java.util.LinkedList:-

public class java.util.LinkedList ***extends*** *java.util.AbstractSequentialList*
implements *java.util.List<E>,*
java.util.Deque<E>,
java.lang.Cloneable,
java.io.Serializable

1) Introduced in 1.2 version.
2) Heterogeneous objects are allowed.
3) Null insertion is possible.
4) Insertion order is preserved.
5) LinkedList methods are non-synchronized.
6) Duplicate objects are allowed.
7) The under laying data structure is double linkedlist.
8) cursors :- Iterator,ListIterator

constructors:-

LinkedList(); *it builds a empty LinkedList.*

LinkedList(java/util/Collection<? extends E>);

it builds a LinkedList that initialized with the collection data.

Example:- LinkedList basic operations.

```
import java.util.*;
class Test
{       public static void main(String[] args)
        {       LinkedList<String> l=new LinkedList<String>();
                l.add("B");
                l.add("C");
                l.add("D");
                l.add("E");
                l.addLast("Z");//it add object in last position
                l.addFirst("A");//it add object in first position
                l.add(1,"A1");//add the Object spcified index
                System.out.println("original content:-"+l);
                l.removeFirst();        //remove first  Object
```

```
                l.removeLast();          //remove last t Object
                System.out.println("after deletion first & last:-"+l);
                l.remove("E");           //remove specified Object
                l.remove(2);             //remove the object of specified index
                System.out.println("after deletion :-"+l);//A1 B D
                String val = l.get(0);  //get method used to get the element
                l.set(2,val+"cahged");//set method used to replacement
                System.out.println("after seting:-"+l);
        }
};
```

D:\>java Test
original content:-[A, A1, B, C, D, E, Z]
after deletion first & last:-[A1, B, C, D, E]
after deletion :-[A1, B, D]
after seting:-[A1, B, A1cahged]

Example:-Adding one collection data into another Collection.

```
import java.util.*;
class Test
{       public static void main(String[] args)
        {       ArrayList<String> al = new ArrayList<String>();
                al.add("ratan");
                al.add("balu");

                LinkedList<String> linked = new LinkedList<String>(al);
                linked.add("anu");
                linked.add("simran");
                System.out.println(linked);
        }
}
```

E:\>java Test
[ratan, balu, anu, simran]

Example :- LinkedList cloneing process:-

```
import java.util.*;
class Test
{       public static void main(String[] args)
        {
         LinkedList<String> arrl = new LinkedList<String>();
         arrl.add("First");
        arrl.add("Second");
        arrl.add("Third");
        arrl.add("Random");
        System.out.println("Actual LinkedList:"+arrl);
        LinkedLis copy = (LinkedLis) arrl.clone();
        System.out.println("Cloned LinkedList:"+copy);
   }
}
```

E:\>java Test
Actual LinkedList:[First, Second, Third, Random]
Cloned LinkedList:[First, Second, Third, Random]