1. **Concept Of multi-threading**

Multithreading is a Java feature that allows concurrent execution of two or more parts of a program for maximum utilization of CPU. Each part of such program is called a thread. So, threads are light-weight processes within a process.  
  
Threads can be created by using two mechanisms :  
1. Extending the Thread class  
2. Implementing the Runnable Interface

For Code Refer Git Java Repo.

**Thread Class vs Runnable Interface**  
  
1. If we extend the Thread class, our class cannot extend any other class because Java doesn’t support multiple inheritance. But, if we implement the Runnable interface, our class can still extend other base classes.  
  
2. We can achieve basic functionality of a thread by extending Thread class because it provides some inbuilt methods like yield(), interrupt() etc. that are not available in Runnable interface.

1. **LinkedList, Linked List Vs List , Generic Syntax of linked List**

LinkedList in Java

Linked List are linear data structures where the elements are not stored in contiguous locations and every element is a separate object with a data part and address part. The elements are linked using pointers and addresses. Each element is known as a node. Due to the dynamicity and ease of insertions and deletions, they are preferred over the arrays. It also has few disadvantages like the nodes cannot be accessed directly instead we need to start from the head and follow through the link to reach to a node we wish to access.  
To store the elements in a linked list we use a doubly linked list which provides a linear data structure and also used to inherit an abstract class and implement list and deque interfaces.

In Java, LinkedList class implements the [list interface](https://www.geeksforgeeks.org/list-interface-java-examples/). The LinkedList class also consists of various constructors and methods like other java collections.

**Constructors for Java LinkedList:**

LinkedList(): Used to create an empty linked list.

LinkedList(Collection C): Used to create a ordered list which contains all the elements of a specified collection, as returned by the collection’s iterator.

**Linked List vs Array**

Both [Arrays](https://www.geeksforgeeks.org/array-data-structure/) and [Linked List](https://www.geeksforgeeks.org/data-structures/linked-list/)can be used to store linear data of similar types, but they both have some advantages and disadvantages over each other.





Comparision between [ArrayList](https://www.geeksforgeeks.org/arraylist-in-java/" \t "_blank)and [LinkedList](http://geeksquiz.com/linked-list-in-java/):-

1. Insertions are easy and fast in LinkedList as compared to ArrayList because there is no  
   risk of resizing array and copying content to new array if array gets full which makes  
   adding into ArrayList of O(n) in worst case, while adding is O(1) operation in LinkedList  
   in Java. ArrayList also needs to be update its index if you insert something anywhere except  
   at the end of array.
2. Removal also better in LinkedList than ArrayList due to same reasons as insertion.
3. LinkedList has more memory overhead than ArrayList because in ArrayList each index only  
   holds actual object (data) but in case of LinkedList each node holds both data and address  
   of next and previous node.
4. Both LinkedList and ArrayList require O(n) time to find if an element is present or not. However we can do Binary Search on ArrayList if it is sorted and therefore can search in O(Log n) time.



