***Technical Concepts Handbook***

***- Siddhant Patni***

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

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

Contents

[***Java*** 20](#_Toc195360641)

[ **Explain how to install java? Explain directories in it?** 20](#_Toc195360642)

[ **How to configure java in windows?** 20](#_Toc195360643)

[ **Explain JDK, JRE?** 20](#_Toc195360644)

[ **Explain JVM?** 20](#_Toc195360645)

[ **Explain Java Program execution?** 20](#_Toc195360646)

[ **Explain the byte stream in java?** 22](#_Toc195360647)

[ **What is Classloader? Different types of it?** 22](#_Toc195360648)

[ **Why is java more secure?** 22](#_Toc195360649)

[ **Explain public static void main (String[] args) method?** 22](#_Toc195360650)

[ **What is the marker interface?** 22](#_Toc195360651)

[ **What is serialization and deserialization?** 22](#_Toc195360652)

[ **What is the use of the transient keyword in serialization?** 22](#_Toc195360653)

[ **What is the final class?** 22](#_Toc195360654)

[ **How to create custom exceptions?** 22](#_Toc195360655)

[ **Why is string immutable in java?** 23](#_Toc195360656)

[ **What is StringBuffer?** 23](#_Toc195360657)

[ **What is StringBuilder?** 23](#_Toc195360658)

[ **Difference between String, StringBuffer and StringBuilder?** 23](#_Toc195360659)

[ **What is the final keyword in java?** 23](#_Toc195360660)

[ **Where can we use the final keyword?** 23](#_Toc195360661)

[ **What will happen if I create the final class?** 23](#_Toc195360662)

[ **What is the difference between comparable and comparator?** 23](#_Toc195360663)

[ **Explain OOPs concept?** 23](#_Toc195360664)

[ **What is an object?** 23](#_Toc195360665)

[ **Explain method in Object class? Explain 9 methods in Object class?** 23](#_Toc195360666)

[ **How many ways can we create objects of class in java?** 23](#_Toc195360667)

[ **What is Encapsulation? Explain with a real time example** 24](#_Toc195360668)

[ **How do we achieve encapsulation in Java?** 24](#_Toc195360669)

[ **What is abstract in java?** 25](#_Toc195360670)

[ **What is static in java?** 26](#_Toc195360671)

[ **What is the difference between HashMap and HashSet** 27](#_Toc195360672)

[ **Explain Java8 features?** 27](#_Toc195360673)

[ **What is Functional Interface?** 27](#_Toc195360674)

[ **What is the default method in functional interface?** 27](#_Toc195360675)

[ **What is the map function in stream?** 27](#_Toc195360676)

[ **What is the filter in the stream?** 27](#_Toc195360677)

[ **What are the methods of thread?** 28](#_Toc195360678)

[ **What is thread pool? What are the types of thread pool?** 28](#_Toc195360679)

[ **What are thread pool methods?** 28](#_Toc195360680)

[ **What is method overriding?** 28](#_Toc195360681)

[ **How to achieve method overriding?** 28](#_Toc195360682)

[ **What is method overloading?** 28](#_Toc195360683)

[ **What is Polymorphism? Explain with an example?** 28](#_Toc195360684)

[ **What is Inheritance?** 28](#_Toc195360685)

[ **What is the interface in java?** 28](#_Toc195360686)

[ **Can we have a static method inside the interface?** 28](#_Toc195360687)

[ **Can we override the static method?** 28](#_Toc195360688)

[ **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?** 28](#_Toc195360689)

[ **Difference between ArrayList and LinkedList?** 28](#_Toc195360690)

[ **Which is more efficient among the ArrayList and LinkedList?** 28](#_Toc195360691)

[ **Which among ArrayList and LinkedList will be more efficient for random access of data?** 28](#_Toc195360692)

[ **How to sort the ArrayList?** 29](#_Toc195360693)

[ **How to sort the ArrayList?** 29](#_Toc195360694)

[ **Difference between ArrayList and Vector?** 29](#_Toc195360695)

[ **What is HashMap?** 29](#_Toc195360696)

[ **Explain internal working of HashMap?** 29](#_Toc195360697)

[ **What is hash collision in HashMap?** 29](#_Toc195360698)

[ **What are the types of HashMap?** 29](#_Toc195360699)

[ **Can we store null value in HashMap?** 29](#_Toc195360700)

[ **Can we store infinite data in a HashMap?** 29](#_Toc195360701)

[ **Can we access null value in HashMap?** 29](#_Toc195360702)

[ **What is exception handling?** 29](#_Toc195360703)

[ **Explain the hierarchy of Exception?** 29](#_Toc195360704)

[ **What is the difference between checked exception and unchecked exception?** 29](#_Toc195360705)

[ **How to handle unchecked exceptions?** 29](#_Toc195360706)

[ **Explain Generics? What is the use of Generics?** 29](#_Toc195360707)

[ **What is composition and aggregation?** 29](#_Toc195360708)

[ **Parent and child class methods and how to access those?** 29](#_Toc195360709)

[ **What is the use map method in java 8 stream?** 30](#_Toc195360710)

[ **Which collection you will use in order to remove duplicate elements and preserve the insertion order?** 30](#_Toc195360711)

[ **How to find the second largest element from the list?** 30](#_Toc195360712)

[ **What is the difference between Map and FlatMap in stream?** 30](#_Toc195360713)

[ **Where can you use FlatMap in a stream?** 30](#_Toc195360714)

[ **What is multithreading in java?** 30](#_Toc195360715)

[ **What are all methods present in Thread class?** 30](#_Toc195360716)

[ **How do notify methods work in multithreading?** 30](#_Toc195360717)

[ **What is synchronized in multithreading?** 30](#_Toc195360718)

[ **What is Singleton in java?** 30](#_Toc195360719)

[ **Explain Java 8 Functional interface?** 30](#_Toc195360720)

[ **What is the lambda expression?** 30](#_Toc195360721)

[ **What is the prerequisite for lambda expression?** 30](#_Toc195360722)

[ **Can we have more than one abstract method in a functional interface?** 30](#_Toc195360723)

[ **What are all functional interfaces present in java?** 30](#_Toc195360724)

[ **What is JVM?** 30](#_Toc195360725)

[ **What are the different loaders in JVM?** 31](#_Toc195360726)

[ **What is garbage collection in java?** 31](#_Toc195360727)

[ **What will happen if I assign null to some object then will that be garbage collected?** 31](#_Toc195360728)

[ **When does the finalize method get called?** 31](#_Toc195360729)

[ **How to handle IOException?** 31](#_Toc195360730)

[ **What is JDBC?** 31](#_Toc195360731)

[ **Explain Hashset?** 31](#_Toc195360732)

[ **What is the difference between list and set?** 31](#_Toc195360733)

[ **Consider we have Strings as String s1 = “Welcome”, String s2 = “Welcome” and String s3 = “WelcomeOne”. What does s1==s2 represent?** 31](#_Toc195360734)

[ **Explain String constant pool?** 31](#_Toc195360735)

[ **What is the difference between equals() and hashCode() method?** 31](#_Toc195360736)

[ **What is the default size of ArrayList?** 31](#_Toc195360737)

[ **How to increase the size of ArrayList?** 31](#_Toc195360738)

[ **Can we add infinite elements to ArrayList?** 31](#_Toc195360739)

[ **Can we store infinite elements in a list?** 31](#_Toc195360740)

[ **Does Java support Pointers?** 31](#_Toc195360741)

[ **Why doesn't java support pointers?** 32](#_Toc195360742)

[ **Explain which language you prefer in terms of security among C, C++, Java and Python?** 32](#_Toc195360743)

[ **Is Java 100% object oriented?** 32](#_Toc195360744)

[ **How can we make java 100% object oriented?** 32](#_Toc195360745)

[ **Can we create objects of static class?** 32](#_Toc195360746)

[ **What is the difference between ArrayList and LinkedList? Which is better?** 32](#_Toc195360747)

[ **What is Enumeration in java?** 32](#_Toc195360748)

[ **Explain Thread life cycle?** 32](#_Toc195360749)

[ **What is Garbage Collection? Explain?** 32](#_Toc195360750)

[ **Explain JVM, JDK and JRE?** 32](#_Toc195360751)

[ **Which compiler is used by Java?** 32](#_Toc195360752)

[ **What is Functional Interface in java?** 32](#_Toc195360753)

[ **Where are the hashCode() and equals() methods defined in java?** 32](#_Toc195360754)

[ **What is a deadlock? Explain with an example?** 32](#_Toc195360755)

[ **How many design patterns are present in java?** 32](#_Toc195360756)

[ **Can we override the static method?** 32](#_Toc195360757)

[ **Explain access modifiers in java?** 33](#_Toc195360758)

[ **What is volatile in java? Where can we use it?** 33](#_Toc195360759)

[ **What is an idempotent method in java?** 33](#_Toc195360760)

[ **How to create threads in java?** 33](#_Toc195360761)

[ **How to start a thread in java?** 33](#_Toc195360762)

[ **Explain heap memory in java?** 33](#_Toc195360763)

[ **When does the garbage collection get called?** 33](#_Toc195360764)

[ **What is final, static and non-static (instance) in java?** 33](#_Toc195360765)

[ **How to create custom exceptions?** 33](#_Toc195360766)

[ **What will happen if I use Exception and custom exceptions together? Priority?** 33](#_Toc195360767)

[ **How will I use two threads using a singleton design pattern?** 33](#_Toc195360768)

[ **How to implement thread safety in java using singleton?** 33](#_Toc195360769)

[ **Write an immutable class in Java?** 33](#_Toc195360770)

[ **Can we override the protected method?** 33](#_Toc195360771)

[ **How to create threads in two different ways?** 33](#_Toc195360772)

[ **Thread safety with Singleton class?** 33](#_Toc195360773)

[ **What is synchronized in java?** 33](#_Toc195360774)

[ **Compare ArrayList objects from the employee and return the highest employee age?** 34](#_Toc195360775)

[ **How does ArrayList internally work?** 34](#_Toc195360776)

[ **What is the default size of ArrayList?** 34](#_Toc195360777)

[ **How to increase the size of ArrayList? By how much?** 34](#_Toc195360778)

[ **Which data structure is used by ArrayList?** 34](#_Toc195360779)

[ **What is predicate and consumer in java 8?** 34](#_Toc195360780)

[ **What is marker interface in java? What is the purpose of the marker interface?** 34](#_Toc195360781)

[ **Explain Thread executors?** 34](#_Toc195360782)

[ **What is the use of finally block in exception handling?** 34](#_Toc195360783)

[ **How can we ignore the finally block?** 34](#_Toc195360784)

[ **What is a try with resources?** 34](#_Toc195360785)

[ **Best time complexity of Arraylist and Linkedlist? Which is better?** 34](#_Toc195360786)

[ **What is the use of a concurrent hashmap?** 34](#_Toc195360787)

[ **What is synchronous and asynchronous call?** 34](#_Toc195360788)

[ **Difference between HashMap and LinkedHashMap?** 34](#_Toc195360789)

[ **What is the difference between Fail Fast and Fail Safe iterator?** 34](#_Toc195360790)

[ **Can we write the main method as private?** 34](#_Toc195360791)

[ **What is the use of static keywords in java?** 35](#_Toc195360792)

[ **What is inner class? Have you used inner class in your project?** 35](#_Toc195360793)

[ **What is the purpose of inner class?** 35](#_Toc195360794)

[ **Scenario based question - 1?** 35](#_Toc195360795)

[ **What is Enum? Write a syntax for enum?** 35](#_Toc195360796)

[ **Write a syntax for integer ArrayList?** 35](#_Toc195360797)

[ **Difference between wait and sleep?** 35](#_Toc195360798)

[ **Difference between static and non-static in java?** 35](#_Toc195360799)

[ **Difference between Fail Fast and Fail Safe iterator?** 35](#_Toc195360800)

[ **Scenario based question - 2?** 35](#_Toc195360801)

[ **Scenario based question - 3?** 36](#_Toc195360802)

[ **Scenario based question - 4?** 36](#_Toc195360803)

[ **Scenario based question - 5?** 36](#_Toc195360804)

[ **How can we break the singleton pattern?** 36](#_Toc195360805)

[ **Explain memory management below? Int and Integer?** 36](#_Toc195360806)

[ **What is the difference between default and protected access modifier?** 36](#_Toc195360807)

[ **How to create three threads so that it will execute simultaneously?** 36](#_Toc195360808)

[ **What is the difference between HashMap and HashTable?** 36](#_Toc195360809)

[ **What is Runnable and Callable interface?** 37](#_Toc195360810)

[ **What can we replace instead of this keyword in the statement? synchronized(this) {?** 37](#_Toc195360811)

[ **How can we add data while iterating ArrayList?** 37](#_Toc195360812)

[ **What is the difference between List and Map?** 37](#_Toc195360813)

[ **Which data structure used by HashSet?** 37](#_Toc195360814)

[ **Explain sorted collections in java?** 37](#_Toc195360815)

[ **Which data structure is used by HashMap?** 37](#_Toc195360816)

[ **How can Set can’t store duplicate data?** 37](#_Toc195360817)

[ **What is the difference between stack and heap?** 37](#_Toc195360818)

[ **Why is string immutable?** 37](#_Toc195360819)

[ **Give an example of Functional Interface in java?** 37](#_Toc195360820)

[ **Difference between Map and Set?** 37](#_Toc195360821)

[ **How HashMap calls hashCode and equals method internally?** 37](#_Toc195360822)

[ **Explain internal working of ArrayList?** 37](#_Toc195360823)

[ **How does ArrayList increase its size dynamically?** 37](#_Toc195360824)

[ **How does ArrayList grow?** 37](#_Toc195360825)

[ **Explain internal working of ensureCapacity method?** 38](#_Toc195360826)

[ **How do you write a custom ArrayList which will take the input capacity?** 38](#_Toc195360827)

[ **How will you find the 3rd last element from the linked list in a single iteration?** 38](#_Toc195360828)

[ **How does ConcurrentHashMap works?** 38](#_Toc195360829)

[ **How to create a Thread pool in java?** 38](#_Toc195360830)

[ **What is an immutable class?** 38](#_Toc195360831)

[ **What is the Consumer interface in java?** 38](#_Toc195360832)

[ **What is Reentrant lock in java?** 38](#_Toc195360833)

[ **What is Atomic integer in java?** 38](#_Toc195360834)

[ **What is Callable interface?** 38](#_Toc195360835)

[ **What is a Blocking Queue?** 38](#_Toc195360836)

[ **How can you make a singleton class as thread safe?** 38](#_Toc195360837)

[ **What is the difference between comparTo and compare method?** 38](#_Toc195360838)

[ **What is the return type of compare method?** 38](#_Toc195360839)

[ **What is contract in equals and hashcode method?** 38](#_Toc195360840)

[ **What is the difference between Marker interface and Functional interface?** 39](#_Toc195360841)

[ **What is the difference between Abstract class and Interface? Explain with scenarios which one you will it be used with conditions?** 39](#_Toc195360842)

[ **If you have try, catch and finally block and if I have written another exception inside catch block so how control will flow?** 39](#_Toc195360843)

[ **What is the use of static and instance variables in java?** 39](#_Toc195360844)

[ **How to create threads using java 1.7 and java 1.8?** 39](#_Toc195360845)

[ **Which method you will override in order to create a thread?** 39](#_Toc195360846)

[ **What is the lambda expression in java 8?** 39](#_Toc195360847)

[ **How to use lambda expressions with one example?** 39](#_Toc195360848)

[ **What is Stream and Parallel stream?** 39](#_Toc195360849)

[ **Explain Optional class in java 8?** 39](#_Toc195360850)

[ **What is the purpose of Optional class?** 39](#_Toc195360851)

[ **How will you access JNDI in java code?** 39](#_Toc195360852)

[ **Why is the main method in java static?** 39](#_Toc195360853)

[ **What is the use of the static keyword?** 39](#_Toc195360854)

[ **How many ways can we create beans in java?** 39](#_Toc195360855)

[ **Explain different types of calling beans?** 39](#_Toc195360856)

[ **Difference between Map and Filter in java 8?** 40](#_Toc195360857)

[ **What are the default methods?** 40](#_Toc195360858)

[ **Can I add one more new default method in my child class? Does it give any problems with my existing classes?** 40](#_Toc195360859)

[ **What is a predicate?** 40](#_Toc195360860)

[ **What is the difference between Comparable and Comparator interface?** 40](#_Toc195360861)

[ **Which Collection will you avoid duplicating and store data in a sorted manner?** 40](#_Toc195360862)

[ **In which scenarios Linked list comes into fixture in HashMap?** 40](#_Toc195360863)

[ **In Java, which class is the base class of all the classes?** 40](#_Toc195360864)

[ **Why is the equals method required?** 40](#_Toc195360865)

[ **Is it possible to have the same hashcode for multiple objects?** 40](#_Toc195360866)

[ **Enlist a few predefined interfaces from Java 8?** 40](#_Toc195360867)

[ **What is Functional interface? What is the use of it?** 40](#_Toc195360868)

[ **Suppose you are using lambda expressions and other than lambda expressions then are we getting any performance benefits?** 40](#_Toc195360869)

[ **Can you elaborate on the Stream API, purpose and use of it?** 40](#_Toc195360870)

[ **Have you created a custom exception? What is the purpose of that?** 40](#_Toc195360871)

[ **Difference between Array and Collections?** 40](#_Toc195360872)

[ **Have you ever got outOfMemory exceptions? Can you tell me any scenarios where you have faced this exception?** 41](#_Toc195360873)

[ **If I have a 200 MB excel file so can java read that file?** 41](#_Toc195360874)

[ **What is immutability and mutability in java?** 41](#_Toc195360875)

[ **How can Set not store duplicate values? Explain internal working?** 41](#_Toc195360876)

[ **What is Stream API?** 41](#_Toc195360877)

[ **Have you worked on Collections? How is the difference between Stream and collections?** 41](#_Toc195360878)

[ **How hashcode calculations happened on the Employee object?** 41](#_Toc195360879)

[ **Suppose I have multiple Employee objects with different names, so how does hashmap differentiate between different objects?** 41](#_Toc195360880)

[ **Can we have duplicate keys in hashmap?** 41](#_Toc195360881)

[ **How does hashmap come to know that two objects are the same or different?** 41](#_Toc195360882)

[ **Have you worked on NullPointerException? How to handle NullPointerException in java 8?** 41](#_Toc195360883)

[ **Custom implementation of Singleton beans?** 41](#_Toc195360884)

[ **How to sort the list of objects in java using Stream API?** 41](#_Toc195360885)

[ **What are the practical use cases of LinkedList?** 41](#_Toc195360886)

[ **What is the default scope of beans?** 41](#_Toc195360887)

[ **What is the difference between Process and Thread?** 42](#_Toc195360888)

[ **Explain the Thread Life Cycle?** 42](#_Toc195360889)

[ **What is a Synchronized Thread?** 42](#_Toc195360890)

[ **Have you come across the OutOfMemory exception? How will you handle it? Explain with standalone code?** 42](#_Toc195360891)

[ **What are the ways of blocking the threads in java?** 42](#_Toc195360892)

[ **What is thread synchronization in java?** 42](#_Toc195360893)

[ **What is the necessity of thread synchronization?** 42](#_Toc195360894)

[ **Explain Composition in Oops?** 42](#_Toc195360895)

[ **Difference between static and final?** 42](#_Toc195360896)

[ **Difference between interface and abstract class?** 42](#_Toc195360897)

[ **Where will you use interface and abstract class with real time examples?** 42](#_Toc195360898)

[ **Explain about String class in java?** 42](#_Toc195360899)

[ **What is a special feature of String?** 42](#_Toc195360900)

[ **Create an immutable class in java?** 42](#_Toc195360901)

[ **How to access Singleton class from another class?** 42](#_Toc195360902)

[ **Where will the Singleton class be stored in memory?** 42](#_Toc195360903)

[ **Explain Java memory management?** 43](#_Toc195360904)

[ **What is Stack / Heap memory?** 43](#_Toc195360905)

[ **Where local variables are stored in memory?** 43](#_Toc195360906)

[ **Where objects will be stored in memory?** 43](#_Toc195360907)

[ **Difference between HashMap and ConcurrentHashMap?** 43](#_Toc195360908)

[ **Where will you use ConcurrentHashMap?** 43](#_Toc195360909)

[ **Explain HashMap vs ConcurrentHashMap?** 43](#_Toc195360910)

[ **How is HashMap faster as compared to ConcurrentHashMap?** 43](#_Toc195360911)

[ **Difference between bucket level and segment level locking?** 43](#_Toc195360912)

[ **How is the static method bound with an object?** 43](#_Toc195360913)

[ **What is the difference between Array, ArrayList and LinkedList? When you will use ArrayList and LinkedList?** 43](#_Toc195360914)

[ **What is the use of the intern() method in String class?** 43](#_Toc195360915)

[ **What is encapsulation? How are you currently using it?** 43](#_Toc195360916)

[ **What is the use of Map?** 43](#_Toc195360917)

[ **What is the intermediate and terminal operator in Java stream API?** 43](#_Toc195360918)

[ **Advantages of Stream over for loop?** 43](#_Toc195360919)

[ **What is Optional class? What are the advantages of using Optional class?** 44](#_Toc195360920)

[ **Explain Compile time and Run time polymorphism?** 44](#_Toc195360921)

[ **Optional case returning null and String, how to handle it? What are the advantages of using it?** 44](#_Toc195360922)

[ **What is the use of @FunctionalInterface annotation? Will it be checked manually or by the compiler?** 44](#_Toc195360923)

[ **What is the added version after Java 17 to Java 21?** 44](#_Toc195360924)

[ **What is the inbuilt functional interface in Java 8?** 44](#_Toc195360925)

[ **Difference between map and flatmap?** 44](#_Toc195360926)

[ **Difference between Predicate and Function?** 44](#_Toc195360927)

[ **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](#_Toc195360928)

[ **What is the default capacity of HashMap?** 44](#_Toc195360929)

[***Java Programs*** 45](#_Toc195360930)

[ **Program – 1** 45](#_Toc195360931)

[ **Program – 2** 45](#_Toc195360932)

[ **Program – 3** 45](#_Toc195360933)

[ **Program – 4** 45](#_Toc195360934)

[ **Program – 5** 45](#_Toc195360935)

[ **Program – 6** 45](#_Toc195360936)

[ **Program – 7** 46](#_Toc195360937)

[ **Program – 8** 46](#_Toc195360938)

[ **Program – 9** 47](#_Toc195360939)

[ **Program – 10** 47](#_Toc195360940)

[ **Program – 11** 47](#_Toc195360941)

[ **Program – 12** 47](#_Toc195360942)

[ **Program – 13** 47](#_Toc195360943)

[ **Program – 14** 47](#_Toc195360944)

[ **Program – 15** 47](#_Toc195360945)

[ **Program – 16** 47](#_Toc195360946)

[ **Program – 17** 47](#_Toc195360947)

[ **Program – 18** 47](#_Toc195360948)

[ **Program – 19** 48](#_Toc195360949)

[ **Program – 20** 48](#_Toc195360950)

[ **Program – 21** 48](#_Toc195360951)

[ **Program – 22** 48](#_Toc195360952)

[ **Program – 23** 48](#_Toc195360953)

[ **Program – 24** 48](#_Toc195360954)

[ **Program – 25** 48](#_Toc195360955)

[ **Program – 26** 48](#_Toc195360956)

[ **Program – 27** 48](#_Toc195360957)

[ **Program – 28** 48](#_Toc195360958)

[ **Program – 29** 49](#_Toc195360959)

[ **Program – 30** 49](#_Toc195360960)

[ **Program – 31** 49](#_Toc195360961)

[ **Program – 32** 50](#_Toc195360962)

[ **Program – 33** 50](#_Toc195360963)

[ **Program – 34** 50](#_Toc195360964)

[ **Program – 35** 50](#_Toc195360965)

[ **Program – 36** 50](#_Toc195360966)

[ **Program – 37** 50](#_Toc195360967)

[ **Program – 38** 50](#_Toc195360968)

[ **Program – 39** 50](#_Toc195360969)

[ **Program – 40** 50](#_Toc195360970)

[ **Program – 41** 51](#_Toc195360971)

[ **Program – 42** 51](#_Toc195360972)

[ **Program – 43** 51](#_Toc195360973)

[ **Program – 44** 51](#_Toc195360974)

[ **Program – 45** 51](#_Toc195360975)

[ **Program – 46** 51](#_Toc195360976)

[ **Program – 47** 51](#_Toc195360977)

[ **Program – 48** 51](#_Toc195360978)

[ **Program – 49** 51](#_Toc195360979)

[ **Program – 50** 51](#_Toc195360980)

[ **Program – 51** 52](#_Toc195360981)

[Advance Java / J2EE 52](#_Toc195360982)

[ **What is servlet?** 52](#_Toc195360983)

[ **Explain servlet lifecycle?** 52](#_Toc195360984)

[ **What is a dispatcher servlet?** 52](#_Toc195360985)

[ **What is JSP?** 52](#_Toc195360986)

[ **What is the difference between servlet and JSP?** 52](#_Toc195360987)

[ **What is the JSP lifecycle?** 52](#_Toc195360988)

[ **What is Spring MVC? Explain architecture of spring MVC?** 52](#_Toc195360989)

[ **What is the singleton design pattern?** 52](#_Toc195360990)

[ **What is the REST API?** 52](#_Toc195360991)

[ **Difference between API and REST?** 52](#_Toc195360992)

[ **Difference between SOAP and REST?** 52](#_Toc195360993)

[ **Which among SOAP and REST is easy?** 52](#_Toc195360994)

[ **Explain REST architecture?** 52](#_Toc195360995)

[ **What do you know about DTD and XSD?** 53](#_Toc195360996)

[ **Explain SOAP Web Service?** 53](#_Toc195360997)

[ **What is Path in XML?** 53](#_Toc195360998)

[ **Write a path for a given xml file?** 53](#_Toc195360999)

[ **What is the use of XSLT?** 53](#_Toc195361000)

[ **How does SOAP work?** 53](#_Toc195361001)

[ **question?** 53](#_Toc195361002)

[ **question?** 53](#_Toc195361003)

[ **question?** 53](#_Toc195361004)

[ **question?** 53](#_Toc195361005)

[ **question?** 53](#_Toc195361006)

[ **question?** 53](#_Toc195361007)

[ **question?** 53](#_Toc195361008)

[ **question?** 53](#_Toc195361009)

[ **question?** 53](#_Toc195361010)

[ **question?** 53](#_Toc195361011)

[ **question?** 53](#_Toc195361012)

[Data Structure 55](#_Toc195361013)

[ **Find the middle element from the singly linked list without finding the length?** 55](#_Toc195361014)

[ **Complexity to find length of singly linked list?** 55](#_Toc195361015)

[ **question?** 55](#_Toc195361016)

[ **question?** 55](#_Toc195361017)

[ **question?** 55](#_Toc195361018)

[ **question?** 55](#_Toc195361019)

[ **question?** 55](#_Toc195361020)

[ **question?** 55](#_Toc195361021)

[ **question?** 55](#_Toc195361022)

[ **question?** 55](#_Toc195361023)

[ **question?** 55](#_Toc195361024)

[Design Patterns 56](#_Toc195361025)

[ **What is the singleton design pattern? Explain it?** 56](#_Toc195361026)

[ **Explain Factory design pattern?** 56](#_Toc195361027)

[ **What is the Observer design pattern?** 56](#_Toc195361028)

[Spring 57](#_Toc195361029)

[ **Explain Spring MVC architecture?** 57](#_Toc195361030)

[ **Explain Spring IOC?** 57](#_Toc195361031)

[ **Explain Spring Dependency Injection?** 57](#_Toc195361032)

[ **What are the spring bean scopes?** 57](#_Toc195361033)

[ **Explain any scope of spring bean?** 57](#_Toc195361034)

[ **How to add scope in bean configuration?** 57](#_Toc195361035)

[ **Difference between session and prototype scope?** 57](#_Toc195361036)

[ **What is the Spring batch? Explain where you have used the spring batch in your project?** 57](#_Toc195361037)

[ **How spring batch works?** 57](#_Toc195361038)

[ **How to implement the spring batch?** 57](#_Toc195361039)

[ **What is the difference between SOAP and REST? Which is more efficient?** 57](#_Toc195361040)

[ **How REST API works?** 57](#_Toc195361041)

[ **Difference between Spring and Spring Boot?** 57](#_Toc195361042)

[ **Difference between application context and beanfactory in spring?** 57](#_Toc195361043)

[ **Explain Spring MVC architecture?** 57](#_Toc195361044)

[ **Difference between Spring and Spring Boot?** 58](#_Toc195361045)

[ **What is the use of @Transactional annotation in spring?** 58](#_Toc195361046)

[ **What are different stereotypes annotations in spring?** 58](#_Toc195361047)

[ **question?** 58](#_Toc195361048)

[ **question?** 58](#_Toc195361049)

[ **question?** 58](#_Toc195361050)

[ **question?** 58](#_Toc195361051)

[ **question?** 58](#_Toc195361052)

[ **question?** 58](#_Toc195361053)

[ **question?** 58](#_Toc195361054)

[ **question?** 58](#_Toc195361055)

[ **question?** 58](#_Toc195361056)

[ **question?** 58](#_Toc195361057)

[ **question?** 58](#_Toc195361058)

[Hibernate 60](#_Toc195361059)

[ How data is persisted in database by using hibernate**?** 60](#_Toc195361060)

[ **How to retrieve select query using Hibernate?** 62](#_Toc195361061)

[ **How to retrieve data through hibernate?** 62](#_Toc195361062)

[ **Hibernate caching?** 62](#_Toc195361063)

[ **Explain L1 and L2 cache in Hibernate?** 62](#_Toc195361064)

[ **Explain the hibernate sessions?** 63](#_Toc195361065)

[ **Difference between JDBC and Hibernate?** 63](#_Toc195361066)

[ **What is the difference between get and load method in hibernate?** 63](#_Toc195361067)

[ **Explain bean scopes?** 63](#_Toc195361068)

[ **Can we have beans without bean id?** 63](#_Toc195361069)

[ **Difference between get and load method?** 63](#_Toc195361070)

[ **Explain Performance tuning and indexing in hibernate?** 63](#_Toc195361071)

[ **Explain different classes and annotations in hibernate?** 63](#_Toc195361072)

[ **How to call stored procedures using hibernate?** 63](#_Toc195361073)

[ **What is Many to many association in hibernate?** 63](#_Toc195361074)

[ **question?** 63](#_Toc195361075)

[ **question?** 63](#_Toc195361076)

[ **question?** 63](#_Toc195361077)

[ **question?** 63](#_Toc195361078)

[ **question?** 63](#_Toc195361079)

[ **question?** 63](#_Toc195361080)

[ **question?** 63](#_Toc195361081)

[Spring Boot 65](#_Toc195361082)

[ **Difference between Spring and Spring Boot?** 65](#_Toc195361083)

[ **What are the advantages of SprigBoot?** 65](#_Toc195361084)

[ **What is a yaml file in SpringBoot?** 65](#_Toc195361085)

[ **What is what is @transient in spring boot?** 65](#_Toc195361086)

[ **What are the types of propagation?** 65](#_Toc195361087)

[ **What @Qualifier in spring? What is the use of @Qualifier?** 65](#_Toc195361088)

[ **What are the advantages of Spring boot?** 65](#_Toc195361089)

[ **How to create multiple modules using spring boot?** 65](#_Toc195361090)

[ **What are all methods present in the controller?** 65](#_Toc195361091)

[ **Explain annotations in Spring boot?** 65](#_Toc195361092)

[ **What is the query parameter?** 65](#_Toc195361093)

[ **How to change the server in Spring boot?** 65](#_Toc195361094)

[ **Difference between @Controller and @RestController?** 65](#_Toc195361095)

[ **What is the use of @Responsebody annotation?** 65](#_Toc195361096)

[ **Explain @RestController annotation in Spring boot?** 65](#_Toc195361097)

[ **Which two annotations are used in built by @RestController?** 65](#_Toc195361098)

[ **How to access application.properties properties in java code?** 66](#_Toc195361099)

[ **Explain Spring boot application? How does it work?** 66](#_Toc195361100)

[ **What is the use of @Service annotation in spring?** 66](#_Toc195361101)

[ **What is the use of @Component annotation in spring?** 66](#_Toc195361102)

[ **What is the use of @Repository annotation in spring?** 66](#_Toc195361103)

[ **What is the difference between @Service and @Controller annotation?** 66](#_Toc195361104)

[ **What is @ComponentScan?** 66](#_Toc195361105)

[ **What is the use of @Autowired annotation?** 66](#_Toc195361106)

[ **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?** 66](#_Toc195361107)

[ **How to handle Exceptions in Spring boot?** 66](#_Toc195361108)

[ **How to provide security to spring boot applications?** 66](#_Toc195361109)

[ **What is @ControllerAdvice in Spring boot?** 66](#_Toc195361110)

[ **How to retrieve query parameters in Spring boot?** 66](#_Toc195361111)

[ **How is the data persisted in DB with the REST API?** 66](#_Toc195361112)

[ **Explain GET and POST methods?** 66](#_Toc195361113)

[ **Write a SQL or JPQL to retrieve the data from DB?** 66](#_Toc195361114)

[ **How will you add Employee object data in the DB using POST call?** 67](#_Toc195361115)

[ **What is the size of Spring in your project? Who will decide that?** 67](#_Toc195361116)

[ **Explain about spring security?** 67](#_Toc195361117)

[ **Explain the flow of REST api?** 67](#_Toc195361118)

[ **What will happen when we replace @Service instead of @Repository?** 67](#_Toc195361119)

[ **Explain uses of SpringBoot application?** 67](#_Toc195361120)

[ **How is https used in SpringBoot applications?** 67](#_Toc195361121)

[ **Difference between RequestMapping and GetMapping?** 67](#_Toc195361122)

[ **question?** 67](#_Toc195361123)

[ **question?** 67](#_Toc195361124)

[ **question?** 67](#_Toc195361125)

[ **question?** 67](#_Toc195361126)

[ **question?** 67](#_Toc195361127)

[ **question?** 67](#_Toc195361128)

[ **question?** 67](#_Toc195361129)

[ **question?** 67](#_Toc195361130)

[ **question?** 67](#_Toc195361131)

[Microservices 68](#_Toc195361132)

[ **Monolith vs Microservices differences and when to choose which?** 68](#_Toc195361133)

[ **How to design a microservice from scratch?** 68](#_Toc195361134)

[ **API Gateway pattern and its advantages?** 68](#_Toc195361135)

[ **Inter-service communication: REST vs Messaging?** 68](#_Toc195361136)

[ **Circuit Breaker pattern and its implementation using Resilience4j?** 68](#_Toc195361137)

[ **Load balancing in microservices using Spring Cloud Load Balancer?** 68](#_Toc195361138)

[ **How Spring Cloud Config helps in centralized configuration management?** 68](#_Toc195361139)

[ **Service discovery using Eureka or Consul?** 68](#_Toc195361140)

[ **Feign Client vs WebClient: Which one to use and why?** 68](#_Toc195361141)

[ **Event-driven architecture and Kafka integration?** 68](#_Toc195361142)

[ **Database per service vs Shared Database: Pros and cons?** 68](#_Toc195361143)

[ **Saga Pattern for distributed transactions in microservices?** 68](#_Toc195361144)

[ **JWT-based authentication and OAuth2 in microservices?** 68](#_Toc195361145)

[ **How to handle security in an API Gateway?** 68](#_Toc195361146)

[ **Observability: Logging, tracing, and monitoring best practices?** 68](#_Toc195361147)

[ **Role of Prometheus and Grafana in microservices monitoring?** 69](#_Toc195361148)

[ **Kubernetes deployment strategies for microservices?** 69](#_Toc195361149)

[ **Blue-Green and Canary deployments in microservices?** 69](#_Toc195361150)

[ **When to use WebFlux for reactive microservices?** 69](#_Toc195361151)

[ **CQRS and Event Sourcing: When and why to use them?** 69](#_Toc195361152)

[SQL 70](#_Toc195361153)

[ **What is ordered by and range in the database?** 70](#_Toc195361154)

[ **Write a SQL query to get the second highest salary of an employee from the employee table?** 70](#_Toc195361155)

[ **Write a SQL query to print employee id, employee\_name, depatment\_name Employee - employee\_id, employee\_name, department\_id** 70](#_Toc195361156)

[**Department - department\_id, department\_name?** 70](#_Toc195361157)

[ **What is a collision in the oracle database?** 70](#_Toc195361158)

[ **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?** 70](#_Toc195361159)

[ **What are DDL and DML commands?** 70](#_Toc195361160)

[ **Write a query to update data in the table?** 70](#_Toc195361161)

[ **What is the use of joins in the database?** 70](#_Toc195361162)

[ **Explain ACID properties in the database?** 70](#_Toc195361163)

[ **Explain Normalization?** 70](#_Toc195361164)

[ **What are the types of Normalization?** 70](#_Toc195361165)

[ **What is Denormalization?** 70](#_Toc195361166)

[ **What is the use of join?** 70](#_Toc195361167)

[ **Write a SQL query to return the 7th highest salary of an employee?** 71](#_Toc195361168)

[ **What is Union in SQL?** 71](#_Toc195361169)

[ **Can we perform union operations on one table?** 71](#_Toc195361170)

[ **Difference between where and let?** 71](#_Toc195361171)

[ **What is Semaphore?** 71](#_Toc195361172)

[ **What is View and Index in SQL?** 71](#_Toc195361173)

[ **What is Trigger in SQL? What is the use of Trigger in SQL?** 71](#_Toc195361174)

[ **What is inner join in SQL?** 71](#_Toc195361175)

[ **Write a SQL query to get emp\_sal > 5000 and company\_name = "Kotak"** 71](#_Toc195361176)

[ **Write a JPA query to retrieve the above result?** 71](#_Toc195361177)

[ **What is an index in SQL?** 71](#_Toc195361178)

[ **Explain different types of Index?** 71](#_Toc195361179)

[ **Explain different types of joins?** 72](#_Toc195361180)

[ **Difference between Outer join and Full outer join?** 72](#_Toc195361181)

[ **Write a SQL query to find the name of a city which has more than one customer in it?** 72](#_Toc195361182)

[ **Write SQL query to display 10th highest salary of employee?** 72](#_Toc195361183)

[ **What is the difference between DDL and DML commands?** 72](#_Toc195361184)

[ **Difference between Inner join and Outer join?** 72](#_Toc195361185)

[ **Difference between left outer join and right outer join?** 72](#_Toc195361186)

[ **What are Triggers?** 72](#_Toc195361187)

[ **SQL query to get the youngest employee from the employee table?** 72](#_Toc195361188)

[ **SQL query to get list of employees from Finance department?** 72](#_Toc195361189)

[ **question?** 72](#_Toc195361190)

[ **question?** 72](#_Toc195361191)

[ **question?** 72](#_Toc195361192)

[ **question?** 72](#_Toc195361193)

[ **question?** 72](#_Toc195361194)

[Kafka 73](#_Toc195361195)

[ **question?** 73](#_Toc195361196)

[ **question?** 73](#_Toc195361197)

[ **question?** 73](#_Toc195361198)

[ **question?** 73](#_Toc195361199)

[ **question?** 73](#_Toc195361200)

[GitHub 74](#_Toc195361201)

[ **What is Version control i.e. Git?** 74](#_Toc195361202)

[ **How to push and commit code in git using commands?** 74](#_Toc195361203)

[ **Difference between rebase and merge?** 74](#_Toc195361204)

[ **question?** 74](#_Toc195361205)

[ **question?** 74](#_Toc195361206)

[ **question?** 74](#_Toc195361207)

[ **question?** 74](#_Toc195361208)

[Docker 75](#_Toc195361209)

[ **question?** 75](#_Toc195361210)

[ **question?** 75](#_Toc195361211)

[ **question?** 75](#_Toc195361212)

[ **question?** 75](#_Toc195361213)

[ **question?** 75](#_Toc195361214)

[ **question?** 75](#_Toc195361215)

[Kubernetes 76](#_Toc195361216)

[ **question?** 76](#_Toc195361217)

[ **question?** 76](#_Toc195361218)

[ **question?** 76](#_Toc195361219)

[ **question?** 76](#_Toc195361220)

[ **question?** 76](#_Toc195361221)

[ **question?** 76](#_Toc195361222)

[AWS 77](#_Toc195361223)

[ **What is cloud computing?** 77](#_Toc195361224)

[ **Explain services provided by cloud computing?** 77](#_Toc195361225)

[ **question?** 77](#_Toc195361226)

[ **question?** 77](#_Toc195361227)

[ **question?** 77](#_Toc195361228)

[ **question?** 77](#_Toc195361229)

[Azure 78](#_Toc195361230)

[ **question?** 78](#_Toc195361231)

[ **question?** 78](#_Toc195361232)

[ **question?** 78](#_Toc195361233)

[ **question?** 78](#_Toc195361234)

[ **question?** 78](#_Toc195361235)

[ **question?** 78](#_Toc195361236)

[Angular 79](#_Toc195361237)

[ **What are the components in the Angular?** 79](#_Toc195361238)

[ **How to pass data from components in Angular?** 79](#_Toc195361239)

[ **What does the component.ts file contain in angular?** 79](#_Toc195361240)

[ **Explain two way data binding in angular?** 79](#_Toc195361241)

[ **How to make backend calls in angular?** 79](#_Toc195361242)

[ **What is authguard in angular?** 79](#_Toc195361243)

[ **What is interpolation in angular?** 79](#_Toc195361244)

[ **How to create components in angular?** 79](#_Toc195361245)

[ **Explain angular directives?** 79](#_Toc195361246)

[ **What are observables in angular?** 79](#_Toc195361247)

[ **How to implement dependency injection in angular?** 79](#_Toc195361248)

[ **question?** 79](#_Toc195361249)

[React 80](#_Toc195361250)

[ **question?** 80](#_Toc195361251)

[ **question?** 80](#_Toc195361252)

[ **question?** 80](#_Toc195361253)

[ **question?** 80](#_Toc195361254)

[ **question?** 80](#_Toc195361255)

[ **question?** 80](#_Toc195361256)

[Others 81](#_Toc195361257)

[ **Explain Jira?** 81](#_Toc195361258)

[ **What is the output of below program?** 81](#_Toc195361259)

[ **What is https protocol?** 81](#_Toc195361260)

[ **How does https internally work?** 81](#_Toc195361261)

[ **What is a certificate?** 81](#_Toc195361262)

[ **Difference between http and https?** 81](#_Toc195361263)

[ **What is a Sprint retrospective?** 81](#_Toc195361264)

[ **Do you know anything about CI/CD?** 81](#_Toc195361265)

[ **How will Jenkins' job work? Explain the process?** 81](#_Toc195361266)

[ **question?** 81](#_Toc195361267)

[ **question?** 81](#_Toc195361268)

[ **question?** 81](#_Toc195361269)

[ **question?** 81](#_Toc195361270)

[ **question?** 81](#_Toc195361271)

[ **question?** 82](#_Toc195361272)

[ **question?** 82](#_Toc195361273)

[ **question?** 82](#_Toc195361274)

[ **question?** 82](#_Toc195361275)

[ **question?** 82](#_Toc195361276)

[ **question?** 82](#_Toc195361277)

[Linux 83](#_Toc195361278)

[ **Explain Linux commands?** 83](#_Toc195361279)

[ **Command - Read a file and print first two lines from file Ans. head -2 filename?** 83](#_Toc195361280)

[ **Search a word from a file?** 83](#_Toc195361281)

[ **Why do you want to change your current organization?** 83](#_Toc195361282)

[Puzzles 84](#_Toc195361283)

[ **Puzzle – 1** 84](#_Toc195361284)

[ **Puzzle – 2** 84](#_Toc195361285)

[ **Puzzle – 3** 84](#_Toc195361286)

# ***Java***

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

* Answer

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

* Answer

### **Explain JDK, JRE?**

* answer

### **Explain JVM?**

* 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

### **How to create custom exceptions?**

* 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

### **What is the difference between comparable and comparator?**

* Answer

### **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

### **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).

### **What is the difference between HashMap and HashSet**

* Answer

### **Explain Java8 features?**

* answer

### **What is Functional Interface?**

* answer

### **What is the default method in functional interface?**

* answer

### **What is the map function in stream?**

* answer

### **What is the filter in the stream?**

* 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

### **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

### **Difference between ArrayList and LinkedList?**

* Answer

### **Which is more efficient among the ArrayList and LinkedList?**

* answer

### **Which among ArrayList and LinkedList will be more efficient for random access of data?**

* answer

### **How to sort the ArrayList?**

* Answer

### **How to sort the ArrayList?**

* answer

### **Difference between ArrayList and Vector?**

* answer

### **What is HashMap?**

* 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

### **What is exception handling?**

* answer

### **Explain the hierarchy of Exception?**

* answer

### **What is the difference between checked exception and unchecked exception?**

* answer

### **How to handle unchecked exceptions?**

* answer

### **Explain Generics? What is the use of Generics?**

* answer

### **What is composition and aggregation?**

* answer

### **Parent and child class methods and how to access those?**

* answer

### **What is the use map method in java 8 stream?**

* answer

### **Which collection you will use in order to remove duplicate elements and preserve the insertion order?**

* answer

### **How to find the second largest element from the list?**

* answer

### **What is the difference between Map and FlatMap in stream?**

* answer

### **Where can you use FlatMap in a stream?**

* answer

### **What is multithreading in java?**

* answer

### **What are all methods present in Thread class?**

* answer

### **How do notify methods work in multithreading?**

* answer

### **What is synchronized in multithreading?**

* answer

### **What is Singleton in java?**

* answer

### **Explain Java 8 Functional interface?**

* answer

### **What is the lambda expression?**

* answer

### **What is the prerequisite for lambda expression?**

* answer

### **Can we have more than one abstract method in a functional interface?**

* answer

### **What are all functional interfaces present in java?**

* answer

### **What is JVM?**

* answer

### **What are the different loaders in JVM?**

* answer

### **What is garbage collection 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

### **How to handle IOException?**

* answer

### **What is JDBC?**

* answer

### **Explain Hashset?**

* answer

### **What is the difference between list and set?**

* 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

### **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

### **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

### **Explain Thread life cycle?**

* answer

### **What is Garbage Collection? Explain?**

* answer

### **Explain JVM, JDK and JRE?**

* answer

### **Which compiler is used by Java?**

* answer

### **What is Functional Interface in java?**

* answer

### **Where are the hashCode() and equals() methods defined in java?**

* answer

### **What is a deadlock? Explain with an example?**

* 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

### **How to create threads in java?**

* answer

### **How to start a thread in java?**

* answer

### **Explain heap memory in java?**

* answer

### **When does the garbage collection get called?**

* answer

### **What is final, static and non-static (instance) in java?**

* answer

### **How to create custom exceptions?**

* answer

### **What will happen if I use Exception and custom exceptions together? Priority?**

* 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

### **How to create threads in two different ways?**

* answer

### **Thread safety with Singleton class?**

* answer

### **What is synchronized in java?**

* answer

### **Compare ArrayList objects from the employee and return the highest employee age?**

* 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

### **Which data structure is used by ArrayList?**

* answer

### **What is predicate and consumer in java 8?**

* answer

### **What is marker interface in java? What is the purpose of the marker interface?**

* answer

### **Explain Thread executors?**

* answer

### **What is the use of finally block in exception handling?**

* answer

### **How can we ignore the finally block?**

* answer

### **What is a try with resources?**

* answer

### **Best time complexity of Arraylist and Linkedlist? Which is better?**

* answer

### **What is the use of a concurrent hashmap?**

* answer

### **What is synchronous and asynchronous call?**

* answer

### **Difference between HashMap and LinkedHashMap?**

* answer

### **What is the difference between Fail Fast and Fail Safe iterator?**

* 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

### **Scenario based question - 1?**

String s1 = “abc”; and String s2 = new String(“abc”); Where will s1 and s2 be stored?

* answer

### **What is Enum? Write a syntax for enum?**

* answer

### **Write a syntax for integer ArrayList?**

* answer

### **Difference between wait and sleep?**

* answer

### **Difference between static and non-static in java?**

* answer

### **Difference between Fail Fast and Fail Safe iterator?**

* 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

### **How can we break the singleton pattern?**

* answer

### **Explain memory management below? Int and Integer?**

* answer

### **What is the difference between default and protected access modifier?**

* answer

### **How to create three threads so that it will execute simultaneously?**

* answer

### **What is the difference between HashMap and HashTable?**

* answer

### **What is Runnable and Callable interface?**

* answer

### **What can we replace instead of this keyword in the statement? synchronized(this) {?**

* 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

### **What is the difference between stack and heap?**

* answer

### **Why is string immutable?**

* answer

### **Give an example of Functional Interface in java?**

* 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

### **How will you find the 3rd last element from the linked list in a single iteration?**

* answer

### **How does ConcurrentHashMap works?**

* answer

### **How to create a Thread pool in java?**

* answer

### **What is an immutable class?**

* answer

### **What is the Consumer interface in java?**

* answer

### **What is Reentrant lock in java?**

* answer

### **What is Atomic integer in java?**

* answer

### **What is Callable interface?**

* answer

### **What is a Blocking Queue?**

* answer

### **How can you make a singleton class as thread safe?**

* answer

### **What is the difference between comparTo and compare method?**

* answer

### **What is the return type of compare method?**

* answer

### **What is contract in equals and hashcode method?**

* answer

### **What is the difference between Marker interface and Functional interface?**

* answer

### **What is the difference between Abstract class and Interface? Explain with scenarios which one you will it be used with conditions?**

* 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

### **What is the use of static and instance variables in java?**

* answer

### **How to create threads using java 1.7 and java 1.8?**

* answer

### **Which method you will override in order to create a thread?**

* 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

### **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

### **Difference between Map and Filter in java 8?**

* 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

### **What is a predicate?**

* answer

### **What is the difference between Comparable and Comparator interface?**

* 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

### **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

### **What is Functional interface? What is the use of it?**

* 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

### **Have you created a custom exception? What is the purpose of that?**

* answer

### **Difference between Array and Collections?**

* answer

### **Have you ever got outOfMemory exceptions? Can you tell me any scenarios where you have faced this exception?**

* answer

### **If I have a 200 MB excel file so can java read that file?**

* answer

### **What is immutability and mutability in java?**

* answer

### **How can Set not store duplicate values? Explain internal working?**

* answer

### **What is Stream API?**

* answer

### **Have you worked on Collections? How is the difference between Stream and collections?**

* answer

### **How hashcode calculations happened on the Employee object?**

* answer

### **Suppose I have multiple Employee objects with different names, so how does hashmap differentiate between different objects?**

* answer

### **Can we have duplicate keys in hashmap?**

* answer

### **How does hashmap come to know that two objects are the same or different?**

* answer

### **Have you worked on NullPointerException? How to handle NullPointerException in java 8?**

* answer

### **Custom implementation of Singleton beans?**

* answer

### **How to sort the list of objects in java using Stream API?**

* answer

### **What are the practical use cases of LinkedList?**

* answer

### **What is the default scope of beans?**

* answer

### **What is the difference between Process and Thread?**

* answer

### **Explain the Thread Life Cycle?**

* answer

### **What is a Synchronized Thread?**

* answer

### **Have you come across the OutOfMemory exception? How will you handle it? Explain with standalone code?**

* answer

### **What are the ways of blocking the threads in java?**

* answer

### **What is thread synchronization in java?**

* answer

### **What is the necessity of thread synchronization?**

* answer

### **Explain Composition in Oops?**

* answer

### **Difference between static and final?**

* answer

### **Difference between interface and abstract class?**

* answer

### **Where will you use interface and abstract class with real time examples?**

* 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

### **Explain Java memory management?**

* answer

### **What is Stack / Heap memory?**

* answer

### **Where local variables are stored in memory?**

* answer

### **Where objects will be stored in memory?**

* answer

### **Difference between HashMap and ConcurrentHashMap?**

* answer

### **Where will you use ConcurrentHashMap?**

* answer

### **Explain HashMap vs ConcurrentHashMap?**

* answer

### **How is HashMap faster as compared to ConcurrentHashMap?**

* answer

### **Difference between bucket level and segment level locking?**

* answer

### **How is the static method bound with an object?**

* answer

### **What is the difference between Array, ArrayList and LinkedList? When you will use ArrayList and LinkedList?**

* answer

### **What is the use of the intern() method in String class?**

* answer

### **What is encapsulation? How are you currently using it?**

* answer

### **What is the use of Map?**

* answer

### **What is the intermediate and terminal operator in Java stream API?**

* answer

### **Advantages of Stream over for loop?**

* answer

### **What is Optional class? What are the advantages of using Optional class?**

* answer

### **Explain Compile time and Run time polymorphism?**

* answer

### **Optional case returning null and String, how to handle it? What are the advantages of using it?**

* answer

### **What is the use of @FunctionalInterface annotation? Will it be checked manually or by the compiler?**

* answer

### **What is the added version after Java 17 to Java 21?**

* answer

### **What is the inbuilt functional interface in Java 8?**

* answer

### **Difference between map and flatmap?**

* answer

### **Difference between Predicate and Function?**

* 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

# ***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:

* Answer

### **Program – 45**

Problem Definition:

* Answer

### **Program – 46**

Problem Definition:

* Answer

### **Program – 47**

Problem Definition:

* Answer

### **Program – 48**

Problem Definition:

* Answer

### **Program – 49**

Problem Definition:

* Answer

### **Program – 50**

Problem Definition:

* Answer

### **Program – 51**

Problem Definition:

* Answer

### **Program – 52**

Problem Definition:

* Answer

### **Program – 53**

Problem Definition:

* Answer

### **Program – 54**

Problem Definition:

* Answer

### **Program – 55**

Problem Definition:

* Answer

### **Program – 56**

Problem Definition:

* Answer

### **Program – 57**

Problem Definition:

* Answer

### **Program – 58**

Problem Definition:

* Answer

### **Program – 59**

Problem Definition:

* Answer

### **Program – 60**

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

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* 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

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

# Design Patterns

### **What is the singleton design pattern? Explain it?**

* answer

### **Explain Factory design pattern?**

* answer

### **What is the Observer 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

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* 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 association in hibernate?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* 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

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* 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

# 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

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

# Kafka

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* 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

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

# Docker

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

# Kubernetes

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

# AWS

### **What is cloud computing?**

* answer

### **Explain services provided by cloud computing?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

# Azure

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **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

### **question?**

* answer

# React

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* 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

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* answer

### **question?**

* 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?