Video 1 - Constructor Interview Questions

1. What are constructors?  
   They are used to create objects of a class. They are always invoked with “new” keyword.  
   There are 2 types of constructors - default constructor and parameterized constructor.  
   Default constructor are the constructors without parameters.  
   Parameterized constructors are the constructor which accepts one or more parameters.
2. Can we overload the constructors?  
   Yes, we can overload the constructor.  
   For Example:   
   Animal a = new Animal(“Dog”);  
   Animal a = new Animal(“Dog”, “Black”);
3. Do you have to explicitly return any value from constructor?  
   No. Compiler returns the reference to the object that is created using the constructor.  
   Also constructor does not have any return type.
4. Are there any rules to define a constructor?  
   a. Constructor must have same name as class name.  
   b. Constructor must not have explicit return type.  
   c. Constructor can’t be abstract static or final.

Video 2 - Polymorphism Interview Questions.

1. What is Polymorphism?  
   Polymorphism is a OOPs concept which means one name but many form.
2. What are different types of Polymorphism?  
   a. Compile time Polymorphism (Achieved using method overloading).  
   b. Runtime Polymorphism (Achieved using method overriding )
3. Why is method overloading not possible just by changing the return type?  
   It can explained with help of example:  
   public int add(int x, int y){return x+y;}

public double add(int x, int y) {return x+y;}

obj.add(4+3);// compiler gets confused

1. What is difference between Method Overloading and Method Overriding?

|  |  |  |
| --- | --- | --- |
|  | Method Overloading | Method Overriding |
| When it happens? | Compile time. Hence called compile time polymorphism | Runtime. Hence called runtime polymorphism. |
| Where it happens? | In same class or within sub class. | Only in sub class. |

1. Can we Overload Main method?  
   Yes. But JVM will always call the main method with below signature.  
   public static void main(String [] args)

Video 3 – Inheritance interview questions

1. What is inheritance?  
   It is a mechanism by which child class acquires properties and behaviour of parent class.
2. What is the use of “extends” keyword?  
   It indicates that we are making a new class that derives properties and behaviour from parent class.
3. What are different types of inheritance in java?  
   a. Single level inheritance

b. Multi level inheritance

c. Hierarchical inheritance. (When 2 or more class inherits from single class)

Video 3 – Public static void main

1. What is public static void main(String [] args)?

public : This access specifier indicates that main method can be called from anywhere.  
static: No object is required to call the main method.

void : Main method does not return any value.

String[] args : arguments passed while running the program.

1. Can we change the order of public, static and void keywords?  
   Yes, but void should come just before method name.

Video 4 – Micro services, Database, Java 8 Interview Questions | TCS Mock Technical Java Interview

1. Introduce Yourself (technical part)  
   My project includes Technologies like Core Java, Spring, Spring Boot, Microservices, Java 8 related features along with mysql db. I am working in finance domains.
2. Why Java is called platform independent?  
   Because we can compile our program on one machine and can run it anywhere.  
   This feature is called Write Once Run Anywhere (WORA).
3. JDK v/s JRE v/s JVM
4. What is Garbage Collection?
5. What is contract between equals and hashcode?  
   The basic rule of the contract states that if two objects are equal to each other based on equals() method, then the hash code must be the same, but vice versa is not true i.e. if the hash code is the same, then equals() can return false.
6. Difference between Abstract class and interface.  
   Both are used to mark abstraction in java.

|  |  |  |
| --- | --- | --- |
|  | Abstract Class | Interface |
| Type Of Method | Abstract & Non Abstract methods. | Only abstract methods.  default and static methods(Java8) |
| Type Of Variable | Final, non final, static and non static variables. | Static and final variables |
| Accessibility of data member | Private, public and default | Public by default. |
| Multiple Inheritance | Does not supports | Supports |
| Multiple Implementation | Can extend one or more interfaces only. | Can extend only one java class and one or more interface. |

1. When to go for Abstract class and when to go interface  
   when Is-A-Relationship (parent-child) exists then prefer going with Abstract class, otherwise prefer Interface.
2. What are feature available in java 8?
3. How to handle Exceptions in java  
   Surrounding our code with try-catch block.   
   (imp) We need to catch more specialized exception first, and then catch generalized exception.
4. What is Dependency Injection (DI) and Inversion Of Control (IOC)?  
   DI: It is a technique in which an object receives other object that it depends on.  
   For example: In a Student class we have a dependency on address, so whenever we try to create object of Student class, the address should be created and should be injected in student object.  
   There are 2 ways in which a dependency can be injected.  
   a. Setter Injection (use setter method for a field)  
   b. Constructor Injection (use Constructor to set values for a field)  
   IOC: Giving control of Object Creation to Spring so that Object can be created and injected into another class. For Example, Instead of creating Address object using new keyword Spring IOC Container creates object of address type and injects into student object. So here we can see that the control is inverted from you creating the object to Spring creating the object
5. What are Application Context and Bean Factory?  
     
   Application Context represents Spring IOC Container  
   Bean Factory is deprecated from Spring 3.0.  
   Current version is Spring 5  
   Info taken from 🡪 https://www.geeksforgeeks.org/spring-difference-between-beanfactory-and-applicationcontext/

|  |  |  |
| --- | --- | --- |
|  | Bean Factory | Application Context |
| Type Of Applications | Suitable to build stand alone applications | Suitable to build web applications integrated with AOP & ORM. |
| Functionality | Fundamental container that provides basic functionality | Advanced container that extends Bean Factory that provides basic + advanced functionality. |
| Support for Annotation | Does not support annotation. In Bean Autowiring, we need to configure the properties in XML file only. | It supports Annotation based configuration in Bean Autowiring. |
| Memory | Requires less memory as it provides basic features | Requires more memory. |

1. What is Aspect Oriented Programming?
2. Why to choose Spring Boot over Spring?  
   There are various configurations that we have to do manually, Spring boot bootstraps all the configurations for us and it makes it very simple to create an application.
3. What are micro services?  
   It is used when we divide our monolithic application into small services which are oriented around only one functionality.
4. What is API gateway?  
   It is the entry point for the micro services, which usually contains the logic which has to implemented for every micro service for example: Authentication, rate limiting(limiting number of requests per minute), caching some of the responses.
5. What is Service Discovery?  
   If one MS wants to discover the URL & communicate to another service then for that we can use service discovery.   
   There are 2 types of service discovery  
   a. Client Side Discovery  
   b. Server Side Discovery  
   Consul(by HashiCorp) and Eureka(by netflix) are used for service discovery.
6. What are ACID Properties?  
   A -> Atomicity(Either a transaction should be completed fully or should rollback completely)  
   C-> Consistency(DB must be consistent before and after transaction)  
   I -> Isolation(Multiple Transactions occur independently without interference)  
   D -> Durability(Changes for successful transaction occurs even if system failure occurs)

Video 5 - Java 8, Spring Boot, database interview Questions | Capgemini Technical Mock Java Interview

1. What is Meta Space in Java8?

Before java8 there was a concept of PermGen, where the java classes were loaded and were never unloaded. From Java 8, we have concept like Meta Space which can load and unload classes.

1. Can we start a thread twice?  
   No. We cannot do that. It may give illlegalThreadStateException
2. How to create Immutable class in java? For more detailed answer refer GFG.  
   a. Make variables private & final.  
   b. Set the value of instance variable only via parameterized constructor.  
   c. Provide only getters and no setters are required.

d. Make class final. Serialization

Byte Stream

Object

1. What is Serialization in java?  
   Suppose we have an object in the heap. And converting this object to byte stream. This object conversion is called serialization. We can store byte stream into our files or we can transfer this byte stream over the network.  
   Serialization only happens if we have default constructor.
2. Suppose we have 5 variables in Object and out of which we want to convert only 3 variables into Byte Stream. How can we achieve this?  
   We can mark a variable as Transient. Via this we ensure that transient variables are not serialized to Byte Class. For example: Passwords can be marked as Transient.
3. What is serialVersionUID?  
   It is an identifier that is used to serialize/deserialize an object of a Serializable class.
4. What is Deserialization?  
   Converting of Byte Stream back to object is called deserialization. After deserialization object is stored back in heap.
5. What is lifecycle of bean?  
   Bean life cycle is managed by the spring container. When we run the program then, first of all, the spring container gets started. After that, the container creates the instance of a bean as per the request, and then dependencies are injected. And finally, the bean is destroyed when the spring container is closed. Therefore, if we want to execute some code on the bean instantiation and just after closing the spring container, then we can write that code inside the init() method and the destroy() method.  
     
   Configuration Techniques  
   a. XML (init-method attribute & destroy-method attribute)  
   b. Spring Interface(Initialization Bean Interface & Disposable Bean Interface)  
   c. Annotation(@PreConstruct & @PreDestroy)
6. Is there any way via which I can run my application on any other server instead of tomcat server?  
   Yes we can change our embedded server
7. How to use Swagger in Spring Boot?

Video 6 - Method Overloading In Java

Video 7 – Method Overriding in Java

Video 8 – String Comparisons in java

Video 9 - Core Java, Spring Boot, Microservice Interview Questions | Cognizant L1 Technical Interview

1. Suppose we have 2 interfaces(I1 & I2) which are inherited in same class. Both have default method with same name(ex: print). How can we call print method of I1?  
   I1.super().print();
2. When to use array list and when to use linked List in java?  
   If we want to frequently add elements, then Linked list is good option.  
   If we want to frequently retrieve elements, then Array list is good option.
3. Explain internal Working of Hashmap? (Imp)  
   We need to cover 3 points on high level basis for this question  
   **a. It works on principle of Hashing.**  
   Hash Map internally works on principle of hashing.  
   Hashing means using some function or algorithm to map object data to some integer value.  
   hashCode() method returns that hash code. Hence it is necessary to write hashCode() method properly for better performance of HashMap.  
   If we override hashCode() method, it is necessary to fulfill Equals and Hashcode Contract.

b. How put method works internally.  
  
MAP is an object, which maps keys to values.  
So there must be some mechanism in HashMap to store this key-value pair.  
Everything in HashMap is stored in a bucket internally.  
Firstly hash value is calculated by calling hashCode() method.  
This hash value is used to calculate index for storing Entry objects.  
  
c. How get method works internally.

1. Difference Between Comparable and Comparator Interface.

|  |  |  |
| --- | --- | --- |
|  | Comparable | Comparator |
| Method used | compareTo() | compare() |
| Package | Java.lang | Java.util |
| Sorting Sequence | Provides Single Sorting Sequence | Provides multiple Sorting Sequence |
| Sorting order | Sort data according to natural sorting order. | Sort data according to customized sorting order |
| Logic of sorting | Logic of sorting must be in same class | Logic of sorting should be in different class |
| Implemented By | Implemented By Calendar, Date & String | Implemented to sort instances of third party instances. |

1. How can we monitor our Spring Boot Application and Micro services?  
   By using actuators. By using Logs.
2. Horizontal Scaling and Vertical Scaling.
3. How we deploy our application if we are scaling the application horizontally?
4. What is Hystrix in Micro Services?  
   It is used to make our application fault tolerant.  
   For Example: In case of cascading calls amongst MS where MS A calls MS B and MS B calls MS C and if MS C fails then MS B fails which lead to failure of MS A. To avoid this cascading failure we use the framework known as hystrix where if MS C fails then MS C return default response and it does not make it as fail. This default response is then passed to MS B then to MS A.

Video 10 – API Gateway in Micro services  
Video 11 – Micro service interview questions

1. What is monolithic architecture and its drawbacks?

In this architecture all features are build in one big application.   
Application is deployed as single unit.   
Any change in requirement requires complete deployment.   
Bug in one module can bring down entire application.  
Scaling of individual component is difficult

1. What is micro service architecture?  
   It is collection of services or small objects.  
   Every service is deployed independently.  
   Individual service is highly maintainable and scalable.
2. How Micro service communicate with each other?  
   We use RestTemplate to communicate. It is synchronous communication.
3. How to handle Fault Tolerance in Micro service?

Fault tolerance means Micro service should continue to work even if some other micro service fails in system.  
It is used to avoid cascading failures.   
Hystrix is used for circuit breaker and return fallback(default) response in case of failure.

Video 12 - Sort List using streams and lambda expression in Java

Suppose an Employee Array List contains 3 records. Each record is having name and salary attribute.  
How can we sort based on salary?

Solution1  
  
List <Employee> sortedList = employees.stream().sorted(new Comparator<Employee>(){

public int compare(Employee e1,Employee e2){

return e1.getSalary() – e2.getSalary();

}

}).collect(Collectors.toList());

Solution 2  
List<Employee> sortedList = employees.stream().sorted((e1,e2) -> e1.getSalary()-e2.getSalary()) .collect(Collectors.toList());

Video 13- Java 8 Stream API and String | Capgemini Mock L1 Coding Interview

1. Suppose there is list of employees (employee class has name, age and salary as properties).  
   Write a Program in which we have to increase salary of employees by 10% if there age is greater than 25.

List <Employee> newEmpList = empList.stream().map(e->{

If(e.getAge()>25){ e.setSalary(e.getSalary()\*1.10);}

return e;

}).collect(Collectors.toList());

1. a. String str1 = “ABC”;   
   b. String str2 = “ABC”;  
   c. String str3 = new String(“ABC”);

a & b will be created in Spring pool and are called string literals

c will be created in Heap.

str1 == str2 // true;

str1 == str3 //false

str1.equals(str3)// true

Video 14 – (Imp) Sort HashMap By Value | Infosys L1 Mock Coding Interview

1. Suppose we are given a map and we want to sort it via value.  
   LinkedHashMap <String, Integer> sortedMap = map.entrySet().stream().sorted( (e1,e2) -> {

return e1.getValue() - e2.getValue();

}).collect(Collectors.toMap(Map.Entry::getKey,Map.Entry::getValue, (e1,e2)->e1,LinkedHashMap::new));

Note: *in collectors.toMap() we have passed 4 parameters*

1. Map.Entry::getKey 🡪 Indicate how to get key for map
2. Map.Entry:: getValue 🡪 Indicate how to get value for map
3. (e1,e2) -> e1 -> function that indicates that, in the case of a collision, we keep the existing(e1) entry.
4. By default, a toMap() method will return a HashMap.for returning LinkedHashMap we use LinkedHashMap::new

Video 15 - Find Palindrome words from sentence | Cognizant L1 Java Coding Interview

String str = “My name is nitin and I can speak malayalam”;  
String Words = str.split(“ ”);

public boolean isPallindrome(String s) {

int i1 = 0;

int i2 = s.length()-1;

while(i1<i2) {

if(s.charAt(i1)!=s.charAt(i2)) return false;

i1++;

i2--;

}

return true;

}

Words.stream.forEach (word -> {

If(isPallindrome(word)) { System.out.println(word); }

});

Video 16 – Java8 Streams | Group By List of Employees By City | Accenture Mock L1 Coding Interview

1. Employee class has 2 attributes (name and city) and we want to groupBy employees based on city.  
   Input :   
   Name : Amar, City: Pune  
   Name: Raj, City: Pune

Name: Neha, City: Mumbai

Name: Sam, City: Mumbai

Output:   
Pune : Amar, Raj  
Mumbai: Neha, Sam

Solution:  
Map<String, List<employees>> groupedBy = employees.stream().collect(Collectors.groupingBy(Employee::getCity));  
Note : Collectors.groupingBy by default returns a Map and always take a value via which we can group a list of collection.

Video 17: Java Program to check Armstrong number | Amdocs Mock Coding Round

Video 18: IBM Mock Interview 2 Years Experience

1. How to avoid collisions in hash map  
   Write good hash functions so that it return unique hash code for different keys.

Video 19: CapG MS Mock Interview

1. Design System architecture for E-Commerce application such as Amazon or Flipkart.
2. What kind of DB should be used?  
   If there is some structure in data then use RDBMS, if there is no structure then use NoSQL DB.

Video 20: Accenture Java Interview 2022

1. Features of Java 8
2. Parallel Stream v/s Sequential Stream

|  |  |  |
| --- | --- | --- |
|  | Sequential | Parallel |
| Core | Runs on single core and one thread | Runs on multiple core and multiple thread. |
| Type of O/P | O/P is predictable | O/P is not predictable |
| Performance | Poor Performance | High Performance |

1. Example of terminal & intermediate operation available on stream  
   Intermediate: - map(), filter(), distinct(), sorted()  
   Terminal:- forEach(),reduce(), collect(), min(), max(), count(), anyMatch(), allMatch(), noneMatch(), findFirst(), findAny()
2. How to Insert a record in DB(only theoretical explanation)?  
   a. Create Spring Boot Project(from Spring Initialiser). Add web dependency and other necessary dependencies. Download the zip file and import it via Intellij.  
   b. Create a rest controller. Annotate it with @RestController.

c. Create a service class for writing business logic. Annotate it with @Service.  
d. Create a DAO class. Annotate it with @Repository.  
e. Create Modal classes which actually hold type of data.  
f. Create methods in controller to perform CRUD operations and annotate it with @GetMapping, @PostMapping, @PutMapping, @DeleteMapping.

1. Suppose there is only one DB and it goes down….so what will be our approach to solve this problem?  
   For this we can follow a master slave kind of architecture.  
   So if master goes down then slave should comes up.
2. How we are deploying our micro services?
3. Difference Between Authentication and Authorization
4. Write a program to find max number from a list  
   Integer max = numbers.stream().max((I,j)->i.compareTo(j)).get();

Video 21:- MasterCard Java Interview 2022.

1. What is Optional Class in java?  
   Every Java Programmer is familiar with NullPointerException. It can crash your code. And it is very hard to avoid it without using too many null checks. So, to overcome this, Java 8 has introduced a new class Optional in java.util package. It can help in writing a neat code without using too many null checks. By using Optional, we can specify alternate values to return or alternate code to run. This makes the code more readable because the facts which were hidden are now visible to the developer.  
   Some important methods in Optional Class are

a. isPresent() - Return true if there is a value present, otherwise false.

b. get() - If a value is present in this Optional, returns the value, otherwise throws NoSuchElementException.

c. ofNullable(T value) - if non-null, Returns an Optional describing the specified value otherwise returns an empty Optional.

1. What is Functional Interface
2. What is predicate?
3. What is method Reference?
4. How to convert stream to an array in java8?   
   list.stream().toArray();
5. What is fail-fast and fail-safe in java?  
   While iterating through the collection and if another thread tries to modify the list on which we are iterating, in that case there is concurrent modification exception and it is called fail-fast.  
   Fail-Fast happens on Arraylist, Linkedlist  
   While iterating over a collection we are given a copy of collection, so we don’t iterate on original collection, we iterate on copy of collection. By this we our iteration is not hampered. This is called fail-safe. Fail Safe happens on concurrent hashmap, copy-on-write arraylist.
6. What is CI/CD?  
   Continous Integration and Continous Deployment.
7. Write a code to generate random numbers.  
   Here we are going to use Java’s own Random class.   
   Random random = new Random();  
   random.ints().forEach(System.out::println);
8. Write a program to write only 10 numbers.  
   random.ints().limit(10).forEach(System.out::println);
9. Write a program to print 10 number between 1 & 100.  
   random.ints(1,100).limit(10).forEach(System.out::println);
10. Write a program to sort the above randomly generated numbers.  
    random.ints(1,100).limit(10).sorted().forEach(System.out::println);

Video 22 – Java 8, Streams, Concurrent Hashmap | Infosys Java Interview.

1. findFirst() v/s findAny()

**findFirst()** method finds the **first element** in a Stream. So, we use this method when we specifically want the first element from a sequence. It returns an optional.

**findAny()** method allows us to **find any element** from a Stream. We use it when we're looking for an element without paying an attention to the encounter order. It also returns an Optional

1. What is ConcurrentModificationException?  
   It occurs when we try to modify a collection concurrently.   
   For example: While iterating any collection if we try to modify the structure or add or remove elements then it may result in ConcurrentModificationException.
2. In continuation to above question, In concurrentHashMap we will get concurrentModificationException . what if I use HashTable which is also thread safe. Will we get concurrentModificationException?  
   If multiple threads try to accept the hashtable, the thread will acquire a lock on entire table, but in case of concurrentHashMap, if there are 16 segments, thread will be acquired on only one segment and rest of segments are available for changes.
3. Difference b/w ConcurrentHashmap and ConcurrentMap
4. Many questions on ConcurrentHashMap
5. Why wait() and notify() method belongs to object and not to thread class?  
   Because locking mechanism belongs to Object class and not Thread class.
6. What is deadlock in database

Video 23 – TCS java 8 Stream Coding Interview

1. Given a list of integers, write a program to find list of integers that starts with 1.  
   numbers  
   .stream()  
   .map(num-> “”+num)  
   .filter(num->num.startsWith(1))  
   .forEach(System.out::println);
2. Reverse an integer number.

Video 24- Find Highest Paid employee from Department | Parallel & Sequential Stream

1. Input:  
   List<Employee> employees = Arrays.asList(  
   new Employee(“Emp1”, “CS”, 10000),  
   new Employee(“Emp2”, “CS”, 15000),  
   new Employee(“Emp3”, “IT”, 20000),  
   new Employee(“Emp4”, “IT”, 25000)  
   );

Output: To find highest employee in each department.  
CS: Employee(“Emp2”, “CS”, 15000)  
IT: Employee(“Emp4”, “CS”, 25000)

Solution: Employees.stream().collect(Collectors.groupingBy(e -> e.getDepartment()), Collectors. collectingAndThen(Collectors.maxBy(Comparator.compareInt(e->e.getSalary())),Optional::get))

1. Practical Implementation of How Parallel Stream is different from Sequential Stream.  
   List<Integer> numbers = Arrays.asList(1,2,3,4,5,6,7,8,9);  
   //Sequential Stream  
   numbers.stream().forEach(num -> {  
    System.out.println(e+ “ ” + Thread.currentThread().getName());  
   });

//Parallel Stream

numbers.parallelStream().forEach(num -> {  
 System.out.println(e+ “ ” + Thread.currentThread().getName());  
});

Video 25: Java8 Coding Interview | L2 Capgemini Java Interview  
 List <Employee> employees = new ArrayList<Employee>();  
 employees.add(new Employee(“ABC”,30 ,“Female”, “HR”));  
 employees.add(new Employee(“PQR”,25 ,“Male”, “IT”));  
 employees.add(new Employee(“LMN”, 30, “Male”, “HR”));  
 employees.add(new Employee(“XYZ”, 28, “Female”, “IT”));

1. Find List of distinct department.  
   employees.stream().map(Employee:: getDepartment).distinct().Collect(Collectors.toList())
2. Find Count Of Employees working in each department  
   Map <String, Long> employeeCount = employees.stream().collect(Collectors.groupingBy(Employee::getDepartment, Collectors.counting()))
3. Find Average age of Male Employee and female employee.  
   Map<String, Double> avgAge = employees.stream().collect(Collectors.groupingBy(Employee::getGender, Collectors.averagingDouble(Employee::getAge)));
4. Find Average Age Of all Employees.  
   employees.stream().collect(Collectors.averagingDouble(Employee::getAge));

Video 26: CapG Java8 Interview.

Video27: Java8 Coding Interview Round Questions

1. From a list of numbers find the list of non duplicate integers.  
   List<Integer> numbers = Arrays.asList(10,20,27,25,20,25,30);  
   Set <Integer> hs = new HashSet<Integer>();  
   numbers.stream().filter(num ->hs.add (num) ).collect(Collectors.toList());
2. Find the list of duplicate integers from list.  
   in above example if we add this particular condition  
   numbers.stream().filter(num->!hs.add(num)).collect(Collectors.toList());
3. Sort the list of Integers in descending order.  
   numbers.stream().sorted(Collections.reverseOrder()).collect(Collectors.toList());

Video 28: TCS Java Interview

1. What are different types of Memory area allocated by JVM?  
   a. Class Area  
   b. Heap  
   c. Stack   
   d. Program Counter Register
2. What is JIT Compiler?  
   It stands for Just In Time Compiler
3. What is the component that loads our class in Java?  
   Classloader(Can be explained more along with type of classloader).

Video 29: Infosys Java Coding Round

1. Given two arrays. Find common numbers from these two arrays/ intersection b/w 2 arrays.  
   Set <Integer> s = new HashSet<>();  
   for (int i = 0; i<arr1.length;i++){s.add(arr[i]);}  
   for(int i = 0;i<arr2.length;i++ ){  
   if(s.contains(arr2[i])){System.out.println(“ ”+arr2[i]);}  
   }
2. Given two arrays. Find uncommon numbers from these two arrays/ union b/w 2 arrays.  
   Set <Integer> s = new HashSet<>();  
   for (int i = 0; i<arr1.length; i++){ s.add(arr[i]); }  
   for(int i = 0; i<arr2.length; i++){ s.add(arr[i]); }  
   System.out.println(s);

Video 30: TCS Coding interview

1. From a given string we need to count the number of special character and the string without special character.

String s = “Shivam @123!”;

String stringWithSpecialCharacterRemoved = “”;

int count = 0;  
for (int i = 0; i < s.length(); i++) {

if(!Character.isDigit(s.charAt(i)) && !Character.isLetter(s.charAt(i)) && !Character.isWhiteSpace(s.charAt(i))) {count++}

}

else {

stringWithSpecialCharacterRemoved = stringWithSpecialCharacterRemoved+s.charAt(i);

}  
System.out.print(“Count is ”+count);

1. 2 Strings are given to us. How to find whether these both strings are rotations of each other.  
   use this following condition 🡪 (str1+str2).subString(str2)

Video 31: Capgemini Java Coding Interview

1. First n natural numbers are present in an array except one natural number, which is missing. Write a program to find that natural number.  
   int sumOfNaturalNumber = (n\*(n+1))/2;  
   int actualSum = 0;

for (int i = 0;i<arr1.length;i++){

actualSum = actualSum+arr1[i];

}

int missingNumber = sumOfNaturalNumber – actualNumber;

1. Consider a given string and modify the string in such a way that first half of string should be in lowercase and second half of string should be in uppercase. (DIY)

Video 32: Infosys Java Coding Interview | Round2

Video 33: Accenture java Coding Interview

1. We are given a String(str) and we have to remove all the occurrence of a particular character(c).

String newString = “”;  
for (int i =0; i < str.length;i++){  
if(str.charAt(i)!=c) {newString = newString+str.charAt(i);}

}  
System.out.println(“”+newString);

1. We are given two arrays. Find whether these 2 arrays are same or not?  
   if (arr1.length()! = arr2.length() ) return “Arrays are not same”;

Set <Integer> hs = new HashSet<integer>();

for(int i = 0;i< arr1.length;i++) {  
hs.add(arr[i]);  
}

bool same = true;

for (int i = 0; i< arr2.length; i++) {

if(!hs.contains(arr2[i])){same = false;break;}

}

If(same == false){return “Array not same”;}

Video 34: Cognizant Java Coding Interview