difference between RequestParam and path variable? Give me one example with one URL?** 93](#_Toc195454334)

[ **How to change the port of the embedded tomcat server in Spring boot? Write a syntax for that?** 93](#_Toc195454335)

[ **What are Profiles in Spring boot applications? Give one example?** 94](#_Toc195454336)

[ **Which annotations are we using in order to use the Exception?** 94](#_Toc195454337)

[ **What is the default scope for spring beans?** 94](#_Toc195454338)

[ **Which database have you used? How do you integrate databases with JPA?** 94](#_Toc195454339)

[ **How to communicate between two spring boot applications using asynchronous calls?** 94](#_Toc195454340)

[ **Have you worked messaging services?** 94](#_Toc195454341)

[ **How will you handle errors / exceptions in your project?** 94](#_Toc195454342)

[ **What are Actuators?** 94](#_Toc195454343)

[ **How are you keeping track of logs in your project?** 94](#_Toc195454344)

[ **If you have millions of data then how will you manage it?** 94](#_Toc195454345)

[ **What is the difference between REST API and Http?** 94](#_Toc195454346)

[ **Explain @Autowired usage?** 94](#_Toc195454347)

[ **How does the REST API work?** 94](#_Toc195454348)

[ **How can we handle a yaml file when it is exposed?** 94](#_Toc195454349)

[ **How to externalize variables in spring boot?** 94](#_Toc195454350)

[ **What is the Spring reactive framework?** 94](#_Toc195454351)

[ **Why should we use Spring reactive?** 94](#_Toc195454352)

[ **What is the use of block and subscribe?** 95](#_Toc195454353)

[ **What is a novel based server?** 95](#_Toc195454354)

[ **How will the findById or findByName method in the repository work with name? When we go with camelcase then where is it implemented internally?** 95](#_Toc195454355)

[ **How does the JPA Repository know there is a class?** 95](#_Toc195454356)

[ **How does the Spring Boot application work internally?** 95](#_Toc195454357)

[ **How will you authenticate API in Spring Boot?** 95](#_Toc195454358)

[ **In the Spring REST API what is the White label error accessing from the browser? What is the status code you will get?** 95](#_Toc195454359)

[Microservices 96](#_Toc195454360)

[ **Monolith vs Microservices differences and when to choose which?** 96](#_Toc195454361)

[ **How to design a microservice from scratch?** 96](#_Toc195454362)

[ **API Gateway pattern and its advantages?** 96](#_Toc195454363)

[ **Inter-service communication: REST vs Messaging?** 96](#_Toc195454364)

[ **Circuit Breaker pattern and its implementation using Resilience4j?** 96](#_Toc195454365)

[ **Load balancing in microservices using Spring Cloud Load Balancer?** 96](#_Toc195454366)

[ **How Spring Cloud Config helps in centralized configuration management?** 96](#_Toc195454367)

[ **Service discovery using Eureka or Consul?** 96](#_Toc195454368)

[ **Feign Client vs WebClient: Which one to use and why?** 96](#_Toc195454369)

[ **Event-driven architecture and Kafka integration?** 96](#_Toc195454370)

[ **Database per service vs Shared Database: Pros and cons?** 96](#_Toc195454371)

[ **Saga Pattern for distributed transactions in microservices?** 96](#_Toc195454372)

[ **JWT-based authentication and OAuth2 in microservices?** 96](#_Toc195454373)

[ **How to handle security in an API Gateway?** 96](#_Toc195454374)

[ **Observability: Logging, tracing, and monitoring best practices?** 96](#_Toc195454375)

[ **Role of Prometheus and Grafana in microservices monitoring?** 97](#_Toc195454376)

[ **Kubernetes deployment strategies for microservices?** 97](#_Toc195454377)

[ **Blue-Green and Canary deployments in microservices?** 97](#_Toc195454378)

[ **When to use WebFlux for reactive microservices?** 97](#_Toc195454379)

[ **CQRS and Event Sourcing: When and why to use them?** 97](#_Toc195454380)

[ **Difference between Monolithic and Microservice architecture?** 97](#_Toc195454381)

[ **Explain Microservices architecture?** 97](#_Toc195454382)

[ **Where are you deploying your microservice?** 97](#_Toc195454383)

[ **What are different Aggregation patterns in Microservices?** 97](#_Toc195454384)

[ **What are challenges in Microservice architecture?** 97](#_Toc195454385)

[ **How are you handling load in your microservices?** 97](#_Toc195454386)

[ **How will you handle resilience and fault tolerance in microservice?** 97](#_Toc195454387)

[ **Can you explain Event driven microservice and how you have used it in your project?** 97](#_Toc195454388)

[SQL 98](#_Toc195454389)

[ **What is ordered by and range in the database?** 98](#_Toc195454390)

[ **Write a SQL query to get the second highest salary of an employee from the employee table?** 98](#_Toc195454391)

[ **Write a SQL query to print employee id, employee\_name, depatment\_name Employee - employee\_id, employee\_name, department\_id** 98](#_Toc195454392)

[**Department - department\_id, department\_name?** 98](#_Toc195454393)

[ **What is a collision in the oracle database?** 98](#_Toc195454394)

[ **If you create multiple folders one inside another and store one file into the last folder then how can you implement using database tables in oracle? How many tables are required?** 98](#_Toc195454395)

[ **What are DDL and DML commands?** 98](#_Toc195454396)

[ **Write a query to update data in the table?** 98](#_Toc195454397)

[ **What is the use of joins in the database?** 98](#_Toc195454398)

[ **Explain ACID properties in the database?** 98](#_Toc195454399)

[ **Explain Normalization?** 98](#_Toc195454400)

[ **What are the types of Normalization?** 98](#_Toc195454401)

[ **What is Denormalization?** 98](#_Toc195454402)

[ **What is the use of join?** 98](#_Toc195454403)

[ **Write a SQL query to return the 7th highest salary of an employee?** 99](#_Toc195454404)

[ **What is Union in SQL?** 99](#_Toc195454405)

[ **Can we perform union operations on one table?** 99](#_Toc195454406)

[ **Difference between where and let?** 99](#_Toc195454407)

[ **What is Semaphore?** 99](#_Toc195454408)

[ **What is View and Index in SQL?** 99](#_Toc195454409)

[ **What is Trigger in SQL? What is the use of Trigger in SQL?** 99](#_Toc195454410)

[ **What is inner join in SQL?** 99](#_Toc195454411)

[ **Write a SQL query to get emp\_sal > 5000 and company\_name = "Kotak"** 99](#_Toc195454412)

[ **Write a JPA query to retrieve the above result?** 99](#_Toc195454413)

[ **What is an index in SQL?** 99](#_Toc195454414)

[ **Explain different types of Index?** 99](#_Toc195454415)

[ **Explain different types of joins?** 100](#_Toc195454416)

[ **Difference between Outer join and Full outer join?** 100](#_Toc195454417)

[ **Write a SQL query to find the name of a city which has more than one customer in it?** 100](#_Toc195454418)

[ **Write SQL query to display 10th highest salary of employee?** 100](#_Toc195454419)

[ **What is the difference between DDL and DML commands?** 100](#_Toc195454420)

[ **Difference between Inner join and Outer join?** 100](#_Toc195454421)

[ **Difference between left outer join and right outer join?** 100](#_Toc195454422)

[ **What are Triggers?** 100](#_Toc195454423)

[ **SQL query to get the youngest employee from the employee table?** 100](#_Toc195454424)

[ **SQL query to get list of employees from Finance department?** 100](#_Toc195454425)

[ **What is Indexing in SQL?** 100](#_Toc195454426)

[ **What are the disadvantages of Index?** 100](#_Toc195454427)

[ **What is the difference between primary key and unique key?** 100](#_Toc195454428)

[ **What is the difference between implicit cursor and explicit cursor?** 100](#_Toc195454429)

[ **What is a correlated subquery?** 100](#_Toc195454430)

[ **What is the left outer join?** 101](#_Toc195454431)

[ **What is the use of stored procedure?** 101](#_Toc195454432)

[ **Explain the database schema for Students and Course table?** 101](#_Toc195454433)

[ **What is UNION ALL?** 101](#_Toc195454434)

[ **SQL query to get second highest salary?** 101](#_Toc195454435)

[ **Write a SQL query to display a list of students which have a Mathematics subject?** 101](#_Toc195454436)

[ **What is Normalization?** 101](#_Toc195454437)

[ **Can you explain different forms of Normalization with one example?** 101](#_Toc195454438)

[ **What kind of fetching strategy are you using in hibernate?** 101](#_Toc195454439)

[ **What is OneToMany mapping?** 101](#_Toc195454440)

[ **Write a SQL query to get the 3rd highest salary of an employee?** 101](#_Toc195454441)

[ **What is dense and rank in SQL?** 101](#_Toc195454442)

[ **What is the difference between SQL and NoSQL?** 101](#_Toc195454443)

[ **Difference between Stored Procedure and Function?** 101](#_Toc195454444)

[ **What is Transaction in SQL?** 102](#_Toc195454445)

[ **Write a SQL query:** 102](#_Toc195454446)

[ **SQL query that categorizes employees based on their salary into different percentage ranges.** 102](#_Toc195454447)

[ **Write a SQL query to find the department name who does not have an employee? (Employee and Department table given)?** 102](#_Toc195454448)

[ **What is the difference between Primary Key and Unique Key?** 102](#_Toc195454449)

[ **What is the Composite key?** 103](#_Toc195454450)

[ **What is Self join?** 103](#_Toc195454451)

[ **We have Table A (2 columns and 4 rows) and Table B (2 columns and 4 rows) and I want to print all 8 rows? Write a query for it? (Use Union All)?** 103](#_Toc195454452)

[ **What is the difference between SQL Union and Union All?** 103](#_Toc195454453)

[ **Write a SQL query and print the employee name which does not have any department?** 103](#_Toc195454454)

[ **Write a query - Employee table and add one extra column in output as category? Let's say if salary is in 1 to 4000 then Category 1, 4000 - 7000 then category 2, 7000 - 10000 then category 3?** 103](#_Toc195454455)

[Kafka 104](#_Toc195454456)

[ **What is Apache Kafka?** 104](#_Toc195454457)

[ **Why does Kafka compare to RabbitMQ?** 104](#_Toc195454458)

[ **How to scale up Kafka?** 104](#_Toc195454459)

[ **What are the default partitions in kafka?** 104](#_Toc195454460)

[ **Difference between topic and partition in Kafka?** 104](#_Toc195454461)

[ **Is there any ordering of messages in partitions in kafka?** 104](#_Toc195454462)

[ **How does partition assignment happen? What method is used?** 104](#_Toc195454463)

[ **What is offset and partition in Kafka?** 104](#_Toc195454464)

[GitHub 105](#_Toc195454465)

[ **What is Version control i.e. Git?** 105](#_Toc195454466)

[ **How to push and commit code in git using commands?** 105](#_Toc195454467)

[ **Difference between rebase and merge?** 105](#_Toc195454468)

[ **Which versioning tool are you using? Explain a few Git commands?** 105](#_Toc195454469)

[ **What is the Git command to create a new branch from an existing branch?** 105](#_Toc195454470)

[ **If we have more consumer groups than partitions and how will it manage?** 105](#_Toc195454471)

[Docker 106](#_Toc195454472)

[ **Explain use cases of Docker?** 106](#_Toc195454473)

[ **You have 2 containers 1 and 2, can they communicate with each other?** 106](#_Toc195454474)

[ **If both are with different hosts then whether they will be able to communicate with each other or not?** 106](#_Toc195454475)

[ **What is the host in this case?** 106](#_Toc195454476)

[ **Imagine you have one volume and you want to share it with multiple containers? How will you do it?** 106](#_Toc195454477)

[ **Where does an object get stored when created inside a docker container?** 106](#_Toc195454478)

[ **What is the process of Docker?** 106](#_Toc195454479)

[ **How is Docker helpful?** 106](#_Toc195454480)

[ **What is the use of Docker?** 106](#_Toc195454481)

[Kubernetes 107](#_Toc195454482)

[ **What are Kubernetes Secrets??** 107](#_Toc195454483)

[AWS 108](#_Toc195454484)

[ **What is cloud computing?** 108](#_Toc195454485)

[ **Explain services provided by cloud computing?** 108](#_Toc195454486)

[ **How to deploy service to AWS?** 108](#_Toc195454487)

[ **What are services in AWS?** 108](#_Toc195454488)

[ **Which services have you used?** 108](#_Toc195454489)

[ **What are different services in EKS?** 108](#_Toc195454490)

[ **What is the benefit of Serverless services? Is EKS serverless?** 108](#_Toc195454491)

[Azure 109](#_Toc195454492)

[ **question?** 109](#_Toc195454493)

[Angular 110](#_Toc195454494)

[ **What are the components in the Angular?** 110](#_Toc195454495)

[ **How to pass data from components in Angular?** 110](#_Toc195454496)

[ **What does the component.ts file contain in angular?** 110](#_Toc195454497)

[ **Explain two way data binding in angular?** 110](#_Toc195454498)

[ **How to make backend calls in angular?** 110](#_Toc195454499)

[ **What is authguard in angular?** 110](#_Toc195454500)

[ **What is interpolation in angular?** 110](#_Toc195454501)

[ **How to create components in angular?** 110](#_Toc195454502)

[ **Explain angular directives?** 110](#_Toc195454503)

[ **What are observables in angular?** 110](#_Toc195454504)

[ **How to implement dependency injection in angular?** 110](#_Toc195454505)

[ **What are different components of Angular?** 110](#_Toc195454506)

[ **How to handle caching in Angular?** 110](#_Toc195454507)

[React 111](#_Toc195454508)

[ **Features of ReactJS?** 111](#_Toc195454509)

[ **How does Virtual DOM work?** 111](#_Toc195454510)

[ **Features of Virtual DOM?** 111](#_Toc195454511)

[ **What is reducing modifications in Virtual DOM?** 111](#_Toc195454512)

[ **What do you call the backend API from ReactJS?** 111](#_Toc195454513)

[ **What are other Hooks in React?** 111](#_Toc195454514)

[ **What is the purpose of Hook?** 111](#_Toc195454515)

[ **What is the dependency array in useEffect?** 111](#_Toc195454516)

[ **What is the use and significance of dependency arrays?** 111](#_Toc195454517)

[ **How to make a call to an external API from ReactJS?** 111](#_Toc195454518)

[ **What is an alternative to Axios?** 111](#_Toc195454519)

[Others 112](#_Toc195454520)

[ **Explain Jira?** 112](#_Toc195454521)

[ **What is the output of below program?** 112](#_Toc195454522)

[ **What is https protocol?** 112](#_Toc195454523)

[ **How does https internally work?** 112](#_Toc195454524)

[ **What is a certificate?** 112](#_Toc195454525)

[ **Difference between http and https?** 112](#_Toc195454526)

[ **What is a Sprint retrospective?** 112](#_Toc195454527)

[ **Do you know anything about CI/CD?** 112](#_Toc195454528)

[ **How will Jenkins' job work? Explain the process?** 112](#_Toc195454529)

[ **What is Node JS?** 112](#_Toc195454530)

[ **How is your application getting deployed? Which platform are you using in your project?** 112](#_Toc195454531)

[ **What do you know about Asynchronous communication?** 112](#_Toc195454532)

[ **What are the measure objects i.e key things in order to make Asynchronous calls?** 112](#_Toc195454533)

[ **If I have two different technologies then how can we achieve asynchronous communication?** 113](#_Toc195454534)

[ **Do you have exposure in AI and ML?** 113](#_Toc195454535)

[ **Examples of Fund Transfer are Synchronous or Asynchronous?** 113](#_Toc195454536)

[ **Is WhatsApp Synchronous or Asynchronous?** 113](#_Toc195454537)

[ **Difference between 2 Tier and 3 Tier Architecture?** 113](#_Toc195454538)

[ **Have you heard about the log4j vulnerability?** 113](#_Toc195454539)

[ **What is https protocol?** 113](#_Toc195454540)

[ **Difference between http and https?** 113](#_Toc195454541)

[ **Suppose in server one job at 9 AM and copy json file and paste it in temp folder another job 6 PM read it and load in db and no one should change the file. How to prevent file from any Change?** 113](#_Toc195454542)

[ **If we increase timeout then how can we resolve it? Deployment related issue and how will you handle it?** 113](#_Toc195454543)

[Linux 114](#_Toc195454544)

[ **Explain Linux commands?** 114](#_Toc195454545)

[ **Command - Read a file and print first two lines from file Ans. head -2 filename?** 114](#_Toc195454546)

[ **Search a word from a file?** 114](#_Toc195454547)

[ **Why do you want to change your current organization?** 114](#_Toc195454548)

[Puzzles 115](#_Toc195454549)

[ **Puzzle – 1** 115](#_Toc195454550)

[ **Puzzle – 2** 115](#_Toc195454551)

[ **Puzzle – 3** 115](#_Toc195454552)

# ***Java***

## **Java Fundamentals**

### **Explain how to install java? Explain directories in it?**

**Step 1: Download Java**

Go to the official Oracle website or use OpenJDK: Oracle JDK / OpenJDK

**Step 2: Choose Version**

Choose the appropriate JDK version (e.g., Java 17 or Java 21 – latest LTS versions).

Select the installer for your operating system (Windows, macOS, Linux).

**Step 3: Install Windows:**

Run the .exe installer and follow the wizard.

It installs Java to a default path like: C:\Program Files\Java\jdk-21

macOS/Linux:

Use .tar.gz or package managers like Homebrew, apt, yum.

sudo apt install openjdk-21-jdk # For Ubuntu

**Step 4: Set Environment Variables (Windows)**

Go to System Properties > Environment Variables.

Add to Path: C:\Program Files\Java\jdk-21\bin

Set JAVA\_HOME: JAVA\_HOME = C:\Program Files\Java\jdk-21

**Step 5: Verify Installation**

Open a terminal/command prompt and run:

java -version

javac -version



### **How to configure java in windows?**

* Answer

### **Explain Java Program execution?**

* **1. Writing the Source Code**
  + You create a .java file containing your Java code (e.g., HelloWorld.java)
  + Example:

public class HelloWorld {

public static void main(String[] args) {

System.out.println("Hello, World!");

}

}

**2. Compilation (javac)**

* + The Java compiler (javac) converts your source code into **bytecode** (platform-independent intermediate code)
  + Creates .class files (e.g., HelloWorld.class)
  + Bytecode is not machine code - it's instructions for the **Java Virtual Machine (JVM)**

**3. Class Loading**

When you run java HelloWorld:

**a) Bootstrap ClassLoader**

* Loads core Java classes (from rt.jar and other core libraries)
* Written in native code (not Java)

**b) Extension ClassLoader**

* Loads classes from Java extension directories

**c) Application ClassLoader**

* Loads your application classes (from the classpath)

**4. Bytecode Verification**

* JVM verifies the bytecode to ensure:
  + No illegal memory access
  + Proper stack manipulation
  + Correct method calls
  + No violation of access restrictions
* This provides security by preventing malicious code

**5. Just-In-Time (JIT) Compilation**

* The JVM's JIT compiler converts frequently executed bytecode into **native machine code**
* Happens at runtime
* Optimizes performance by:
  + Inlining methods
  + Removing dead code
  + Optimizing loops

**6. Execution**

* The JVM executes the program:
  + Creates the main thread
  + Allocates memory for objects in the Heap
  + Manages method calls using the Stack
  + Handles garbage collection automatically

**7. Runtime Memory Areas**

The JVM manages these memory areas during execution:

a) Method Area

* Stores class structures, method code, and static variables

b) Heap

* Stores all objects and their instance variables
* Garbage collection works here

c) JVM Stacks

* Each thread has its own stack
* Stores frames for each method call (local variables, operands, return values)

d) PC Registers

* Tracks the execution position for each thread

e) Native Method Stacks

* For native code (non-Java code)

**8. Garbage Collection**

* Automatically reclaims memory from objects no longer in use
* Runs in the background
* Different algorithms available (Serial, Parallel, G1, ZGC, etc.)



### **Explain the byte stream in java?**

* answer

### **What is Classloader? Different types of it?**

* Answer

### **Why is java more secure?**

* Answer

### **Explain public static void main (String[] args) method?**

* Answer

### **What is the marker interface?**

* Answer

### **What is serialization and deserialization?**

* Answer

### **What is the use of the transient keyword in serialization?**

* Answer

### **What is the final class?**

* Answer

### **Why is string immutable in java?**

* Answer

### **What is StringBuffer?**

* Answer

### **What is StringBuilder?**

* Answer

### **Difference between String, StringBuffer and StringBuilder?**

* Answer

### **What is the final keyword in java?**

* Answer

### **Where can we use the final keyword?**

* Answer

### **What will happen if I create the final class?**

* Answer

### **Difference between comparable and comparator?**

* Answer

### **What is static in java?**

* **Static** is a modifier that can be applied to variables, methods, blocks, and nested classes.

**Static Variable (Class Variable):**

* Belongs to the class rather than any object
* Shared by all instances of the class
* Initialized when class is loaded

class Counter {

static int count = 0; // static variable

Counter() {

count++;

}

}

**Static Method:**

* Belongs to the class rather than instances
* Can be called without creating an object
* Can only access static members directly

class MathUtils {

static int add(int a, int b) { // static method

return a + b;

}

}

// Usage: MathUtils.add(5, 3);

**Static Block:**

* Used for static initialization of a class
* Executed when the class is loaded

class MyClass {

static {

System.out.println("Static block executed");

}

}

**Example Combining Both:**

abstract class Database {

static final String DEFAULT\_URL = "jdbc:default"; // static constant

abstract void connect(); // abstract method

static void printDefaultUrl() { // static method

System.out.println(DEFAULT\_URL);

}

}



**Note:**

Static members are resolved at compile-time (early binding), while abstract methods enable runtime polymorphism (late binding).

### **Explain Generics? What is the use of Generics?**

* answer

### **How to find the second largest element from the list?**

* answer

### **What is Singleton in java?**

* answer

### **What will happen if I assign null to some object then will that be garbage collected?**

* answer

### **When does the finalize method get called?**

* answer

### **What is JDBC?**

* answer

### **Consider we have Strings as String s1 = “Welcome”, String s2 = “Welcome” and String s3 = “WelcomeOne”. What does s1==s2 represent?**

* answer

### **Explain String constant pool?**

* answer

### **What is the difference between equals() and hashCode() method?**

* answer

### **Does Java support Pointers?**

* answer

### **Why doesn't java support pointers?**

* answer

### **Explain which language you prefer in terms of security among C, C++, Java and Python?**

* answer

### **Is Java 100% object oriented?**

* answer

### **How can we make java 100% object oriented?**

* answer

### **Can we create objects of static class?**

* Answer

### **What is the difference between ArrayList and LinkedList? Which is better?**

* answer

### **What is Enumeration in java?**

* answer

### **Which compiler is used by Java?**

* answer

### **Where are the hashCode() and equals() methods defined in java?**

* answer

### **How many design patterns are present in java?**

* answer

### **Can we override the static method?**

* answer

### **Explain access modifiers in java?**

* answer

### **What is volatile in java? Where can we use it?**

* answer

### **What is an idempotent method in java?**

* answer

### **What is final, static and non-static (instance) in java?**

* answer

### **How will I use two threads using a singleton design pattern?**

* answer

### **How to implement thread safety in java using singleton?**

* answer

### **Write an immutable class in Java?**

* answer

### **Can we override the protected method?**

* answer

### **What is marker interface in java? What is the purpose of the marker interface?**

* answer

### **How can we ignore the finally block?**

* answer

### **What is synchronous and asynchronous call?**

* answer

### **Can we write the main method as private?**

* answer

### **What is the use of static keywords in java?**

* answer

### **What is inner class? Have you used inner class in your project?**

* answer

### **What is the purpose of inner class?**

* answer

### **What is Enum? Write a syntax for enum?**

* answer

### **Difference between static and non-static in java?**

* answer

### **How can we break the singleton pattern?**

* answer

### **Difference between default and protected access modifier?**

* answer

### **Why is string immutable?**

* answer

### **What is an immutable class?**

* answer

### **What is atomic integer in java?**

* answer

### **How can you make a singleton class as thread safe?**

* answer

### **Difference between compareTo and compare method?**

* answer

### **What is the return type of compare method?**

* answer

### **What is contract in equals and hashcode method?**

* answer

### **Difference between Marker interface and Functional interface?**

* answer

### **Difference between Abstract class and Interface? Explain with scenarios which one you will it be used with conditions?**

* answer

### **What is the use of static and instance variables in java?**

* answer

### **How will you access JNDI in java code?**

* answer

### **Why is the main method in java static?**

* answer

### **What is the use of the static keyword?**

* answer

### **How many ways can we create beans in java?**

* answer

### **Explain different types of calling beans?**

* answer

### **What are the default methods?**

* answer

### **Can I add one more new default method in my child class? Does it give any problems with my existing classes?**

* answer

### **Difference between Comparable and Comparator interface?**

* answer

### **In Java, which class is the base class of all the classes?**

* answer

### **Why is the equals method required?**

* answer

### **Is it possible to have the same hashcode for multiple objects?**

* answer

### **Enlist a few predefined interfaces from Java 8?**

* answer

### **If I have a 200 MB excel file so can java read that file?**

* answer

### **What is immutability and mutability in java?**

* answer

### **Custom implementation of Singleton beans?**

* answer

### **What is the default scope of beans?**

* answer

### **Difference between static and final?**

* answer

### **Explain about String class in java?**

* answer

### **What is a special feature of String?**

* answer

### **Create an immutable class in java?**

* answer

### **How to access Singleton class from another class?**

* answer

### **Where will the Singleton class be stored in memory?**

* answer

### **Difference between bucket level and segment level locking?**

* answer

### **How is the static method bound with an object?**

* answer

### **What is the use of the intern() method in String class?**

* answer

## **Java Memory Management**

### **Explain JDK, JRE?**

* answer

### **Explain JVM?**

* Answer

### **What are the different loaders in JVM?**

* answer

### **Explain Java memory management?**

* answer

### **What is Stack / Heap memory?**

* Answer

### **Explain memory management below? Int and Integer?**

* answer

### **Where local variables are stored in memory?**

* answer

### **Where objects will be stored in memory?**

* answer

### **Explain Heap memory in java?**

* Answer

### **Difference between Stack and Heap?**

* answer

### **What is Garbage Collection in java?**

* answer

## **OOPs**

### **Explain OOPs concept?**

* Object-Oriented Programming (OOPs) is a programming paradigm based on the concept of "objects", which can contain data and code. Java is a fully object-oriented language (except for primitive types) and supports the following core OOPs concepts:
* **Abstraction**
  + Hides internal implementation and shows only functionality.
  + Achieved using **abstract classes** and **interfaces**.
* **Encapsulation**
  + Bundling of data (variables) and methods into a single unit (class).
  + Achieved using **access modifiers** (private, public, protected).
* **Inheritance**
  + Allows a class (child) to inherit fields and methods from another class (parent).
  + Promotes code reusability and method overriding.
* **Polymorphism**
  + One entity behaves differently based on context.
  + **Compile-time (method overloading)** and **Run-time (method overriding)** are the two types.

### **What is an object?**

* answer

### **Explain method in Object class? Explain 9 methods in Object class?**

* Answer

### **How many ways can we create objects of class in java?**

* answer

### **What is Encapsulation? Explain with a real time example**

* Encapsulation is one of the fundamental OOPs principles in Java. It refers to wrapping data (variables) and the code (methods) that operates on the data into a single unit — typically a class. It helps protect the internal state of an object from unwanted external modifications.
* Encapsulation is achieved using:
  + Private fields to restrict direct access.
  + Public getters and setters to allow controlled access.
* Example: Bank Account

public class BankAccount {

private double balance; // private field — can't be accessed directly

public BankAccount(double initialBalance) {

this.balance = initialBalance;

}

// public method to get balance

public double getBalance() {

return balance;

}

// public method to deposit money (with validation)

public void deposit(double amount) {

if (amount > 0) {

balance += amount;

}

}

// public method to withdraw money (with validation)

public void withdraw(double amount) {

if (amount > 0 && balance >= amount) {

balance -= amount;

}

}

}

**Explanation:**

* The balance variable is **encapsulated** — it's private.
* Access is provided **only through public methods** (getBalance, deposit, withdraw), ensuring that:
  + The balance can't be set to a negative value directly.
  + Deposits and withdrawals go through validation.

This prevents incorrect or unauthorized changes to the data, which is the essence of **encapsulation** in real-world applications.

### **How do we achieve encapsulation in Java?**

* Encapsulation is one of the four fundamental OOP concepts in Java. It refers to bundling data (variables) and methods that operate on that data into a single unit (class) while restricting direct access to some of the object's components.

**Key Ways to Achieve Encapsulation:**

**1. Using Private Access Modifier**

Make class fields private to prevent direct access from outside the class.

public class Person {

private String name; // private field

private int age;

}

**2. Providing Public Getter and Setter Methods**

Create public methods to access and modify private fields.

public class Person {

private String name;

private int age;

// Getter for name

public String getName() {

return name;

}

// Setter for name

public void setName(String name) {

this.name = name;

}

// Getter for age

public int getAge() {

return age;

}

// Setter for age with validation

public void setAge(int age) {

if(age > 0) { // validation logic

this.age = age;

}

}

}

**3. Implementing Constructors with Validation**

Initialize fields through constructors with validation.

public Person(String name, int age) {

this.name = name;

if(age > 0) {

this.age = age;

}

}

**Benefits of Encapsulation:**

* **Data Hiding**: Internal representation is hidden from outside
* **Increased Flexibility**: Can change internal implementation without affecting other code
* **Reusability**: Encapsulated code is easier to reuse
* **Control**: Can add validation logic in setters
* **Maintainability**: Easier to maintain and modify code

**Example Usage:**

public class Main {

public static void main(String[] args) {

Person person = new Person("Alice", 30);

// Access through getters

System.out.println(person.getName()); // Alice

System.out.println(person.getAge()); // 30

// Modify through setters

person.setAge(31);

person.setName("Alice Smith");

// Invalid age won't be set

person.setAge(-5); // age remains 31

}

}

### **What is abstract in java?**

* **Abstract** is a non-access modifier that can be applied to classes and methods.

**Abstract Class:** Abstract is a non-access modifier that can be applied to classes and methods.

* Cannot be instantiated (cannot create objects)
* Can contain both abstract and concrete methods
* Used as a base class for inheritance

abstract class Animal {

// Abstract method (no implementation)

abstract void makeSound();

// Concrete method

void eat() {

System.out.println("Animal is eating");

}

}

**Abstract Method:**

* Has no body (no implementation)
* Must be overridden by the first concrete subclass
* Can only exist in abstract classes

abstract class Shape {

abstract double calculateArea(); // abstract method

}

**Key Points:**

* If a class has even one abstract method, the class must be declared abstract
* Abstract classes can have constructors (called when subclass is instantiated)
* Used to define common interface for subclasses

### **Explain Composition in Oops?**

* Answer

### **What is composition and aggregation?**

* answer

### **Parent and child class methods and how to access those?**

* answer

### **Difference between interface and abstract class?**

* answer

### **Where will you use interface and abstract class with real time examples?**

* answer

### **What is encapsulation? How are you currently using it?**

* answer

### **Explain Compile time and Run time polymorphism?**

* answer

### **What is method overriding?**

* Answer

### **How to achieve method overriding?**

* Answer

### **What is method overloading?**

* answer

### **What is Polymorphism? Explain with an example?**

* answer

### **What is Inheritance?**

* Answer

### **What is the interface in java?**

* Answer

### **Can we have a static method inside the interface?**

* Answer

### **Can we override the static method?**

* answer

### **We have Class A and Class B which contains m1() method in both the class, and class B extends parent class A. Asked which method will get called on which object?**

* answer

## **Collection Framework**

### **Difference between List and Set?**

* answer

### **Difference between ArrayList and LinkedList?**

* Answer

### **Difference between Array and Collections?**

* Answer

### **Difference between HashMap and LinkedHashMap?**

* Answer

### **Difference between HashMap and HashTable?**

* answer

### **Difference between Fail-Fast and Fail-Safe iterator?**

* answer

### **Which is more efficient among the ArrayList and LinkedList?**

* Answer

### **Difference between Stream and Collections?**

* Answer

### **Write a syntax for integer ArrayList?**

* answer

### **How can Set not store duplicate values? Explain internal working?**

* Answer

### **Which collection you will use in order to remove duplicate elements and preserve the insertion order?**

* answer

### **Which Collection will you avoid duplicating and store data in a sorted manner?**

* answer

### **In which scenarios Linked list comes into fixture in HashMap?**

* answer

### **Which among ArrayList and LinkedList will be more efficient for random access of data?**

* Answer

### **Difference between Array, ArrayList and LinkedList? When you will use ArrayList and LinkedList?**

* answer

### **How to sort the ArrayList?**

* Answer

### **How to sort the ArrayList?**

* Answer

### **What are the practical use cases of LinkedList?**

* answer

### **Compare ArrayList objects from the employee and return the highest employee age?**

* Answer

### **How will you find the 3rd last element from the linked list in a single iteration?**

* answer

### **How does ArrayList internally work?**

* answer

### **What is the default size of ArrayList?**

* answer

### **How to increase the size of ArrayList? By how much?**

* Answer

### **Difference between Map and Set?**

* answer

### **How HashMap calls hashCode and equals method internally?**

* answer

### **Explain internal working of ArrayList?**

* answer

### **How does ArrayList increase its size dynamically?**

* answer

### **How does ArrayList grow?**

* Answer

### **Explain internal working of ensureCapacity method?**

private void ensureCapacityInternal(int minCapacity)

{

if (elementData == DEFAULTCAPACITY\_EMPTY\_ELEMENTDATA) {

minCapacity = Math.max(DEFAULT\_CAPACITY, minCapacity);

}

ensureExplicitCapacity(minCapacity);

}

### **How do you write a custom ArrayList which will take the input capacity?**

answer

### **Which data structure is used by ArrayList?**

* answer

### **Difference between ArrayList and Vector?**

* Answer

### **Best time complexity of Arraylist and Linkedlist? Which is better?**

* Answer

### **What is HashMap?**

* Answer

### **What is the use of a ConcurrentHashMap?**

* answer

### **Explain internal working of HashMap?**

* Answer

### **What is hash collision in HashMap?**

* Answer

### **What are the types of HashMap?**

* answer

### **Can we store null value in HashMap?**

* answer

### **Can we store infinite data in a HashMap?**

* answer

### **Can we access null value in HashMap?**

* Answer

### **How hashcode calculations happened on the Employee object?**

* answer

### **Can we have duplicate keys in HashMap?**

* answer

### **How does HashMap come to know that two objects are the same or different?**

* answer

### **How does ConcurrentHashMap works?**

* answer

### **Difference between HashMap and ConcurrentHashMap?**

* Answer

### **Difference between HashMap and HashSet**

* Answer

### **What is the default size of ArrayList?**

* answer

### **How to increase the size of ArrayList?**

* answer

### **Can we add infinite elements to ArrayList?**

* answer

### **Can we store infinite elements in a list?**

* answer

### **How can we add data while iterating ArrayList?**

* answer

### **What is the difference between List and Map?**

* answer

### **Which data structure used by HashSet?**

* answer

### **Explain sorted collections in java?**

* answer

### **Which data structure is used by HashMap?**

* answer

### **How can Set can’t store duplicate data?**

* answer

### **Explain HashSet?**

* answer

### **If we have added the same key in HashMap with the same value and different key with the same hash with the same value, then how will it be stored? At what position data will store and how it will store? What will happen when we use ConcurrentHashMap?**

* answer

### **What is the default capacity of HashMap?**

* Answer

### **Where will you use ConcurrentHashMap?**

* answer

### **Explain HashMap vs ConcurrentHashMap?**

* answer

### **How is HashMap faster as compared to ConcurrentHashMap?**

* answer

### **Suppose I have multiple Employee objects with different names, so how does HashMap differentiate between different objects?**

* answer

### **What is a Blocking Queue?**

* answer

## **Exception Handling**

### **What is Exception Handling?**

* Answer

### **Explain the hierarchy of Exception?**

* answer

### **How to create Custom Exceptions?**

* Answer

### **How to handle IOException?**

* answer

### **Difference between Checked Exception and Unchecked Exception?**

* answer

### **How to handle unchecked exceptions?**

* answer

### **What will happen if I use Exception and custom exceptions together? Priority?**

* answer

### **Have you created a custom exception? What is the purpose of that?**

* answer

### **What is the use of finally block in exception handling?**

* answer

### **What is a try with resources?**

* answer

### **Have you come across the OutOfMemory exception? How will you handle it? Explain with standalone code?**

* Answer

### **If you have try, catch and finally block and if I have written another exception inside catch block so how control will flow?**

* Answer

### **Have you ever got outOfMemory exceptions? Can you tell me any scenarios where you have faced this exception?**

* answer

## **Java 8 Features**

### **Explain Java8 features?**

✅ **Java 8 Overview**

Java 8, released in **March 2014**, was a major release that introduced **functional programming**, **streams**, and **default methods**, making Java much more expressive and modern.

🔹 **1. Lambda Expressions (JEP 126)**

Introduces anonymous functions to make code more concise.

List<String> names = List.of("Alice", "Bob");

names.forEach(name -> System.out.println(name));

✅ Reduces boilerplate for iterators, callbacks, and functional logic.

🔹 **2. Functional Interfaces**

An interface with a single abstract method, used as the target for lambda expressions.

@FunctionalInterface

interface MyFunc {

void show();

}

✅ Examples: Runnable, Callable, Predicate, Function, Consumer

🔹 **3. Stream API (JEP 107)**

Provides a high-level abstraction for processing sequences of elements (like collections) in a declarative way.

List<String> result = names.stream()

.filter(n -> n.startsWith("A"))

.collect(Collectors.toList());

✅ Supports map, filter, reduce, collect, etc.

🔹 **4. Default and Static Methods in Interfaces**

Allows interfaces to have method implementations.

interface MyInterface {

default void show() {

System.out.println("Default Method");

}

static void print() {

System.out.println("Static Method");

}

}

✅ Enables interface evolution without breaking existing implementations.

🔹 **5. Method References**

Shorthand for calling existing methods.

names.forEach(System.out::println);

✅ Cleaner than writing lambdas when reusing existing methods.

🔹 **6. Optional Class**

A container to handle null values more gracefully.

Optional<String> name = Optional.ofNullable(getName());

name.ifPresent(System.out::println);

✅ Avoids NullPointerException and encourages functional style handling of absence.

🔹 **7. Date and Time API (java.time)**

A comprehensive replacement for legacy java.util.Date and Calendar.

LocalDate today = LocalDate.now();

LocalDate birthday = LocalDate.of(1990, Month.MARCH, 10);

✅ Immutable, thread-safe, and more readable.

🔹 **8. Nashorn JavaScript Engine**

Allows Java to execute JavaScript code.

ScriptEngine engine = new ScriptEngineManager().getEngineByName("nashorn");

engine.eval("print('Hello JavaScript')");

✅ Deprecated in later versions.

✅ **Summary Table:**

| **Feature** | **Description** |
| --- | --- |
| **Lambda Expressions** | Anonymous functions for concise code |
| **Functional Interfaces** | Target types for lambdas |
| **Stream API** | Declarative collection processing |
| **Default/Static Methods** | Method implementation in interfaces |
| **Method References** | Shorthand for lambda expressions |
| **Optional Class** | Avoids null pointer exceptions |
| **Date and Time API** | Modern date/time classes in java.time package |
| **Nashorn Engine** | JavaScript execution in Java (deprecated in later) |

### **What is Functional Interface?**

* Answer

### **What is the default method in functional interface?**

* Answer

### **What is Lambda Expression?**

* answer

### **What is the prerequisite for Lambda Expression?**

* answer

### **What is the map function in stream?**

* answer

### **What is the filter in the stream?**

* answer

### **What is Stream API?**

* answer

### **What is the use of @FunctionalInterface annotation? Will it be checked manually or by the compiler?**

* answer

### **Difference between Map and FlatMap in stream?**

* Answer

### **Difference between Map and Filter in java 8?**

* answer

### **What is the intermediate and terminal operator in Java stream API?**

* answer

### **Advantages of Stream over for loop?**

* answer

### **What is the lambda expression in java 8?**

* answer

### **How to use lambda expressions with one example?**

* answer

### **What is Stream and Parallel stream?**

* answer

### **Explain Optional class in java 8?**

* answer

### **What is the purpose of Optional class?**

* answer

### **Explain Java 8 Functional interface?**

* answer

### **Where can you use FlatMap in a stream?**

* answer

### **What is the lambda expression?**

* answer

### **What is the use map method in java 8 stream?**

* answer

### **How to sort the list of objects in java using Stream API?**

* answer

### **What is the inbuilt functional interface in Java 8?**

* Answer

### **What is the Consumer interface in java?**

* answer

### **What is Predicate and Consumer in java 8?**

* answer

### **Difference between Predicate and Function?**

* answer

### **What is the use of Map?**

* answer

### **What is Optional class? What are the advantages of using Optional class?**

* answer

### **What are all functional interfaces present in java?**

* answer

### **Optional case returning null and String, how to handle it? What are the advantages of using it?**

* answer

### **Give an example of Functional Interface in java?**

* Answer

### **Suppose you are using lambda expressions and other than lambda expressions then are we getting any performance benefits?**

* answer

### **Can you elaborate on the Stream API, purpose and use of it?**

* answer

### **Can we have more than one abstract method in a functional interface?**

* answer

### **Have you worked on NullPointerException? How to handle NullPointerException in java 8?**

* answer

## **Java 11 Features**

### **Explain Java 11 features?**

✅ Java 11 Features Overview

Java 11 (released in September 2018) is a Long-Term Support (LTS) version after Java 8 and introduced several important features and changes.

🔹 **1.** **Local-Variable Syntax for Lambda Parameters**

You can now use var in lambda expressions:

(var a, var b) -> a + b;

✅ Benefit: Useful when you want to add annotations to parameters.

🔹 **2.** New String Methods

Java 11 added several new utility methods to the String class:

| **Method** | **Description** |
| --- | --- |
| isBlank() | Checks if the string is empty or whitespace only |
| lines() | Converts a multiline string into a stream of lines |
| strip() | Removes leading and trailing white space (Unicode-aware) |
| stripLeading() | Removes leading whitespaces |
| stripTrailing() | Removes trailing whitespaces |
| repeat(int) | Repeats the string multiple times |

String str = " Java ";

System.out.println(str.isBlank()); // false

System.out.println("A\nB\nC".lines().count()); // 3

System.out.println(str.strip()); // "Java"

System.out.println("Hello ".repeat(3)); // Hello Hello Hello

🔹 **3.** **HTTP Client (Standardized)**

The new HTTP Client API introduced in Java 9 is now standard.

HttpClient client = HttpClient.newHttpClient();

HttpRequest request = HttpRequest.newBuilder()

.uri(URI.create("https://example.com"))

.build();

HttpResponse<String> response = client.send(request, HttpResponse.BodyHandlers.ofString());

System.out.println(response.body());

🔹 **4. Collection.toArray(IntFunction)**

New method to convert a collection to an array using a generator:

List<String> list = List.of("A", "B", "C");

String[] array = list.toArray(String[]::new);

🔹 **5. Notable Removals**

Java EE and CORBA Modules were removed.

E.g., Removed: javax.xml.bind, javax.activation, java.se.ee, etc.

🔹 **6. Launch Single-File Source Code**

You can run a .java file directly without compiling it manually:

java HelloWorld.java

✅ Useful for quick scripts and testing.

🔹 **7. Flight Recorder and Mission Control**

JDK Flight Recorder and Java Mission Control are now part of the JDK for better performance monitoring and diagnostics.

✅ **Summary Table:**

| **Feature** | **Description** |
| --- | --- |
| var in Lambda | Enables use of var in lambda parameters |
| New String Methods | Utility methods like isBlank(), lines() |
| Standard HTTP Client | Simplified modern HTTP communication |
| Collection.toArray() Overload | Converts collections to arrays efficiently |
| Removed Deprecated Modules | Java EE/CORBA APIs removed |
| Run .java file directly | Java source code launcher |

## **Java 17 Features**

### **Explain Java 17 features?**

✅ Java 17, released in September 2021, is an LTS (Long-Term Support) version like Java 11 and Java 8. It includes many enhancements, new features, and deprecations/removals aimed at improving performance, security, and developer productivity.

🔹 **1. Sealed Classes (Standard Feature)**

Sealed classes restrict which classes can extend or implement them.

public sealed class Vehicle permits Car, Bike {}

final class Car extends Vehicle {}

final class Bike extends Vehicle {}

✅ Use Case: Controlled inheritance for security and design clarity.

🔹 **2. Pattern Matching for instanceof (Standard Feature)**

Simplifies type checks and casting.

if (obj instanceof String s) {

System.out.println(s.toLowerCase());

}

✅ Cleaner and safer casting logic.

🔹 **3. Switch Expressions (Preview earlier, now stable)**

Allows switch to return values and have more concise syntax.

String result = switch (day) {

case MONDAY -> "Start of week";

case FRIDAY -> "Weekend is near";

default -> "Midweek";

};

✅ More readable and expressive control flow.

🔹 **4. Text Blocks (Introduced in Java 15, refined)**

Multi-line string literals with better formatting.

String json = """

{

"name": "ChatGPT",

"type": "AI"

}

""";

✅ Useful for JSON, SQL, XML, etc.

🔹 **5. New Pseudo-Random Number Generators (JEP 356)**

Provides new interfaces and implementations for random number generation.

RandomGenerator rand = RandomGeneratorFactory.of("L64X256MixRandom").create();

int num = rand.nextInt();

✅ More flexible and efficient random generation.

🔹 **6. Foreign Function & Memory API (Incubator)**

Allows Java programs to call native code and allocate memory outside the heap.

✅ Improves native interop without JNI.

🔹 **7. Enhanced NullPointerException Messages**

Helpful exception messages with more context.

Exception in thread "main" java.lang.NullPointerException:

Cannot invoke "String.length()" because "str" is null

✅ Great for debugging.

🔹 **8. Removal/Deprecation of Legacy Features**

Applet API deprecated.

Experimental AOT and JIT compilers removed.

RMI Activation removed.

🔹 **9. JEP 409: Sealed Interfaces**

Interfaces can also be sealed like classes.

public sealed interface Animal permits Dog, Cat {}

✅ **Summary Table:**

| **Feature** | **Description** |
| --- | --- |
| Sealed Classes & Interfaces | Restricts subclassing |
| Pattern Matching for instanceof | Cleaner type checks |
| Switch Expressions | Return values directly from switch |
| Text Blocks | Multiline string literals |
| Enhanced NPE Messages | Better error diagnostics |
| Foreign Function & Memory API | Safe native interop (Incubator) |
| New Random Generators | Flexible PRNG implementations |

## **Java 21 Features**

### **Explain Java 21 features?**

* Answer

✅ **Java 21 Overview**

Java 21, released in September 2023, is a Long-Term Support (LTS) version. It includes powerful new features, especially around pattern matching, virtual threads, and improvements in performance, syntax, and foreign APIs.

🔹 **1. Virtual Threads (JEP 444 - Final)**

Lightweight threads that help with high-throughput concurrent applications.

Thread.startVirtualThread(() -> {

// Non-blocking, lightweight task

System.out.println("Hello from a virtual thread");

});

✅ Great for scalable applications like servers and reactive systems.

🔹 **2. Record Patterns (JEP 440 - Final)**

Deconstruct records more concisely in pattern matching.

record Point(int x, int y) {}

void print(Object obj) {

if (obj instanceof Point(int x, int y)) {

System.out.println("X: " + x + ", Y: " + y);

}

}

✅ Improves destructuring of objects in conditionals.

🔹 **3. Pattern Matching for switch (JEP 441 - Final)**

Enhances switch expressions with type patterns.

static String format(Object obj) {

return switch (obj) {

case Integer i -> "int: " + i;

case String s -> "String: " + s.toUpperCase();

default -> "Unknown";

};

}

✅ Improves readability and reduces boilerplate.

🔹 **4. Sequenced Collections (JEP 431)**

New interface that adds consistent ordering to Collections (first(), last(), etc.).

SequencedCollection<String> names = new SequencedSet<>();

✅ Adds order-sensitive operations.

🔹 **5. String Templates (JEP 430 - Preview)**

Simplifies string construction with embedded expressions.

String name = "Java";

String message = STR."Welcome to \{name} 21!";

✅ Makes dynamic strings concise and readable.

🔹 **6. Scoped Values (JEP 446 - Preview)**

A safer alternative to thread-local variables, designed to work with virtual threads.

✅ Used for managing thread-local-like state safely in a structured way.

🔹 **7. Foreign Function & Memory API (JEP 442 - 3rd Preview)**

Modern and safe access to native code, replacing JNI.

✅ Supports calling native libraries and allocating off-heap memory.

🔹 **8. Deprecations & Removals**

Finalization is deprecated for removal.

Legacy APIs are cleaned and marked deprecated for future removal.

✅ **Summary Table:**

| **Feature** | **Description** |
| --- | --- |
| Virtual Threads | Lightweight threads for scalable concurrency |
| Record Patterns | Deconstruction of record types |
| Pattern Matching for switch | Type-safe, expressive switch |
| Sequenced Collections | Ordered collections with first/last semantics |
| String Templates (Preview) | Interpolated strings like ${} syntax |
| Scoped Values (Preview) | Safe alternative to ThreadLocal |
| Foreign Function & Memory API | Call native functions, work with memory safely |

### **What is the added version after Java 17 to Java 21?**

* answer

## **Multithreading**

### **What is multithreading in java?**

* answer

### **Difference between Process and Thread?**

* answer

### **Explain the Thread Life Cycle?**

* Answer

### **What is a deadlock? Explain with an example?**

* answer

### **What is a Synchronized Thread?**

* answer

### **Difference between synchronized and volatile keyword?**

* Answer

### **What happens if a thread calls notify() instead of notifyAll()?**

* Answer

### **How to create threads using java 1.7 and java 1.8?**

* answer

### **Difference between wait and sleep?**

* answer

### **What is Thread.sleep() method, and what does it do internally?**

* Answer

### **Difference between Thread and Runnable in Java?**

* Answer

### **What is significance of Executors in Java?**

* Answer

### **Explain the Producer Consumer problem and how would you implement it using wait(), notify() or BlockingQueue?**

* Answer

### **Difference between Callable and Runnable. When should you use each?**

* Answer

### **What is ForkJoinPool in Java and how does it differ from normal ExecutorService?**

* Answer

### **Explain Deadlock and how do you avoid it in Java Multithreading?**

* Answer

### **What are the ways of blocking the threads in java?**

* answer

### **What is thread synchronization in java?**

* answer

### **What are the methods of thread?**

* Answer

### **What is thread pool? What are the types of thread pool?**

* answer

### **What are thread pool methods?**

* Answer

### **How to create threads in java?**

* answer

### **How to start a thread in java?**

* answer

### **How to create threads in two different ways?**

* answer

### **What is the necessity of thread synchronization?**

* answer

### **How to create a Thread pool in java?**

* answer

### **Which method you will override in order to create a thread?**

* answer

### **What are all methods present in Thread class?**

* answer

### **How do notify methods work in multithreading?**

* answer

### **What is synchronized in multithreading?**

* answer

### **Thread safety with Singleton class?**

* answer

### **Explain Thread executors?**

* answer

### **How to create three threads so that it will execute simultaneously?**

* Answer

### **What can we replace instead of this keyword in the statement? synchronized(this) {?**

* answer

### **What is Reentrant lock in java?**

* answer

### **What is Callable interface?**

* **Callable** is a functional interface used to execute concurrent tasks that return a result and can throw exceptions.
* It's most commonly used with ExecutorService and Future.
* The **Callable** interface is part of the **java.util.concurrent** package and is similar to the Runnable interface but with some key differences.

**✅ Key Features of Callable:**

* **Returns a result** — Unlike Runnable, which returns void, Callable returns a result of a specified type.
* **Can throw checked exceptions** — Callable.call() method can throw checked exceptions.
* **Used with ExecutorService** — It is typically used with ExecutorService to submit tasks asynchronously.

**Example:**

import java.util.concurrent.\*;

public class CallableExample {

public static void main(String[] args) throws Exception {

ExecutorService executor = Executors.newSingleThreadExecutor();

Callable<String> task = () -> {

Thread.sleep(1000);

return "Task completed!";

};

Future<String> future = executor.submit(task);

System.out.println("Result: " + future.get()); // Blocks until result is ready

executor.shutdown();

}

}

### **What is Runnable interface?**

* The **Runnable** interface is a **functional interface** in Java, designed to represent a task that can be executed by a thread. It is part of the **java.lang** package.
* Runnable is used to execute code concurrently in threads.
* It does **not return a value** and **cannot throw checked exceptions**.
* Often used when the task is simple and doesn't need a return result.

**✅ Key Features of Runnable:**

1. **Single Abstract Method**:
   * void run();
   * Contains no parameters and does not return any result.
2. **Used to Define a Thread Task**:
   * Runnable is often used to define the code that should execute in a separate thread.
3. **Does Not Return a Value**:
   * Unlike Callable, it doesn’t return any result or throw checked exceptions.
4. **Can Be Passed to a Thread**:
   * You can pass an instance of Runnable to a Thread constructor.

**Example:**

public class MyRunnable implements Runnable {

@Override

public void run() {

System.out.println("Running in a thread: " + Thread.currentThread().getName());

}

public static void main(String[] args) {

Thread thread = new Thread(new MyRunnable());

thread.start();

}

}

## **Scenario based Questions**

### **Scenario based question - 1?**

String s1 = “abc”; and String s2 = new String(“abc”); Where will s1 and s2 be stored?

* answer

### **Scenario based question - 2?**

What is the output of the given Java code?

public class Test

{

public static void main(String[] args)

{

method(null);

}

public static void method(Object o)

{

System.out.println("Object method");

}

public static void method(Integer i)

{

System.out.println("Integer method");

}

}

* Answer

### **Scenario based question - 3?**

What is the output?

Map<Integer, String> aMap = new HashMap<>();

Integer a = new Integer(20);

Integer b = 20;

aMap.put(a, "Blume");

aMap.put(b, "BlumeGlobal");

System.out.println(aMap.get(20));

System.out.println(aMap.get(new Integer(20)));

System.out.println(aMap.get(b));

System.out.println(aMap.get(a));

* answer

### **Scenario based question - 4?**

Modify below code in order to print the value of a and b?

class Account {

int a;

int b;

public void setData(int a, int b) {

a = a;

b = b;

}

public void showData() {

System.out.println("Value of A ="+a);

System.out.println("Value of B ="+b);

}

public static void main(String args[]) {

Account obj = new Account();

obj.setData(2,3);

obj.showData();

}

}

* answer

### **Scenario based question - 5?**

What will happen if we declare the list below?

List<Object> var = new ArrayList<String>;

* answer

# ***Java Programs***

### **Program – 1**

Write a program in java to return max sum of contiguous subarray of size 3?

(Write algorithm steps as well)

Example. [2,1,5,1,3,2], where k=3 (max size of sub array)?

* answer

### **Program – 2**

Print even and odd numbers using thread

* Answer

### **Program – 3**

Write a method to return the maximum value from an integer array passed as an input parameter

public class FindLargestNumber

{

public static int returnLargetNumber(int arr[])

{

int i;

int largestNumber = arr[0];

for (i = 1; i < arr.length; i++)

if (arr[i] > largestNumber) {

largestNumber = arr[i];

return largestNumber;

}

public static void main(String[] args)

{

int arr[] = {55, 12, 0, 786, 98};

System.out.println("Largest number in given array : " + returnLargetNumber(arr));

}

}

### **Program – 4**

Problem Definition: Write a program to implement Singleton class?

* Answer

### **Program – 5**

Problem Definition: Write a program - method to return missing number from array of length n=3

Example: n=3, [2,0,3] -> Output -> 1

* Answer

### **Program – 6**

Problem Definition: Write a program in java to check if two strings are Anagram or not. Return boolean method

LISTEN - SILENT

Implement using collections.

import java.util.Arrays;

import java.util.HashMap;

public class Anagram {

public static boolean checkAnagramUsingArray(String str1, String str2) {

str1 = str1.replaceAll("\\s", "");

str2 = str2.replaceAll("\\s", "");

if (str1.length() != str2.length()) {

return false;

}

char[] str1Array = str1.toLowerCase().toCharArray();

char[] str2Array = str2.toLowerCase().toCharArray();

Arrays.sort(str1Array);

Arrays.sort(str2Array);

return Arrays.equals(str1Array, str2Array);

}

public static boolean checkAnagramUsingCollection(String str1, String str2) {

str1 = str1.replaceAll("\\s", "");

str2 = str2.replaceAll("\\s", "");

if (str1.length() != str2.length()) {

return false;

}

HashMap<Character, Integer> hm1 = new HashMap<Character, Integer>();

HashMap<Character, Integer> hm2 = new HashMap<Character, Integer>();

char[] str1Array = str1.toCharArray();

char[] str2Array = str2.toCharArray();

for (char value : str1Array) {

if (hm1.get(value) == null) {

hm1.put(value, 1);

} else {

int c = hm1.get(value);

hm1.put(value, ++c);

}

}

for (char c : str2Array) {

if (hm2.get(c) == null) {

hm2.put(c, 1);

} else {

int d = hm2.get(c);

hm2.put(c, ++d);

}

}

return hm1.equals(hm2);

}

public static void main(String[] args) {

String str1 = "LISTEN";

String str2 = "SILENT";

System.out.println(checkAnagramUsingArray(str1, str2));

System.out.println(checkAnagramUsingCollection(str1, str2));

}

}

### **Program – 7**

Problem Definition: Write a program to Swap two numbers without using a third variable

* Answer

### **Program – 8**

Problem Definition: Write a program to print even and odd numbers without using the modulus operator?

* Answer

### **Program – 9**

Problem Definition: Write a program to return the third largest element from an array using java 8 features?

* Answer

### **Program – 10**

Problem Definition: Write a program to print even and odd numbers using java 8 stream api?

* Answer

### **Program – 11**

Problem Definition: Write a program to print fibonacci series?

* Answer

### **Program – 12**

Problem Definition: Write a program to perform Reverse an array in groups of given size?

**Example:** [1,2,3,4,5,6,7] where k=3

**Output:** [5,6,7,1,2,3,4]

* Answer

### **Program – 13**

Problem Definition: Write a program to print factorials of natural numbers from 1 to 10

* Answer

### **Program – 14**

Problem Definition: Write a program to reverse the integer array without using loops? Using recursion?

* Answer

### **Program – 15**

Problem Definition: Write a program to perform Overriding methods in parent child

* Answer

### **Program – 16**

Problem Definition: Write a Java 8 code to return the student object if gender is male? (filter)

* Answer

### **Program – 17**

Problem Definition: Write a program using string, use only single for loop?

Input - String str = "Siddhant Patni";

Output - SiindtdahPata

* Answer

### **Program – 18**

Problem Definition: Write a program to print all the subsets of the given set with sum equal to given sum

set of numbers = {3, 35, 56, 2, 95, 10, 65, 150, 165, 23, 65, 18, 57}

sum = 28

* Answer

### **Program – 19**

Problem Definition: Write a program to count set bits in an integer

Example : 13 -> 1101 Print number of 1’s in the given binary number

* Answer

### **Program – 20**

Problem Definition: Write a program to balancing of the brackets

Example : Input - {[(a+b)+c]+x+y]}

* Answer

### **Program – 21**

Problem Definition: Write a program to print prime factors for given numbers?

* Answer

### **Program – 22**

Problem Definition: Write a program to print prime numbers between two numbers? Example: 24 to 100

* Answer

### **Program – 23**

Problem Definition: Write code to iterate ArrayList in different ways

* Answer

### **Program – 24**

Problem Definition: Write a program in java to reverse a singly linked list

* Answer

### **Program – 25**

Problem Definition: Write a program to print the first occurrence of a repeating element from integer Array?

Input: [6,10,7,8,9,7,11,12]

Output: 7

* Answer

### **Program – 26**

Problem Definition: Write a program to print occurrence of each character in the given string

Example: String str = "This is an interview is going on with Amdocs";

* Answer

### **Program – 27**

Problem Definition: Merge two sorted linked lists

Input: list1 = 10, 20,30,40;

list2 = 11,13,14,21,22,33,35;

Output: Mergedlist = 10,11,13,20,21,22,30,33,35,40

* Answer

### **Program – 28**

Problem Definition: Sort array with decreasing frequency of element

Input - 9,5,6,9,6,1,.2,9

Output - 9,9,9,6,6,1,2

* Answer

### **Program – 29**

Problem Definition: Write a program to remove the duplicate values from HashMap and return the max key from the HashMap.

* Answer

### **Program – 30**

Problem Definition: Write a program to return the maximum occurrence element from the array?

Input: int[] arr = {1,5,3,5,6,5};

Output: 5

* Answer

### **Program – 31**

**Problem Definition:** You are given a list of student test scores, where each entry contains a student’s name and their test score as a string array: [student\_name, test\_score]. Each student may appear multiple times in the list with different scores.

Your task is to write a Java function that calculates the **highest average score** among all students. If a student appears multiple times, their average score should be computed by averaging all of their scores. In case the average is a floating-point number, return the **floor value** of the average (i.e., round down to the nearest integer). If the input list is empty, return 0

* Answer

public class Solution {

public static int bestAverageGrade(String[][] scores) {

if (scores == null || scores.length == 0) return 0;

Map<String, List<Integer>> studentScores = new HashMap<>();

// Step 1: Populate the map with scores per student

for (String[] entry : scores) {

String name = entry[0];

int score = Integer.parseInt(entry[1]);

studentScores.putIfAbsent(name, new ArrayList<>());

studentScores.get(name).add(score);

}

int bestAverage = Integer.MIN\_VALUE;

// Step 2: Calculate averages and track the max

for (Map.Entry<String, List<Integer>> entry : studentScores.entrySet()) {

List<Integer> marks = entry.getValue();

int sum = 0;

for (int score : marks) {

sum += score;

}

int average = (int) Math.floor((double) sum / marks.size());

bestAverage = Math.max(bestAverage, average);

}

return bestAverage == Integer.MIN\_VALUE ? 0 : bestAverage;

}

// Optional main method for testing

public static void main(String[] args) {

String[][] input = {

{"Bobby", "87"},

{"Charles", "100"},

{"Eric", "64"},

{"Charles", "22"}

};

System.out.println(bestAverageGrade(input)); // Output: 87

}

}

### **Program – 32**

Problem Definition: Write a program to return the number of occurrences for a given word in the string.

Accept the inputs from the user.

* Answer

### **Program – 33**

Problem Definition: Write a java 8 program to print words which start with 'S'?

Example: List<String> list = Arrays.asList("Siddhant", "Patni", "Kotak", "Mahindra");

* Answer

### **Program – 34**

Problem Definition: Write a program to Swap two numbers without using a third variable?

* Answer

### **Program – 35**

Problem Definition: Write a program to check Palindrome number. Input : 59095

* Answer

### **Program – 36**

Problem Definition: Write a program to reverse String without using reverse function (Use brute force approach)

* Answer

### **Program – 37**

Problem Definition: Write a program to display the count of each word in a string?

Input: str = "w1 w2 w2 w3 w3 w4”

* Answer

### **Program – 38**

Problem Definition: Write a java 8 program to make all elements of the list to uppercase and create a new ArrayList

* Answer

public static void main(String[] args) {

//String str = "w1 w2 w2 w3 w3 w4";

List<String> list = Arrays.asList("w1", "w2", "w2","w3","w4");

List<String> newList = list.stream().map(str -> str.toUpperCase()).forEach(str ->

System.out.println(str + " "));

}

### **Program – 39**

Problem Definition: Write a program in java to print PreOrder Traversal

Input: Given the root of a binary tree, return the preorder traversal of its nodes' values.

* Answer

### **Program – 40**

Problem Definition: Write a program to calculate average marks of students and return the student object with average marks

* Answer

### **Program – 41**

Problem Definition: Write a program to get the count of each character in string

* Answer

### **Program – 42**

Problem Definition: Write a java 8 code to get the age greater than 15 from the Employee object?

Input - Employee

id

name

Age

* Answer

### **Program – 43**

Problem Definition: Write a program to get a list of files and directories by passing the path of the directory

* Answer

### **Program – 44**

Problem Definition: What is the output of the below snippet if we print str?

String str =”India”;

str = “Bharat”;

* Answer

### **Program – 45**

Problem Definition: Write a program to print Fibonacci series of nth number?

* Answer

### **Program – 46**

Problem Definition: Write a program to get the count of each word in a given string?

* Answer

### **Program – 47**

Problem Definition: Write a program - How to check whether a number is an Armstrong or not?

* Answer

### **Program – 48**

Problem Definition: Write a program to print a possible set of palindromes of a given string?

Input: ABBAAAAABBA

Output: B, BB, AA, AAA, AAAA, AAAAA, ABBA, BAAAAAB, BBAAAAABB,

ABBAAAAABBA

* Answer

### **Program – 49**

Problem Definition: Write a program to print the reverse of a given number? Input - 1234

* Answer

### **Program – 50**

Problem Definition: Write a program to return the minimum number of platforms required for a train?

Input:

12:05 12:50

13:10 14:25

05:30 07:25

05:35 05:55

02:15 02:20

* Answer

public static int getPlatformDetails(int[] arrival, int[] departure)

{

Arrays.sort(arrival);

Arrays.sort(departure);

int platform=0, finalPlatform=INT\_MIN, i=0, j=0, count =0;;

while(i <arrival.length)

{

if(arrival[i] < departure[j])

{

Platform = Integer.max(platform, ++count);

i++;

} else

{

Count --;

j++;

}

finalPlatform = Integer.max(platform, ++count);;

//here update result

}

return finalPlatform;

for(int i=0; i<num; i++)

{

for(int j= i +1; j< num;j++)

{

if(arrival[i] >= arrival[j] && arrival[i] <= departure[j] )

}

}

}

Public static void main(String[] args)

{

int arrival[] = {1205, 1310, 0530,0535 0215};

int departure[] = {1250, 1425, 0725, 0555, 0220};

int num = 5;

System.out.println(“Platform required are :”+getPlatformDetails(arrival, departure, num));

}

### **Program – 51**

Problem Definition: Write a Java program to count islands in a boolean 2D matrix.

Input :

0 0 1 10

0 0 1 0 0

1 0 0 0 0

0 1 0 0 0

1 1 0 1 0

3 connected island

* Answer

### **Program – 52**

Problem Definition: Write a program to shorten the given URL?

Input: https://docs.google.com/document/d/1g1CwT8dhCGgjh1LtXSaRBZ1ctREqkldD76lAU4KxPOo/edit

* Answer

### **Program – 53**

Problem Definition: Write a java program to find element closest to given target

Input - int arr[] = { 1, 2, 4, 5, 6, 6, 8, 9 };

int target = 11;

Output - 9

* Answer

### **Program – 54**

Problem Definition: If I want to remove the duplicate values from ArrayList then how can we achieve it using java 8?

List<Integer> list = new ArrayList<>(Arrays.asList(1, 10, 1, 2, 2, 3, 10, 3, 3, 4, 5, 5));

List<Integer> newList = list.stream().distinct().collect(Collectors.toList());

### **Program – 55**

Problem Definition: If you have an array of integers then how will you sort it? Write actual logic without using inbuilt Methods?

### **Program – 56**

Problem Definition: Write a Java function to count the number of distinct ways to reach the nth stair by climbing either 1 or 2 steps at a time, and return the result modulo 10^9+7.

public class StairClimber {

static final int MOD = 1\_000\_000\_007;

public static int countWays(int n) {

if (n <= 1) return 1;

int prev = 1, curr = 1;

for (int i = 2; i <= n; i++) {

int temp = (prev + curr) % MOD;

prev = curr;

curr = temp;

}

  return curr;

}

// Test the function

public static void main(String[] args) {

int n1 = 4;

int n2 = 10;

System.out.println("Ways to climb " + n1 + " stairs: " + countWays(n1)); // Output: 5

System.out.println("Ways to climb " + n2 + " stairs: " + countWays(n2)); // Output: 89

}

}

### **Program – 57**

Problem Definition: There is a stair having total no of stairs as n. Requirement is to reach at the top in minimum hop. Each step has some value associated with it which shows how many steps you can move forward from that step. Find the minimum hop required to reach from 0th step to nth step. n= 6 and stairs = {2,0,1,3,1,1};

* Answer

### **Program – 58**

Problem Definition: How will you sort the Student data based on their Age?

* Answer

### **Program – 59**

Problem Definition: Explain the logic of frequency of each character of String?

* Answer

### **Program – 60**

Problem Definition: Write a code to print the count of unique characters from the String str = “Siddhant Patni”?

* Answer

### **Program – 61**

Problem Definition: Given an integer array nums, return all the triplets [nums[i], nums[j], nums[k]] such that:

i, j, and k are distinct indices, nums[i] + nums[j] + nums[k] == 0. The solution set must not contain duplicate triplets. Input: nums = [-1,0,1,2,-1,-4]

Output: [ [-1,-1,2], [-1,0,1] ]

What is Time complexity and Space complexity?

How can you optimize the time complexity?

* Answer

### **Program – 62**

Problem Definition: Input: nums = [2,7,11,15], target = 9 Output: [0,1]

Explanation: nums[0] + nums[1] = 2 + 7 = 9, so return [0, 1].

How will you optimize the time complexity?

What is space complexity?

* Answer

### **Program – 63**

Problem Definition: Explain the output line by line in below code?

class Parent

{

public Parent()

{

System.out.println("in parent Constructor");

}

public void ml()

{

System.out.println("in parent method m1");

}

public static void m2()

{

System.out.println("in parent method m2");

}

}

Class Child extends Parent

{

Public Child()

{

System.out.println("in child Constructor");

}

public void ml()

{

System.out.println("in child method m1");

}

public static void m2()

{

System.out.println("in parent method m2");

}

}

Class Main

{

public static void main (String[] args)

{

Parent p = new Parent();

p.ml();

p.m2();

Child c = new Child();

c.ml();

c.m2();

Child ch = new Parent();

ch.ml();

ch.m2();

Parent pl= new Child();

pl.ml();

pl.m2();

}

}

* Answer

### **Program – 64**

Problem Definition: Write a code to add into the list and it should sort on the basis Address?

class Emp {

int id, str name , Address addr

Emp(id,name, addr){id= this.id, name =this.name, this.addr= addr}

}

class test{

Emp e1 = new Emp(1,'A','pune');

Emp e2 = new Emp(1,'A','nagpur');

.

.

Emp e100 = new Emp(1,'A','zz');

* Answer

### **Program – 65**

Problem Definition: What is the output?

public class Employee {

int id; String name;

Employee(int id, String name){

this.id = id;

this.name = name;

//this.addr = add;

}

}

class test{

public static void main(String[] args)

{

Employee e1 = new Employee(1,"A");

Employee e2 = new Employee(1,"A");

Employee e3 = new Employee(1,"A");

Employee e4 = new Employee(1,"A");

Employee e5 = new Employee(1,"A");

HashMap<Employee,Employee> h = new HashMap();

h.put(e1,e1);

.

.

h.put(e5,e5);

System.out.println("HS size: "+h.size());--1

}

}

* Answer

### **Program – 66**

Problem Definition: What is the output?

public class Employee {

int id; String name;

Employee(int id, String name){

this.id = id;

this.name = name;

//this.addr = add;

}

}

class test{

public static void main(String[] args)

{

Employee e1 = new Employee(1,"A");

Employee e2 = new Employee(1,"A");

Employee e3 = new Employee(1,"A");

Employee e4 = new Employee(1,"A");

Employee e5 = new Employee(1,"A");

HashSet<Employee,Employee> h = new HashSet();

h.add(e1,e1);

.

.

h.add(e5,e5);

System.out.println("HS size: "+h.size());--1

}

}

* Answer

### **Program – 67**

Problem Definition: What will be the output of the below program?

class Simple

{

Simple ( )

{

System.out.println("l am in Simple");

}

public static void main (string[] args)

{

Simple s = new Simple() ;

}

}

* Answer

### **Program – 68**

Problem Definition: Write a program to find the first occurrence of non-repeating character in String

Input: “aabbcdec”

Output: d

* Answer

### **Program – 69**

Problem Definition: Develop a Java program that models a Cinema Booking System with the following

Requirements:

An Auditorium containing multiple Screens.

Each Screen hosts multiple Movie Shows.

Each Movie Show offers seating in three categories: Gold, Silver, and Platinum.

Implement functionality to book tickets, check seat availability, and include additional relevant

methods to enhance the booking system.

* Answer

### **Program – 70**

Problem Definition: Movie booking system related java codes, check availability, book ticket with multiple classes and use of data structures?

* Answer

### **Program – 71**

Problem Definition: Explain the implementation of the User registration method using java? Explain how will you implement it?

* Answer

### **Program – 72**

Problem Definition: Write a method to register and check the available user in repo?

* Answer

### **Program – 73**

Problem Definition: How can you handle WhatsApp notification and mobile notification after user registration?

* Answer

### **Program – 74**

Problem Definition: Write a Java code, we have Product and Feature class with Product having model, color, price, location and List<Feature> and Feature class having picture and resolution fields, Write a code using Java 8 Stream API?

1. Print Product which is iPhone 16

2. get the product and its feature which has highest resolution

3. Print number of iPhone15 and iPhone16 count in each location

* Answer

### **Program – 75**

Problem Definition: We have 2 Employee object and if I add these objects in Set and check for the size then what will be the output 1 or 2?

Employee e1 = new Employee(“Swapnil”);

Employee e2 = new Employee(“Swapnil”);

* Answer

### **Program – 76**

Problem Definition:

* Answer

# Advance Java / J2EE

### **What is servlet?**

* Answer

### **Explain servlet lifecycle?**

* Answer

### **What is a dispatcher servlet?**

* answer

### **What is JSP?**

* answer

### **What is the difference between servlet and JSP?**

* answer

### **What is the JSP lifecycle?**

* answer

### **What is Spring MVC? Explain architecture of spring MVC?**

* Answer

### **What is the singleton design pattern?**

* answer

### **What is the REST API?**

* answer

### **Difference between API and REST?**

* answer

### **Difference between SOAP and REST?**

* answer

### **Which among SOAP and REST is easy?**

* answer

### **Explain REST architecture?**

* answer

### **What do you know about DTD and XSD?**

* answer

### **Explain SOAP Web Service?**

* answer

### **What is Path in XML?**

* answer

### **Write a path for a given xml file?**

* answer

### **What is the use of XSLT?**

* answer

### **How does SOAP work?**

* answer

### **What are the action types of http?**

* answer

### **Explain how you will create a SOAP web service?**

* answer

# Data Structure

### **Find the middle element from the singly linked list without finding the length?**

* Answer

### **Complexity to find length of singly linked list?**

* Answer

### **Do you know how LinkedList works? Can you create a data structure for LinkedList?**

* answer

### **How will you traverse LinkedList? Your List will contains multiple nodes so how will you traverse over it? Can you write a pseudo code? You have your head with you so how will you move to the next node? How will the current node be null?**

* answer

### **Explain Merge Sort??**

* answer

# Design Patterns

### **What is the singleton design pattern? Explain it?**

* answer

### **Explain Factory design pattern?**

* answer

### **What is the Observer design pattern?**

* answer

### **Except for the creational design pattern, can you explain other design patterns?**

* answer

### **Explain Saga design pattern?**

* answer

# Spring

### **Explain Spring MVC architecture?**

* Answer

### **Explain Spring IOC?**

* answer

### **Explain Spring Dependency Injection?**

* answer

### **What are the spring bean scopes?**

* answer

### **Explain any scope of spring bean?**

* Answer

### **How to add scope in bean configuration?**

* Answer

### **Difference between session and prototype scope?**

* Answer

### **What is the Spring batch? Explain where you have used the spring batch in your project?**

* answer

### **How spring batch works?**

* answer

### **How to implement the spring batch?**

* Answer

### **What is the difference between SOAP and REST? Which is more efficient?**

* Answer

### **How REST API works?**

* Answer

### **Difference between Spring and Spring Boot?**

* Answer

### **Difference between application context and beanfactory in spring?**

* Answer

### **Explain Spring MVC architecture?**

* Answer

### **Difference between Spring and Spring Boot?**

* Answer

### **What is the use of @Transactional annotation in spring?**

* Answer

### **What are different stereotypes annotations in spring?**

* Answer

### **What is the front controller?**

* answer

### **Difference between applicationcontext and beanfactory?**

* answer

### **Explain spring bean life cycle?**

* answer

### **If I have prototype bean and singleton bean, then how can we call prototype beans inside singleton bean?**

* answer

### **What are containers present in the spring?**

* answer

### **How transactions work in spring?**

* answer

### **Do you know the web.xml file? What is the significance of a web.xml file?**

* answer

### **What is the dependency injection?**

* answer

### **What are the different ways of injection?**

* answer

### **Who will take care of objects in Spring?**

* answer

### **What is ApplicationContext and its types in Spring?**

* answer

### **Difference between Filter and Interceptor in Spring MVC?**

* answer

### **Which one has access to spring beans, interceptors or filters?**

* answer

### **Why do interceptors have access to Spring beans?**

* answer

### **Difference between Spring data JPA, Hibernate, JDBC, My batis?**

* answer

### **Why do we use Spring data JPA?**

* answer

# Hibernate

### How data is persisted in database by using hibernate**?**

* Hibernate is an ORM (Object-Relational Mapping) framework that simplifies database operations in Java applications. Here's how data gets persisted to a database using Hibernate:

1. Configuration Setup

First, configure Hibernate to connect to your database:

<!-- hibernate.cfg.xml -->

<hibernate-configuration>

<session-factory>

<!-- Database connection settings -->

<property name="hibernate.connection.driver\_class">com.mysql.jdbc.Driver</property>

<property name="hibernate.connection.url">jdbc:mysql://localhost:3306/mydb</property>

<property name="hibernate.connection.username">root</property>

<property name="hibernate.connection.password">password</property>

<!-- SQL dialect -->

<property name="hibernate.dialect">org.hibernate.dialect.MySQLDialect</property>

<!-- Mapping files -->

<mapping class="com.example.Employee"/>

</session-factory>

</hibernate-configuration>

2. Entity Class Definition

Create a Java class annotated with @Entity:

@Entity

@Table(name = "employees")

public class Employee {

@Id

@GeneratedValue(strategy = GenerationType.IDENTITY)

private Long id;

@Column(name = "emp\_name")

private String name;

private double salary;

// Getters and setters

}

3. Persistence Process

Here's the step-by-step persistence flow:

a) Create SessionFactory

Configuration config = new Configuration().configure("hibernate.cfg.xml");

SessionFactory sessionFactory = config.buildSessionFactory();

b) Open Session

Session session = sessionFactory.openSession();

c) Begin Transaction

Transaction tx = session.beginTransaction();

d) Create Entity Object

Employee emp = new Employee();

emp.setName("John Doe");

emp.setSalary(50000);

e) Save/Persist the Object

session.save(emp); // or session.persist(emp);

f) Commit Transaction

tx.commit();

g) Close Session

session.close();

4. What Happens Internally

When you call **session.save():**

Hibernate checks the entity state - determines if it's transient (new), persistent (managed), or detached

Generates SQL - Hibernate creates the appropriate INSERT statement based on your entity mappings

Executes SQL - The statement is executed against the database

Updates entity state - The transient object becomes persistent

Handles ID generation - If using auto-increment, retrieves the generated ID

5. Transaction Management

Hibernate operations should be performed within transactions:

Session session = sessionFactory.openSession();

try {

Transaction tx = session.beginTransaction();

// Perform operations

Employee emp = new Employee("Jane Smith", 60000);

session.save(emp);

tx.commit(); // Changes are flushed to database here

} catch (Exception e) {

if (tx != null) tx.rollback();

throw e;

} finally {

session.close();

}

6. Different Persistence Operations

**Operation Description**

save() Persists the object, returns generated ID

persist() Similar to save() but doesn't guarantee immediate ID assignment

update() Updates a detached object

merge() Copies state of detached object to persistent object

saveOrUpdate() Either saves or updates based on object state

7. Hibernate Cache Flow

**First Level Cache (Session cache):**

* Exists per session
* All persistent objects are stored here
* Cleared when session closes

**Second Level Cache (Optional):**

* Shared across sessions
* Needs explicit configuration
* Reduces database hits for frequently accessed data

Example Complete Flow

public class HibernateExample {

public static void main(String[] args) {

// 1. Configure and build SessionFactory

SessionFactory sessionFactory = new Configuration()

.configure("hibernate.cfg.xml")

.buildSessionFactory();

// 2. Open session

Session session = sessionFactory.openSession();

Transaction tx = null;

try {

// 3. Begin transaction

tx = session.beginTransaction();

// 4. Create and persist object

Employee newEmployee = new Employee();

newEmployee.setName("Michael Johnson");

newEmployee.setSalary(75000);

// 5. Save to database

Long employeeId = (Long) session.save(newEmployee);

System.out.println("Employee saved with ID: " + employeeId);

// 6. Commit transaction

tx.commit();

} catch (Exception e) {

if (tx != null) tx.rollback();

e.printStackTrace();

} finally {

// 7. Close session

session.close();

sessionFactory.close();

}

}

}

**Key Benefits of Hibernate Persistence:**

* Object-Oriented Approach - Work with objects rather than SQL
* Automatic SQL Generation - No need to write CRUD queries
* Transaction Management - Built-in support for ACID properties
* Caching - Improves performance
* Database Independence - Switch databases with minimal code changes

This is the fundamental process of how Hibernate persists Java objects to relational databases while handling all the underlying JDBC complexity.

### **How to retrieve select query using Hibernate?**

* Answer

### **How to retrieve data through hibernate?**

* Answer

### **Hibernate caching?**

* Answer

### **Explain L1 and L2 cache in Hibernate?**

* Answer

### **Explain the hibernate sessions?**

* Answer

### **Difference between JDBC and Hibernate?**

* Answer

### **What is the difference between get and load method in hibernate?**

* Answer

### **Explain bean scopes?**

* Answer

### **Can we have beans without bean id?**

* Answer

### **Difference between get and load method?**

* Answer

### **Explain Performance tuning and indexing in hibernate?**

* Answer

### **Explain different classes and annotations in hibernate?**

* Answer

### **How to call stored procedures using hibernate?**

* answer

### **What is Many to many associations in hibernate?**

* answer

### **What is JPA?**

* answer

### **What is the many to many associations in hibernate?**

* answer

### **What is a Session factory in hibernate?**

* answer

### **Explain Many to many associations in hibernate?**

* answer

### **If I want to retrieve a few entries from an employee table, how to retrieve employee details through JPA?**

* answer

### **Explain a built-in function in JPA?**

* answer

### **What is lazy loading in hibernate?**

* answer

### **What is circular dependency?**

* answer

### **What is the second level cache in hibernate?**

* answer

### **Can we use Hibernate without JPA?**

* answer

### **What are the advantages of ORM?**

* answer

# Spring Boot

### **Difference between Spring and Spring Boot?**

* Answer

### **What are the advantages of SprigBoot?**

* Answer

### **What is a yaml file in SpringBoot?**

* Answer

### **What is what is @transient in spring boot?**

* answer

### **What are the types of propagation?**

* Answer

### **What @Qualifier in spring? What is the use of @Qualifier?**

* answer

### **What are the advantages of Spring boot?**

* answer

### **How to create multiple modules using spring boot?**

* answer

### **What are all methods present in the controller?**

* answer

### **Explain annotations in Spring boot?**

* answer

### **What is the query parameter?**

* answer

### **How to change the server in Spring boot?**

* answer

### **Difference between @Controller and @RestController?**

* answer

### **What is the use of @Responsebody annotation?**

* answer

### **Explain @RestController annotation in Spring boot?**

* answer

### **Which two annotations are used in built by @RestController?**

* answer

### **How to access application.properties properties in java code?**

* answer

### **Explain Spring boot application? How does it work?**

* answer

### **What is the use of @Service annotation in spring?**

* answer

### **What is the use of @Component annotation in spring?**

* answer

### **What is the use of @Repository annotation in spring?**

* answer

### **What is the difference between @Service and @Controller annotation?**

* answer

### **What is @ComponentScan?**

* answer

### **What is the use of @Autowired annotation?**

* answer

### **Write a REST API to accept user name as input, return the response with message as "Hello, user" and status as "success" and Endpoint - /api/message?**

* answer

### **How to handle Exceptions in Spring boot?**

* answer

### **How to provide security to spring boot applications?**

* answer

### **What is @ControllerAdvice in Spring boot?**

* answer

### **How to retrieve query parameters in Spring boot?**

* answer

### **How is the data persisted in DB with the REST API?**

* answer

### **Explain GET and POST methods?**

* answer

### **Write a SQL or JPQL to retrieve the data from DB?**

* answer

### **How will you add Employee object data in the DB using POST call?**

* answer

### **What is the size of Spring in your project? Who will decide that?**

* answer

### **Explain about spring security?**

* answer

### **Explain the flow of REST api?**

* answer

### **What will happen when we replace @Service instead of @Repository?**

* answer

### **Explain uses of SpringBoot application?**

* answer

### **How is https used in SpringBoot applications?**

* answer

### **Difference between RequestMapping and GetMapping?**

* answer

### **If you have to create a Spring boot application? What steps will you follow?**

* answer

### **What is a starter kit in spring boot application?**

* answer

### **What are the starters in Spring boot application?**

* answer

### **What are Actuators in spring boot applications?**

* answer

### **What are custom exceptions? How do you define your own custom exception?**

* answer

### **Have you worked on Unit testing? Which framework have you used in your project?**

* answer

### **What is the difference between lazy loading and eager loading?**

* answer

### **Difference between put and patch?**

* answer

### **Can we use @RequestMapping at class level and method level both?**

* Answer

### **What will happen if I use @RequestMapping at method level?**

* answer

### **How to map Service class to RestController?**

* answer

### **What is Spring security? Have you used Spring security in your project?**

* answer

### **What are Profiles in Spring boot?**

* answer

### **How will you do CRUD operations using spring boot applications?**

* answer

### **How to change from JAR to WAR in Spring boot?**

* answer

### **Where will you give database details in the spring boot application?**

* answer

### **How will you give database details for the individual environment?**

* answer

### **What is a JPA repository?**

* answer

### **What is the use of @SpringBootApplication annotation in Spring boot?**

* answer

### **How will the Spring boot application know whether its JAR or WAR type?**

* answer

### **How are you providing security to your services?**

* answer

### **As an end user how can I consume your services?**

* answer

### **How to validate a token?**

* answer

### **Explain a few security algorithms?**

* answer

### **What is native queries in Spring boot?**

* answer

### **Where do you use native queries?**

* answer

### **Suppose I am having different microservices, if I want to invoke different microservice so how can we interact?**

* answer

### **What kind of information are you passing to the GET method?**

* answer

### **Suppose I want to start a spring boot application, so what are all the steps?**

* answer

### **How to configure a custom server in a Spring boot application?**

* answer

### **How to configure multiple databases in a Spring boot application?**

* answer

### **How does your application know I am using a particular database?**

* answer

### **What is the use of @Qalifier annotation?**

* answer

### **How are REST calls intercepted in our project? How is the call happening?**

* answer

### **When I hit a URL how data will be invoked with your application?**

* answer

### **How is the DTO getting invoked with REST endpoints?**

* answer

### **I have a URL and I hit the URL, so which is the first class which is invoked in our product?**

* answer

### **If you have to expose a new endpoint, what steps will you follow?**

* answer

### **What is the difference between @Bean and @Configuration?**

* answer

### **What is the difference between RequestParam and path variable? Give me one example with one URL?**

* answer

### **How to change the port of the embedded tomcat server in Spring boot? Write a syntax for that?**

* answer

### **What are Profiles in Spring boot applications? Give one example?**

* answer

### **Which annotations are we using in order to use the Exception?**

* answer

### **What is the default scope for spring beans?**

* answer

### **Which database have you used? How do you integrate databases with JPA?**

* answer

### **How to communicate between two spring boot applications using asynchronous calls?**

* answer

### **Have you worked messaging services?**

* answer

### **How will you handle errors / exceptions in your project?**

* answer

### **What are Actuators?**

* answer

### **How are you keeping track of logs in your project?**

* answer

### **If you have millions of data then how will you manage it?**

* answer

### **What is the difference between REST API and Http?**

* answer

### **Explain @Autowired usage?**

* answer

### **How does the REST API work?**

* answer

### **How can we handle a yaml file when it is exposed?**

* answer

### **How to externalize variables in spring boot?**

* answer

### **What is the Spring reactive framework?**

* answer

### **Why should we use Spring reactive?**

* answer

### **What is the use of block and subscribe?**

* answer

### **What is a novel based server?**

* answer

### **How will the findById or findByName method in the repository work with name? When we go with camelcase then where is it implemented internally?**

* answer

### **How does the JPA Repository know there is a class?**

* answer

### **How does the Spring Boot application work internally?**

* answer

### **How will you authenticate API in Spring Boot?**

* answer

### **In the Spring REST API what is the White label error accessing from the browser? What is the status code you will get?**

* answer

# Microservices

### **Monolith vs Microservices differences and when to choose which?**

* answer

### **How to design a microservice from scratch?**

* answer

### **API Gateway pattern and its advantages?**

* answer

### **Inter-service communication: REST vs Messaging?**

* answer

### **Circuit Breaker pattern and its implementation using Resilience4j?**

* answer

### **Load balancing in microservices using Spring Cloud Load Balancer?**

* Answer

### **How Spring Cloud Config helps in centralized configuration management?**

* answer

### **Service discovery using Eureka or Consul?**

* answer

### **Feign Client vs WebClient: Which one to use and why?**

* answer

### **Event-driven architecture and Kafka integration?**

* answer

### **Database per service vs Shared Database: Pros and cons?**

* answer

### **Saga Pattern for distributed transactions in microservices?**

* answer

### **JWT-based authentication and OAuth2 in microservices?**

* answer

### **How to handle security in an API Gateway?**

* answer

### **Observability: Logging, tracing, and monitoring best practices?**

* answer

### **Role of Prometheus and Grafana in microservices monitoring?**

* answer

### **Kubernetes deployment strategies for microservices?**

* answer

### **Blue-Green and Canary deployments in microservices?**

* answer

### **When to use WebFlux for reactive microservices?**

* answer

### **CQRS and Event Sourcing: When and why to use them?**

* Answer

### **Difference between Monolithic and Microservice architecture?**

* answer

### **Explain Microservices architecture?**

* answer

### **Where are you deploying your microservice?**

* answer

### **What are different Aggregation patterns in Microservices?**

* answer

### **What are challenges in Microservice architecture?**

* answer

### **How are you handling load in your microservices?**

* answer

### **How will you handle resilience and fault tolerance in microservice?**

* answer

### **Can you explain Event driven microservice and how you have used it in your project?**

* answer

# SQL

### **What is ordered by and range in the database?**

* Answer

### **Write a SQL query to get the second highest salary of an employee from the employee table?**

* Answer

### **Write a SQL query to print employee id, employee\_name, depatment\_name Employee - employee\_id, employee\_name, department\_id**

### **Department - department\_id, department\_name?**

* answer

### **What is a collision in the oracle database?**

* Answer

### **If you create multiple folders one inside another and store one file into the last folder then how can you implement using database tables in oracle? How many tables are required?**

* answer

### **What are DDL and DML commands?**

* Answer

### **Write a query to update data in the table?**

* Answer

### **What is the use of joins in the database?**

* Answer

### **Explain ACID properties in the database?**

* Answer

### **Explain Normalization?**

* Answer

### **What are the types of Normalization?**

* Answer

### **What is Denormalization?**

* Answer

### **What is the use of join?**

* Answer

### **Write a SQL query to return the 7th highest salary of an employee?**

* Answer

### **What is Union in SQL?**

* Answer

### **Can we perform union operations on one table?**

* Answer

### **Difference between where and let?**

* Answer

### **What is Semaphore?**

* Answer

### **What is View and Index in SQL?**

* Answer

### **What is Trigger in SQL? What is the use of Trigger in SQL?**

* Answer

### **What is inner join in SQL?**

* Answer

### **Write a SQL query to get emp\_sal > 5000 and company\_name = "Kotak"**

Table :

1. Company

- company\_id

- company\_name

2. Employee

- emp\_id

- company\_id

- emp\_name

- emp\_sal?

* Answer

Select e.emp\_sal, c.company\_name

From Employee e

join Company c

On c.company\_id = e.company\_id

Where e.emp\_sal > 5000 and c.company\_name = "Kotak";

### **Write a JPA query to retrieve the above result?**

* answer

### **What is an index in SQL?**

* answer

### **Explain different types of Index?**

* answer

### **Explain different types of joins?**

* answer

### **Difference between Outer join and Full outer join?**

* answer

### **Write a SQL query to find the name of a city which has more than one customer in it?**

Example: Customer

Customer\_name

Customer\_city?

* answer

### **Write SQL query to display 10th highest salary of employee?**

* answer

### **What is the difference between DDL and DML commands?**

* answer

### **Difference between Inner join and Outer join?**

* answer

### **Difference between left outer join and right outer join?**

* answer

### **What are Triggers?**

* answer

### **SQL query to get the youngest employee from the employee table?**

* answer

### **SQL query to get list of employees from Finance department?**

* answer

### **What is Indexing in SQL?**

* answer

### **What are the disadvantages of Index?**

* answer

### **What is the difference between primary key and unique key?**

* answer

### **What is the difference between implicit cursor and explicit cursor?**

* answer

### **What is a correlated subquery?**

* answer

### **What is the left outer join?**

* answer

### **What is the use of stored procedure?**

* answer

### **Explain the database schema for Students and Course table?**

* answer

### **What is UNION ALL?**

* answer

### **SQL query to get second highest salary?**

* answer

### **Write a SQL query to display a list of students which have a Mathematics subject?**

Input - Student

student\_id

student\_name

Subject\_id

Subject

subject\_id

Subject\_name

Where subject=Mathematics?

* answer

### **What is Normalization?**

* answer

### **Can you explain different forms of Normalization with one example?**

* answer

### **What kind of fetching strategy are you using in hibernate?**

* answer

### **What is OneToMany mapping?**

* answer

### **Write a SQL query to get the 3rd highest salary of an employee?**

* answer

### **What is dense and rank in SQL?**

* answer

### **What is the difference between SQL and NoSQL?**

* answer

### **Difference between Stored Procedure and Function?**

* answer

### **What is Transaction in SQL?**

* answer

### **Write a SQL query:**

Consider 2 tables, find out 3 employees who are earning the highest salary in each department?

Employees

employee\_id (primary key)

first\_name

last\_name

department\_id (foreign key to departments.id)

salary

Hire\_date

Departments

id (primary key)

department\_name?

* answer

### **SQL query that categorizes employees based on their salary into different percentage ranges.**

emp

id Name Salary

1 A 10

2 B 100

3 C 4

4 D 60

5 E 70

6 F 40

---------------------

id name sal Percentage

2 B 100 greater than 90

5 E 70 greater than 50

4 D 60 greater than 50

1 A 40 less than 50

3 C 10 less than 10

6 F 5 less than 10

--4 cate--

greater than 90

greater than 50

less than 50

less than 10?

* answer

### **Write a SQL query to find the department name who does not have an employee? (Employee and Department table given)?**

* answer

### **What is the difference between Primary Key and Unique Key?**

* answer

### **What is the Composite key?**

* answer

### **What is Self join?**

* answer

### **We have Table A (2 columns and 4 rows) and Table B (2 columns and 4 rows) and I want to print all 8 rows? Write a query for it? (Use Union All)?**

* answer

### **What is the difference between SQL Union and Union All?**

* answer

### **Write a SQL query and print the employee name which does not have any department?**

* answer

### **Write a query - Employee table and add one extra column in output as category? Let's say if salary is in 1 to 4000 then Category 1, 4000 - 7000 then category 2, 7000 - 10000 then category 3?**

* answer

# Kafka

### **What is Apache Kafka?**

* answer

### **Why does Kafka compare to RabbitMQ?**

* answer

### **How to scale up Kafka?**

* answer

### **What are the default partitions in kafka?**

* answer

### **Difference between topic and partition in Kafka?**

* answer

### **Is there any ordering of messages in partitions in kafka?**

* answer

### **How does partition assignment happen? What method is used?**

* answer

### **What is offset and partition in Kafka?**

* answer

# GitHub

### **What is Version control i.e. Git?**

* Answer

### **How to push and commit code in git using commands?**

* Answer

### **Difference between rebase and merge?**

* answer

### **Which versioning tool are you using? Explain a few Git commands?**

* answer

### **What is the Git command to create a new branch from an existing branch?**

* answer

### **If we have more consumer groups than partitions and how will it manage?**

* answer

# Docker

### **Explain use cases of Docker?**

* answer

### **You have 2 containers 1 and 2, can they communicate with each other?**

* answer

### **If both are with different hosts then whether they will be able to communicate with each other or not?**

* answer

### **What is the host in this case?**

* answer

### **Imagine you have one volume and you want to share it with multiple containers? How will you do it?**

* answer

### **Where does an object get stored when created inside a docker container?**

* answer

### **What is the process of Docker?**

* answer

### **How is Docker helpful?**

* answer

### **What is the use of Docker?**

* answer

# Kubernetes

### **What are Kubernetes Secrets??**

* answer

# AWS

### **What is cloud computing?**

* answer

### **Explain services provided by cloud computing?**

* answer

### **How to deploy service to AWS?**

* answer

### **What are services in AWS?**

* answer

### **Which services have you used?**

* answer

### **What are different services in EKS?**

* answer

### **What is the benefit of Serverless services? Is EKS serverless?**

* answer

# Azure

### **question?**

* answer

# Angular

### **What are the components in the Angular?**

* answer

### **How to pass data from components in Angular?**

* answer

### **What does the component.ts file contain in angular?**

* answer

### **Explain two way data binding in angular?**

* answer

### **How to make backend calls in angular?**

* answer

### **What is authguard in angular?**

* answer

### **What is interpolation in angular?**

* answer

### **How to create components in angular?**

* answer

### **Explain angular directives?**

* answer

### **What are observables in angular?**

* answer

### **How to implement dependency injection in angular?**

* answer

### **What are different components of Angular?**

* answer

### **How to handle caching in Angular?**

* answer

# React

### **Features of ReactJS?**

* answer

### **How does Virtual DOM work?**

* answer

### **Features of Virtual DOM?**

* answer

### **What is reducing modifications in Virtual DOM?**

* answer

### **What do you call the backend API from ReactJS?**

* answer

### **What are other Hooks in React?**

* answer

### **What is the purpose of Hook?**

* answer

### **What is the dependency array in useEffect?**

* answer

### **What is the use and significance of dependency arrays?**

* answer

### **How to make a call to an external API from ReactJS?**

* answer

### **What is an alternative to Axios?**

* answer

# Others

### **Explain Jira?**

* answer

### **What is the output of below program?**

#include<stdio.h>

int main()

{

int a =0;

1 + 1 - 1 + 1

a=a++ + ++a - a++ + ++a;

printf(“%d\n”,a);

return 0;

}

* answer

### **What is https protocol?**

* answer

### **How does https internally work?**

* answer

### **What is a certificate?**

* Answer

### **Difference between http and https?**

* answer

### **What is a Sprint retrospective?**

* answer

### **Do you know anything about CI/CD?**

* answer

### **How will Jenkins' job work? Explain the process?**

* answer

### **What is Node JS?**

* answer

### **How is your application getting deployed? Which platform are you using in your project?**

* answer

### **What do you know about Asynchronous communication?**

* answer

### **What are the measure objects i.e key things in order to make Asynchronous calls?**

* answer

### **If I have two different technologies then how can we achieve asynchronous communication?**

* answer

### **Do you have exposure in AI and ML?**

* answer

### **Examples of Fund Transfer are Synchronous or Asynchronous?**

* answer

### **Is WhatsApp Synchronous or Asynchronous?**

* answer

### **Difference between 2 Tier and 3 Tier Architecture?**

* answer

### **Have you heard about the log4j vulnerability?**

* answer

### **What is https protocol?**

* answer

### **Difference between http and https?**

* answer

### **Suppose in server one job at 9 AM and copy json file and paste it in temp folder another job 6 PM read it and load in db and no one should change the file. How to prevent file from any Change?**

* answer

### **If we increase timeout then how can we resolve it? Deployment related issue and how will you handle it?**

* answer

# Linux

### **Explain Linux commands?**

* answer

### **Command - Read a file and print first two lines from file Ans. head -2 filename?**

* answer

### **Search a word from a file?**

* grep 'word' filename ack 'pattern' /path/to/file.txt

### **Why do you want to change your current organization?**

* answer

# Puzzles

### **Puzzle – 1**

Weighing the 9 balls puzzle, find the heavier ball among 9 balls. How many max iterations will it be required to find the heavier ball?

* answer

### **Puzzle – 2**

You are doing some gardening, and need exactly 4 litres of water to mix up some special formula for your award-winning roses. But you only have a 5-liter and a 3-liter bowl, but do have access to plenty of water.

How would you measure exactly 4 litres?

* Fill the 5-liter bowl. Then fill the 3-liter bowl from the 5-liter bowl. You will now have 2 litres left in the 5 litres bowl. Empty the 3-liter bowl, and then transfer the 2 litres from the 5-liter bowl into it. Now fill the 5-liter bowl again, then pour water carefully from the 5-liter bowl into the 3-liter bowl until it is full - exactly one more litre. The 5-liter bowl now has exactly 4 litres.

### **Puzzle – 3**

Divide square in 5 equal parts?