1. Write code to reverse a linked list, if you able to do it using loops, try to solve with recursion.(solution) and also reverse a string using the recursion method.

<https://www.youtube.com/watch?v=tQur3kprZQ>

<https://www.geeksforgeeks.org/reverse-a-linked-list/>

<https://www.youtube.com/watch?v=O0By4Zq0OFc>

<https://www.geeksforgeeks.org/reverse-a-string-using-recursion/>

2) How to rotate an array by K? (solution)  
  
3) Given an array which contains N-2 numbers in unsorted order, find two missing numbers? ([solution](http://java67.blogspot.com/2014/12/how-to-find-missing-number-in-sorted.html))

<https://www.geeksforgeeks.org/find-two-missing-numbers-set-1-an-interesting-linear-time-solution/>

4) Given an array which has a number in the majority (a number appears more than 50% in the array), find that number? (solution)  
  
5) How to detect a loop in singly linked list? If you are able to detect loop then find the size of linked list? ([solution](http://javarevisited.blogspot.com/2013/05/find-if-linked-list-contains-loops-cycle-cyclic-circular-check.html))  
  
6) Given N steps to climb to reach a floor, a person can take 1 or 2 step at a time to climb. Find the number of ways to reach nth step? (discussion)

<https://www.geeksforgeeks.org/count-ways-reach-nth-stair/>

<https://www.geeksforgeeks.org/count-ways-reach-nth-stair-using-step-1-2-3/>

7) Design Snake and Ladder Game by using Object Oriented analysis and design technique. Explain the reasoning behind interface and class structure? (discussion)  
  
8) How to merge two sorted linked list, write a program in your favorite programming language e.g. Java, C++, or Python.  
  
9) Write a Program which checks if two Strings are Anagram or not? ([solution](http://javarevisited.blogspot.com/2013/03/Anagram-how-to-check-if-two-string-are-anagrams-example-tutorial.html))

<https://www.geeksforgeeks.org/check-whether-two-strings-are-anagram-of-each-other/>

10) How to print all permutations of a given String using recursion? ([solution](http://javarevisited.blogspot.com/2015/08/how-to-find-all-permutations-of-string-java-example.html))

Permutation of all.

Permutation of any string will be total combination of !<string length> (factorial of string length)

<https://www.geeksforgeeks.org/write-a-c-program-to-print-all-permutations-of-a-given-string/>

11) How to check if a binary tree is balanced or not? (solution)  
  
12) How to swap two numbers without using a temp variable, write code which is free from Integer overflow? ([solution](http://java67.blogspot.com/2015/08/how-to-swap-two-integers-without-using.html))  
  
13) How to find all pairs of elements in an integer array, whose sum is equal to a given number? ([solution](http://javarevisited.blogspot.com/2014/08/how-to-find-all-pairs-in-array-of-integers-whose-sum-equal-given-number-java.html))  
  
14) Write a program to check if a binary tree is BST or not? (solution)  
  
15) Write a function to print nth number in Fibonacci series? ([answer](http://javarevisited.blogspot.com/2015/01/print-fibonacci-series-in-java-using.html))

<https://www.geeksforgeeks.org/program-for-nth-fibonacci-number/>

Two way we can write this programme, using recursion and dynamic progrvvaming.

17) Write a function to count a total number of set bits in a 32 bit Integer? ([solution](http://javarevisited.blogspot.com/2014/06/how-to-count-number-of-set-bits-or-1s.html))  
  
18) Write code to implement an LRU cache? (solution)  
  
19) Write a function to remove duplicate characters from String? ([solution](http://java67.blogspot.com/2014/03/how-to-find-duplicate-characters-in-String-Java-program.html))  
  
20) How to find the 3rd element from end, in a singly linked, in a single pass? ([solution](http://javarevisited.blogspot.com/2012/12/how-to-find-middle-element-of-linked-list-one-pass.html))  
  
  
  
Read more: <https://javarevisited.blogspot.com/2016/01/top-20-amazon-and-google-programming-interview-questions.html#ixzz5WC6KMIfS>.

Java general questions:

1. What is marker interface? Can we create user defined marker interface?

https://www.geeksforgeeks.org/marker-interface-java/

1. Serialization and deserialization in java

<https://www.geeksforgeeks.org/serialization-in-java/>

1. How many ways we can break singleton and what is the way to prevent them.

Reflection, Serialization and Cloning can break the Singleton pattern.

<https://www.geeksforgeeks.org/prevent-singleton-pattern-reflection-serialization-cloning/>

1. When class load exception will occur in java?

If the class which we want to use not present in the classpath or all dependent class is not present in the containing jar and same class we are using in our programme.

1. Can abstract class have public constructor?

**Yes**! **Abstract classes can have constructors**!

Yes, when we define a class to be an Abstract Class it cannot be instantiated but that does not mean an Abstract class cannot have a constructor. Each abstract class must have a concrete subclass which will implement the abstract methods of that abstract class.

When we create an object of any subclass all the constructors in the corresponding inheritance tree are invoked in the top to bottom approach (also known as constructor chaining). The same case applies to abstract classes. Though we cannot create an object of an abstract class, when we create an object of a class which is concrete and subclass of the abstract class, the constructor of the abstract class is automatically invoked. Hence we can have a constructor in abstract classes.

Note: A non-abstract class cannot have abstract methods but an abstract class can have a non-abstract method. Reason is similar to that of constructors, difference being instead of getting invoked automatically we can call super(). Also, there is nothing like an abstract constructor as it makes no sense at all.

Public constructor in the abstract class make no sense as we need to call these constructor always from the derived class or concrete class, so better we can use protected access modifier.

1. Determine if a string have all unique characters? Without using any data structure.

<https://www.geeksforgeeks.org/determine-string-unique-characters/>

we have four approaches to solve this problem.

1. Brute Force Technique: time complexity of this approach is O(n^2), in this approach we use two loop, and compare each character to adjacent charcter of the string.
2. Sorting Technique: time complexity of this approach is O(nlogn) which internally uses binary sort.

In this approach we convert all the characters of the string into the char array and then sort it. And using single loop we checked if each character is different to its adjacent character and then return accordingly.

1. Use of extra data structure :this technique use ASCII char set (8 bits). It is for to maintain the Boolean array for the characters. The 256 indices indicates 256 characters. All the array elements initially set to false.

As we iterate over the string, set true at the index equal to the int value of the character. If at any time, we encounter that the array value is already true, it means the character with that int value is repeated.

1. Without extra data structure: this technique is valid for string having only alphabets.
2. Given two string find out the one string is a permutation of other string?

<https://www.geeksforgeeks.org/check-if-two-strings-are-permutation-of-each-other/>

1. URLify a given string

<https://www.geeksforgeeks.org/urlify-given-string-replace-spaces/>

1. Palindrome Permutation

<https://leetcode.com/articles/palindrome-permutation/>

**Java 8 feature programming example link**

<https://www.journaldev.com/2389/java-8-features-with-examples>

Static and default method have been introduced in the Java 8.

//trying to override Object method gives compile time error as

//"A default method cannot override a method from java.lang.Object"

// default String toString(){

// return "i1";

// }

<https://www.journaldev.com/2752/java-8-interface-changes-static-method-default-method>.

We can instantiate an interface with the anonymous class

Runnable r = new Runnable(){

@Override

public void run() {

System.out.println("My Runnable");

}};

Lambda Expression : lambda expressions are means to create anonymous classes of functional interfaces easily. There are no runtime benefits of using lambda expressions.

So use it carefully.

Some of the useful java 8 functional interfaces are Consumer, Supplier, Function and Predicate.

Lambda Expression are the way through which we can visualize **functional programming** in the java object oriented world.

Java language provide support for using lambda expressions only with functional interfaces.

**Why do we need Lambda Expression?**

1. Reduced lines of codes: one of the best example is functional interface rather than using anonymous java class
2. Sequential and Parallel Execution Support: using lambda expression we can benefit from STREAM API sequential and parallel operation support.
3. Passing behaviours into methods.
4. Higher efficiency with laziness.

**Operators in Java 8**

:: double colon operator, it is used for the method reference its short form of lambda expression is i-> method name.

**Lambda Expression Example:**

() -> {} //no parameters, void result

() -> 42 // no parameters, expression body

() ->{return 42;}// no parameters, block body with return.

//complex block body with multiple return

() -> { if (true) return 10; else { int result = 15; for (int i = 1; i < 10; i++) result \*= i; return result; } }

(int x) -> x+1;//Single declared type argument.

(x) -> x+1 //Single inferred type argument

X -> x+1 //Parenthesis optional for single inferred-type case.

**Method and constructor references**

A method reference is used to refer to a method without invoking it.

A constructor reference is similarly used to refer to a constructor without creating a new instance of the named class or array type.

System::getProperty //method reference

System.out::println //method refrence

“abc”::length //method reference

ArrayList::new //method reference

Int[]::new

**Lambda expression tutorial**

<https://www.journaldev.com/2763/java-8-functional-interfaces>

Oracle related question

<https://stackoverflow.com/questions/20091125/nth-max-salary-in-oracle>

We can find out using the rank(), row\_number() in sub query and others also.

**Functional Programming vs Object Oriented Programming:**

If we look into some other programming languages such as C++, JavaScript; they are called **functional programming language** because we can write functions and use them when required.

Some of these languages support Object Oriented Programming as well as Functional Programming.

Java 7 updates

String in switch statement.

Binary Literal.

The try-with-resources or ARM (Automatic resource management).

Catching multiple exceptions by single catch.

Underscores in numeric literals.

**Multithreading interview questions**

<https://www.journaldev.com/1162/java-multithreading-concurrency-interview-questions-answers>

<https://www.educba.com/java-multi-threading-interview-questions/>

* Many other java threads running in background like memory management, system management, signal processing etc.
* Callable interface uses Generics, so it can return any type of Object.
* *Runnable* tasks can be run using the *Thread* class or *ExecutorService* whereas*Callables* can be run only using the latter.
* *Runnable*is the core interface provided for representing multi-threaded tasks and *Callable* is an improved version of *Runnable* that was added in Java 1.5.

**Sorting Algorithm:**

Merge Sort:

<https://www.geeksforgeeks.org/merge-sort/>

Attempted first time without hint programme was not correct next time need to do the practice with help of above link.

**Programming Coding questions Link**

<https://www.geeksforgeeks.org/must-do-coding-questions-for-companies-like-amazon-microsoft-adobe/>