Vector Class:

- Vector is available since jdk1.0V

- It allows duplicates & insertion order is maintained.

- Default capacity when creating an Vector is 10.

- Its capacity increases by (CurrentCapacity*2).

- It is synchronized by default.

 Vector v=new Vector();

  Vector v=new Vector(int capacity);

 Vector v=new Vector(int capacity, int incrementalcapacity);

Vector Methods

Method	Description
addElement(Object o);	Adds the specified component to the end of this vector, increasing its size by one.
removeElement(Object o);	Removes the first (lowest-indexed) occurrence of the argument from this vector.
removeElementAt(int index);	Deletes the component at the specified index.
removeAllElements();	Removes all components from this vector and sets its size to zero.
Object elementAt(int index);	Returns the component at the specified index.
Object lastElement();	Returns the last component of the vector.
Object firstElement();	Returns the first component (the item at index 0) of this vector.

LinkedList Class:

- LinkedList is available since jdk1.2V.

- It allows duplicates, null & insertion order is maintained.

- Default capacity when creating an LinkedList is 0.

- In linked list elements are stored in the form of nodes.

- Each node will have three fields, the data field contains data and the link fields contain references to previous and next nodes.

- It occupies more memory than ArrayList and Construction time is also high. **Syntax**: **LinkedList ll=new LinkedList();**

link	data	link
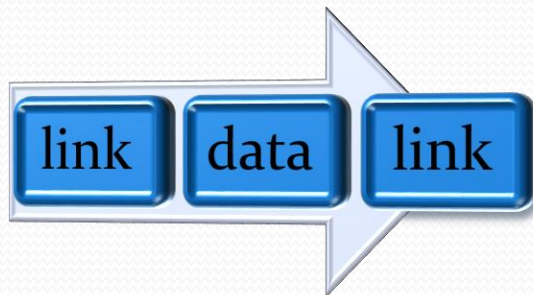
Node Structure

node	node	node
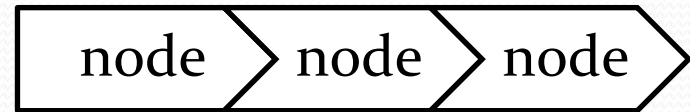
Linked List

LinkedList Methods

Methods	Description
Object getFirst();	Returns the first element in this list
Object getLast();	Returns the last element in this list
Object removeFirst();	Removes and returns the first element from this list
Object removeLast();	Removes and returns the last element from this list.
void addFirst(Element e);	Inserts the specified element at the beginning of this list.
addLast(Element e);	Appends the specified element to the end of this list.