# Collections



# ArrayList

- ArrayList is a part of <u>collection framework</u> and is present in java.util package.
- It provides us dynamic arrays in Java.
- ArrayList inherits AbstractList class and implements List interface.
- ArrayList is initialized by a size, however the size can increase if collection grows or shrunk if objects are removed from the collection.
- Java ArrayList allows us to randomly access the list.

#### **Constructors in Java ArrayList:**

- ArrayList(): This constructor is used to build an empty array list
- ArrayList(Collection c): This constructor is used to build an array list initialized with the elements from collection c
- ArrayList(int capacity): This constructor is used to build an array list with initial capacity being specified

- import java.util.ArrayList;
- class Arr
- {

• {

- public static void main(String arg[])
- ArrayList<String> list=new ArrayList<>();
- list.add("India");
- list.add("usa");
- list.add("China");
- for(String a:list)
- {

• }

• }

• }

System.out.println(a);

## LinkedList

- Java LinkedList is a doubly linked list implementation of Java's List and Deque interfaces. It is part of Java's collections framework.
- Java LinkedList class can contain duplicate elements and null values.
- Java LinkedList is not thread-safe.
- A linked list is a linear data structure where each element is a separate object.

## **Types of Linked Lists**

singly linked list



• doubly linked list



#### Internal Working of LinkList:-



import java.util.\*;
public class Linklst{
 public static void main(String args[]){

LinkedList<String> al=new LinkedList<String>(); al.add("Deep"); al.add("Mani"); al.add("Harry");

Iterator<String> itr=al.iterator();
while(itr.hasNext()){
 System.out.println(itr.next());

#### Difference Between Arraylist and Linkedlist



Java LinkedList Representation



## Difference Between Arraylist and Linkedlist

ArrayList	LinkedList
1) ArrayList internally uses <b>dynamic</b> <b>array</b> to store the elements.	LinkedList internally uses <b>doubly linked</b> <b>list</b> to store the elements.
2) Manipulation with ArrayList is <b>slow</b> because it internally uses array. If any element is removed from the array, all the bits are shifted in memory.	Manipulation with LinkedList is <b>faster</b> than ArrayList because it uses doubly linked list so no bit shifting is required in memory.
<ol> <li>ArrayList class can act as a list only because it implements List only.</li> </ol>	LinkedList class can <b>act as a list and</b> <b>queue</b> both because it implements List and Deque interfaces.