

- **BY: VAIBHAV SINGH CHAUHAN**
- **Email: vschauhan.8791@gmail.com**
- **Contact: 7042259998**

Hi Everyone,

- I am currently working in AMDOCS DEVELOPMENT CENTRE, PUNE
- As a SOFTWARE DEVELOPER
- My Area of Expertise –
- Language – Java and Java 8
- Framework - Spring Boot, Hibernate
- Design – Microservices
- Deployment Tools - Docker, OpenShift, Jenkins, MAVEN
- Project Management Tools – JIRA, Confluence
- SVN - Bitbucket

All Java Questions are Imp , J2EE and DSA – all ques are not imp

INTERVIEW QUESTIONS:

- Java -
- **What** is Object Oriented Programming
- **What** are main pillars of OOPS and explain them in order with Real Time Example (Please don't use word – selective ignorance anywhere)
- Make an object of Bottle, Pen, Chair etc.
- **Is Inheritance** being necessary for Polymorphism? Why?
- **Necessary** Conditions for Polymorphism?
- Types of Inheritance (IS A relationship)
- HAS A relationship – Composition and Aggregation
- What is Method Signature
- Types of Polymorphism
- Method Overloading vs Method Overriding
- Early Binding vs Late Binding
- Compile time polymorphism and Run time polymorphism
- Method Overloading – Automation promotion of Data types
- Rules for Polymorphism
- Polymorphism with Exception Handling – Rules
- Polymorphism with Static method
- Overriding vs Method Hiding
- Static Polymorphism vs Dynamic Polymorphism
- What is var-args method
- Overriding with respect to Var-args methods
- Overriding with respect to variables
- In how many ways we can create Java objects?? – 5 ways
- Constructors and its type
- Constructor vs Instance block/Static block

- Super (), this () vs super, this (keywords)
- We can't create object for abstract class but abstract class can contain constructor what is the need?
- Can we overload constructor
- Return type of constructor – bluffed question
- Singleton Class – private constructor concept, advantage of singleton class
- We are not allowed to create child class but class is not final, how it is Possible? –
- Without using main() method we can able to print some statement to the console?? How
- Without using System.out.println() statement is it possible to print some statement to the console ?
- Static control flow
- Instance control flow
- Type casting
- Compile time checking vs Runtime checking
- Factory method
- Reflection API and important methods
- Object Class
- 11 methods of object class
- .equals() vs ==
- Contract of .equals vs == , 4 contracts
- Clone()
- Deep vs shallow cloning
- Wait() vs Sleep()
- Notify(), notifyall()
- Final , finally , finalize
- Why String class immutable
- How can we create immutable class
- Wrapper class vs primitive data type
- String vs StringBuilder vs StringBuffer
- Importance of String Constant Pool – its memory diagram
- String literals vs new String(); -- difference basis on hashCode, object in scp or heap , .equals
- NumberFormatException
- ClassCastException
- IllegalArgumentException
- NullPointerException , AIOOB, SIOOB
- Autoboxing Autounboxing
- Widening vs Autoboxing
- Keywords and their deep uses and memory representation – Static, transient , final, const
- Serialization vs Deserialization
- Collection framework – why
- Limitations of Object[] array
- Differences between Arrays and Collections ?
- 9(Nine) key interfaces of collection framework
- Collection ,List ,Set, Queue,Map
- What is the difference between Collection and Collections ?

- In collection framework the following are legacy characters
- ArrayList – Internal Working
- Differences between ArrayList and Vector ?
- Getting synchronized version of ArrayList object
- LinkedList , Vector, Stack
- The 3 cursors of java – Enumeration, Iterator ListIterator
- Comparison of Enumeration , Iterator and ListIterator ?
- Set vs List
- HashSet & LinkedHashSet - differences
- TreeSet
- Null acceptance in Set
- Comparable interface
- compareTo() method analysis vs compare()
- Comparator interface
- Comparison of Comparable and Comparator ?
- Comparison of Set implemented class objects
- Map
- Entry interface
- HashMap
- Differences between HashMap and Hashtable ?
- How to get synchronized version of HashMap
- LinkedHashMap, TreeMap
- Hashtable vs Properties
- Queue interface
- Utility classes - Collections class - Sorting the elements of a List, Searching the elements of a List
- Arrays class, Sorting the elements of array , Searching the elements of array, Converting array to List
- Multithreading Multitasking, Multiprocessing
- The ways to define, instantiate and start a new Thread - By extending Thread class, By implementing Runnable interface
- Thread class constructors
- Thread priority
- Getting and setting name of a Thread.
- The methods to prevent(stop) Thread execution.
- yield() , join(), sleep()
- Synchronization.
- Inter Thread communication.
- Deadlock
- Daemon Threads.
- Various Conclusion
- To stop a Thread
- Suspend & resume of a thread
- Thread group
- Green Thread
- Thread Local

- Life cycle of a Thread
- The way to make an object eligible for GC - Nullifying the reference variable, Reassign the reference variable, Objects created inside a method, Island of Isolation
- The methods for requesting JVM to run GC - By System class, By Runtime class
- Finalization – finalize() method usages -
- Case 1 : Just before destroying any object GC calls finalize() method on the object
- Case 2 : We can call finalize() method explicitly
- Case 3 : finalize() method can be call either by the programmer or by the GC
- Case 4 : On any object GC calls finalize() method only once
- Memory leaks
- Can we call static methods from null object? Syntax to call static methods
- Access specifier and Access modifier
- Public , private , default, protected
- Interface vs Abstract Class
- Why we use interface ?
- Marker Interface
- Adapter Class
- Can we create object of interface?
- Basic idea of Generics and Enum
- RegEx - basics
- Inner classes
- JDBC connection code
- Runtime stack mechanism
- Default exception handling in java
- Exception hierarchy
- Customized exception handling by try catch
- Control flow in try catch
- Methods to print exception information
- Try with multiple catch blocks
- Finally
- Difference between final, finally, finalize
- Control flow in try catch finally
- Control flow in nested try catch finally
- Various possible combinations of try catch finally
- throw keyword
- throws keyword
- Exception handling keywords
- Various possible compile time errors in exception handling
- Customized exceptions
- Top-10 exceptions
- try with resources
- multi catch block
- Exception Propagation
- Rethrowing an Exception
- Throwable – Class or Interface

PROGRAMS:

- Whether the given number is palindrome or not?
- Whether the given String is Palindrome or Not? And Anagrams or Not?
- Divide the given array in such a way that both sub arrays has equal sum.
- Flatten the given array [1,2,3,[4,5,[6,7,8,[9,...]]]] . Your Output should be [1,2,3.....]
- Prime Number programs using Minimum Complexity. And using recursion
- Recursion programs : Fibonaaci , sum of array, reverse number, factorial
- Find the missing element in array if all no are consecutive without using any loop. Array has n-1 elements
- Find the repetative element in array if all no are consecutive without using any loop.
- Array has n+1 element
- Input:
- CONTENTSPHERE output should be max repeating characters E, if VAIBHAV then Output will be AV
- Input : yfgglm , Output will be button ...program should for upper and lowercase both (Hint:a-z,b-y, c-x..)
- Input: aaabbbbcccccaabbbaaaoutput : a3b4c5a3b3a3
- Input : "apple" Output : 'a' at position 0 add 0 , 'p' at position 1 at 1 ..output : "aqroi"
- Program for GCD and LCM of n numbers
- Rotate the 2D array at 90degree
- Print Element of Array without using loop
- All strings Programs without using inbuilt function
- Remove Vowels from given string
- Remove Duplicates in String
- Print Frequency of characters
- Print Distinct Element of an Array

DATA STRUCTURES [Coding neccessary]

- LinkedList all opn,
- Doubly Linked
- List,CircularLL,
- Doubly CircularLL
- Stack,
- Queue,
- Circular Queue,
- Priority Queue Using Array and Linked List
- Sorting algos and complexity
- Using OnePointer Delete the Node of LL
- Binary Search On Unsorted array
- Binary search tree
- Binary tree insertion ,
- traversal (preorder,inorder,postorder)and deletion
- Delete whole tree
- BFS and DFS[Imp]
- Find Element in LL using given pointer
- Find MiddleElement of LL
- Find 3rd Element Of LL

- Reserve the LinkList
- If we want to store planets which Data Structure is best[Ans: LL]
- Hashing , Collision, Chaining, Open Addressing
- [Basic Idea] Graphs and Trees theory

PUZZLES(search keywords in google)

- Water jug 3,5,8
- Scooter Tyre 2.5
- Room with switches
- Candle lighter count hrs
- Divide figures in equal parts
- Matching dots in one go
- 31 kgs rice 1 kg weight
- Tournament knockout round
- Handshake
- Toffee and Wrappers
- 9 balls one defective
- Box and Lock Keys CryptoRSA algo based
- Murderer Puzzle ABCDE
- Search Floor no So egg Thrown from that floor, egg not broke[binary search]

DATABASE MYSQL

- Int vs int32
- Normalization and types
- Join and types examples
- Indexes, views, cursors , triggers
- Function – types
- Stored procedure
- Find second highest sal in 5 different ways
- Find nth highest sal
- Like – its usage
- Date function , current date
- Init function(oracle) in mysql
- DML DDL DCL
- Truncate , drop , delete
- Alter column and table
- Alias
- Orderby, group by , where – order
- Nosql vs SQL
- Find() vs find().pretty()
- Average , sum , min max substring uppercase , lowercase, concate
- Dual row
- Rank ?? limit ?

SPRING

- What is Loose Coupling?
- What is a Dependency?
- What is IOC (Inversion of Control)?
- What is Dependency Injection?
- Can you give few examples of Dependency Injection?
- What is Auto Wiring?
- What are the important roles of an IOC Container?
- What are Bean Factory and Application Context?
- Can you compare Bean Factory with Application Context?
- How do you create an application context with Spring?
- How does Spring know where to search for Components or Beans?
- What is a Component Scan?
- How do you define a component scan in XML and Java Configurations?
- How is it done with Spring Boot?
- What does @Component signify?
- What does @Autowired signify?
- What's the difference Between @Controller, @Component, @Repository, and @Service Annotations in Spring?
- What is the default scope of a bean?
- Are Spring beans thread safe?
- What are the other scopes available?
- How is Spring's singleton bean different from Gang of Four Singleton Pattern?
- What are the different types of dependency injections?
- What is setter injection?
- What is constructor injection?
- How do you choose between setter and constructor injections?
- What are the different options available to create Application Contexts for Spring?
- What is the difference between XML and Java Configurations for Spring?
- How do you choose between XML and Java Configurations for Spring?
- How does Spring do Autowiring?
- What are the different kinds of matching used by Spring for Autowiring?
- How do you debug problems with Spring Framework?
- How do you solve NoUniqueBeanDefinitionException?
- How do you solve NoSuchBeanDefinitionException?
- What is @Primary?
- What is @Qualifier?
- What is CDI (Contexts and Dependency Injection)?
- Does Spring Support CDI?
- Would you recommend to use CDI or Spring Annotations?
- What are the major features in different versions of Spring?
- What are new features in Spring Framework 4.0?

- What are new features in Spring Framework 5.0?
- What are important Spring Modules?
- What are important Spring Projects?
- What is the simplest way of ensuring that we are using single version of all Spring related dependencies?
- Name some of the design patterns used in Spring Framework?
- What do you think about Spring Framework?
- Why is Spring Popular?
- Can you give a big picture of the Spring Framework?

SPRING MVC

- What is Model 1 architecture?
- What is Model 2 architecture?
- What is Model 2 Front Controller architecture?
- Can you show an example controller method in Spring MVC?
- Can you explain a simple flow in Spring MVC?
- What is a ViewResolver?
- What is Model?
- What is ModelAndView?
- What is a RequestMapping?
- What is Dispatcher Servlet?
- How do you set up Dispatcher Servlet?
- What is a form backing object?
- How is validation done using Spring MVC?
- What is BindingResult?
- How do you map validation results to your view?
- What are Spring Form Tags?
- What is a Path Variable?
- What is a Model Attribute?
- What is a Session Attribute?
- What is a init binder?
- How do you set default date format with Spring?
- Why is Spring MVC so popular?

SPRING BOOT

- What is Spring Boot?
- What are the important Goals of Spring Boot?

- What are the important Features of Spring Boot?
 - Compare Spring Boot vs Spring?
 - Compare Spring Boot vs Spring MVC?
 - What is the importance of @SpringBootApplication?
 - What is Auto Configuration?
 - How can we find more information about Auto Configuration?
 - What is an embedded server? Why is it important?
 - What is the default embedded server with Spring Boot?
 - What are the other embedded servers supported by Spring Boot?
 - What are Starter Projects?
 - Can you give examples of important starter projects?
 - What is Starter Parent?
 - What are the different things that are defined in Starter Parent?
 - How does Spring Boot enforce common dependency management for all its Starter projects?
 - What is Spring Initializr?
 - What is application.properties?
 - What are some of the important things that can be customized in application.properties?
 - How do you externalize configuration using Spring Boot?
 - How can you add custom application properties using Spring Boot?
 - What is @ConfigurationProperties?
 - What is a profile?
 - How do you define beans for a specific profile?
 - How do you create application configuration for a specific profile?
 - How do you have different configuration for different environments?
 - What is Spring Boot Actuator?
 - How do you monitor web services using Spring Boot Actuator?
 - How do you find more information about your application environment using Spring Boot?
 - What is a CommandLineRunner?
-
- Database Connectivity - JDBC, Spring JDBC & JPA
 - What is Spring JDBC? How is it different from JDBC?
 - What is a JdbcTemplate?
 - What is a RowMapper?
 - What is JPA?
 - What is Hibernate?
 - How do you define an entity in JPA?
 - What is an Entity Manager?
 - What is a Persistence Context?
 - How do you map relationships in JPA?

- What are the different types of relationships in JPA?
- How do you define One to One Mapping in JPA?
- How do you define One to Many Mapping in JPA?
- How do you define Many to Many Mapping in JPA?
- How do you define a datasource in a Spring Context?
- What is the use of persistence.xml
- How do you configure Entity Manager Factory and Transaction Manager?
- How do you define transaction management for Spring – Hibernate integration?

SPRING DATA

- What is Spring Data?
- What is the need for Spring Data?
- What is Spring Data JPA?
- What is a CrudRepository?
- What is a PagingAndSortingRepository?

Unit Testing

- How does Spring Framework Make Unit Testing Easy?
- What is Mockito?
- What is your favorite mocking framework?
- How do you do mock data with Mockito?
- What are the different mocking annotations that you worked with?
- What is MockMvc?
- What is @WebMvcTest?
- What is @MockBean?
- How do you write a unit test with MockMVC?
- What is JSONAssert?
- How do you write an integration test with Spring Boot?
- What is @SpringBootTest?
- What is @LocalServerPort?
- What is TestRestTemplate?

AOP

- What are cross cutting concerns?
- How do you implement cross cutting concerns in a web application?

- If you would want to log every request to a web application, what are the options you can think of?
- If you would want to track performance of every request, what options can you think of?
- What is an Aspect and Pointcut in AOP?
- What are the different types of AOP advices?
- What is weaving?
- Compare Spring AOP vs AspectJ?

SOAP Web Services

- What is a Web Service?
- What is SOAP Web Service?
- What is SOAP?
- What is a SOAP Envelope?
- What is SOAP Header and SOAP Body?
- Can you give an example of SOAP Request and SOAP Response?
- What is a SOAP Header? What kind of information is sent in a SOAP Header?
- Can you give an example of a SOAP Header with Authentication information?
- What is WSDL (Web Service Definition Language)?
- What are the different parts of a WSDL?
- What is Contract First Approach?
- What is an XSD?
- Can you give an example of an XSD?
- What is JAXB?
- How do you configure a JAXB Plugin?
- What is an Endpoint?
- Can you show an example endpoint written with Spring Web Services?
- What is a MessageDispatcherServlet?
- How do you configure a MessageDispatcherServlet?
- How do you generate a WSDL using Spring Web Services?
- How do you implement error handling for SOAP Web Services?
- What is a SOAP Fault?

RESTful Web Services

- What is REST?
- What are the key concepts in designing RESTful API?
- What are the Best Practices of RESTful Services?
- Can you show the code for an example Get Resource method with Spring REST?
- What happens when we return a bean from a Request Mapping Method?
- What is GetMapping and what are the related methods available in Spring MVC?
- Can you show the code for an example Post Resource method with Spring REST?

- What is the appropriate HTTP Response Status for successful execution of a Resource Creation?
- Why do we use ResponseEntity in a RESTful Service?
- What is HATEOAS?
- Can you give an Example Response for HATEOAS?
- How do we implement it using Spring?
- How do you document RESTful web services?
- Can you give a brief idea about Swagger Documentation?
- How do you automate generation of Swagger Documentation from RESTful Web Services?
- How do you add custom information to Swagger Documentation generated from RESTful Web Services?
- What is Swagger-UI?
- What is "Representation" of a Resource?
- What is Content Negotiation?
- Which HTTP Header is used for Content Negotiation?
- How do we implement it using Spring Boot?
- How do you add XML support to your RESTful Services built with Spring Boot?
- How do you implement Exception Handling for RESTful Web Services?
- What are the best practices related to Exception Handling with respect to RESTful Web Services?
- What are the different error status that you would return in RESTful Web Services?
- How would you implement them using Spring Boot?
- What HTTP Response Status do you return for validation errors?
- How do you handle Validation Errors with RESTful Web Services?
- Why do we need Versioning for RESTful Web Services?
- What are the versioning options that are available?
- How do you implement Versioning for RESTful Web Services?

HIBERNATE

- What is Hibernate ?
- What is HQL in Hibernate?
- Explain the advantages of Hibernate?
- Why is ORM preferred over JDBC?
- Mention the Key components of Hibernate?
- Explain Session object in Hibernate?
- Explain the Transaction object in Hibernate?
- Explain the Criteria object in Hibernate?
- Explain the Query object in Hibernate?
- Mention some of the databases that Hibernate supports?
- What is a One-to-One association in Hibernate?
- What is One-to-Many association in Hibernate?
- What is a Many-to-One association in Hibernate?
- What is Many-to-Many association in Hibernate?
- What is Hibernate caching?

- What is first level cache in Hibernate?
- What is second level cache in Hibernate?
- What is Query level cache in Hibernate?
- Explain Hibernate configuration file and Hibernate mapping file?
- What are concurrency strategies?
- What is Lazy loading in hibernate ?
- Explain the persistent classes in Hibernate?
- Explain some of the elements of hbm.xml?
- Describe the method used to create an HQL Query and SQL Query?
- Explain the important benefits of Hibernate framework?

Array

- Reverse the array
- Maximum and minimum of an array using minimum number of comparisons
- Kth smallest element - .
- Sort an array of 0s, 1s and 2s - .
- Move all negative numbers to beginning and positive to end with constant extra space
- Union of two arrays - .
- Cyclically rotate an array by one - .
- Minimum number of jumps - .
- Find the Duplicate Number - .
- Merge Intervals - .
- Count pairs with given sum - .
- Common elements - .
- Subarray with 0 sum - .
- Factorials of large numbers - .
- Maximum Product Subarray - .
- Longest consecutive subsequence - .
- Given an array of size n and a number k, find all elements that appear more than n/k times
- Array Subset of another array - .
- Find minimum number of merge operations to make an array palindrome
- Median of two sorted arrays of same size
- Median of two sorted arrays of different sizes

Strings

- Reverse String - .
- Palindrome String - .
- Print all the duplicates in the input string
- Why strings are immutable in Java?
- Check if strings are rotations of each other or not - .
- Program to Check if a string is a valid shuffle of two distinct strings
- Longest Palindromic Substring - .
- Print all subsequences of a string

- Longest Repeating Subsequence - .
- Permutations of a given string - .
- Split the binary string into substrings with equal number of 0s and 1s
- Minimum characters to be added at front to make string palindrome
- Print Anagrams Together - .
- Remove Consecutive Characters - .

Searching & Sorting

- First and last occurrences of X - .
- Value equal to index value - .
- Search in Rotated Sorted Array - .
- Count Squares - .
- Square root of an integer
- Zero Sum Subarrays - .
- Product array puzzle - .
- Sort by Set Bit Count - .
- Minimum Swaps to Sort - .

.

Linked-List

- Reverse a linked list - .
- Reverse a Linked List in groups of given size - .
- Detect Loop in linked list - .
- Remove loop in Linked List - .
- Find first node of loop in a linked list
- Remove duplicate element from sorted Linked List - .
- Remove duplicates from an unsorted linked list - .
- Move last element to front of a given Linked List
- Add 1 to a number represented as linked list - .
- Add two numbers represented by linked lists - .
- Intersection of two sorted Linked lists - .
- Intersection Point in Y Shapped Linked Lists - .
- Middle of the Linked List - .
- Merge Sort for Linked List - .
- Quick Sort on Linked List - .
- Check If Circular Linked List - .
- Split a Circular Linked List into two halves - .
- Check if Linked List is Palindrome - .
- Deletion from a Circular Linked List
- Reverse a Doubly Linked List - .

- Find pairs with given sum in doubly linked list
- Count triplets in a sorted doubly linked list whose sum is equal to a given value x
- Sort a k sorted doubly linked list
- Rotate Doubly linked list by N nodes
- Reverse a doubly linked list in groups of given size
- Can we reverse a linked list in less than $O(n)$?
- Why Quick Sort preferred for Arrays and Merge Sort for Linked Lists?
- Flattening a Linked List - .
- Given a linked list of 0s, 1s and 2s, sort it - .
- Clone a linked list with next and random pointer - .
- Merge K sorted linked lists - .
- Multiply two linked lists - .
- Delete nodes having greater value on right - .
- Segregate even and odd nodes in a Link List - .
- Nth node from end of linked list - .
- First non-repeating character in a stream - .

Binary Tree

- Level order traversal - .
- Reverse Level Order Traversal - .
- Height of Binary Tree - .
- Diameter of Binary Tree - .
- Create a mirror tree from the given binary tree
- Inorder Tree Traversal – Iterative and Recursive
- Preorder Tree Traversal – Iterative and Recursive
- Postorder Tree Traversal – Iterative and Recursive
- Left View of Binary Tree - .
- Right View of Binary Tree - .
- Top View of Binary Tree - .
- Bottom View of Binary Tree - .
- ZigZag Tree Traversal - .
- Check for Balanced Tree - .
- Diagonal Traversal of Binary Tree - .
- Boundary Traversal of binary tree - .
- Construct Binary Tree from String with bracket representation
- Binary Tree to DLL - .
- Transform to Sum Tree - .
- Construct Tree from Inorder & Preorder - .
- Minimum swap required to convert binary tree to binary search tree
- Sum Tree - .
- Leaf at same level - .
- Duplicate subtree in Binary Tree - .

- Check if a given graph is tree or not
- Find largest subtree sum in a tree
- Maximum sum of nodes in Binary tree such that no two are adjacent
- Print all k-sum paths in a binary tree
- Min distance between two given nodes of a Binary Tree - .
- Kth ancestor of a node in binary tree
- Duplicate Subtrees - .
- Check if Tree is Isomorphic - .

Binary Search Tree

- Binary Search Tree (Search and Insertion)
- Delete Node in a BST - .
- Minimum element in BST - .
- Predecessor and Successor - .
- Check for BST - .
- Populate Inorder Successor for all nodes - .
- Lowest Common Ancestor in a BST - .
- Construct BST from given preorder traversal
- Binary Tree to BST - .
- Convert a normal BST to Balanced BST
- Merge Two Balanced Binary Search Trees
- Kth largest element in BST - .
- k-th smallest element in BST - .
- Brothers From Different Roots - .
- Inorder Tree Traversal without recursion and without stack!
- Find median of BST in O(n) time and O(1) space
- Count BST nodes that lie in a given range - .
- Replace every element with the least greater element on its right
- Given n appointments, find all conflicting appointments
- Preorder to BST - .
- Check whether BST contains Dead End - .
- Flatten BST to sorted list
- Largest BST - .

Stacks & Queues

- Implement stack using array - .
- Implement Queue using array - .
- Implement two stacks in an array - .
- Design a stack with operations on middle element - .
- How to efficiently implement k stacks in a single array?
- Parenthesis Checker - .

- Reverse a string using Stack
- Inserting at the end of stack
- Reverse a stack using recursion
- Sort a stack - .
- Queue using two Stacks - .
- How to efficiently implement k Queues in a single array?
- Circular Queue
- LRU Cache - .
- Queue Reversal - .
- Reverse First K elements of Queue - .
- Interleave the first half of the queue with second half
- Maximum of all subarrays of size k - .
- Sum of minimum and maximum elements of all subarrays of size k
- First non-repeating character in a stream/**string**