# Linked List Implementation: Concept Challenge



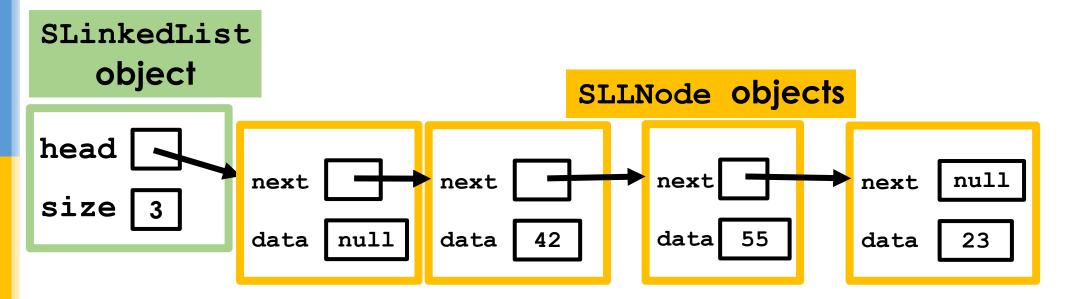
This work is licensed under a <u>Creative Commons</u>
<u>Attribution-ShareAlike 4.0 International License</u>
by Christine Alvarado, Mia Minnes, and Leo Porter, 2015.

### Concept Challenge: Procedure

- Pause Try to solve the problem yourself
- Discuss with other learners (if you can)
- Watch the UC San Diego learners video
- Answer the question again
- Confirm your understanding with our explanation



### Implementing a Singly Linked List in Java



### Implementing a Singly Linked List in Java



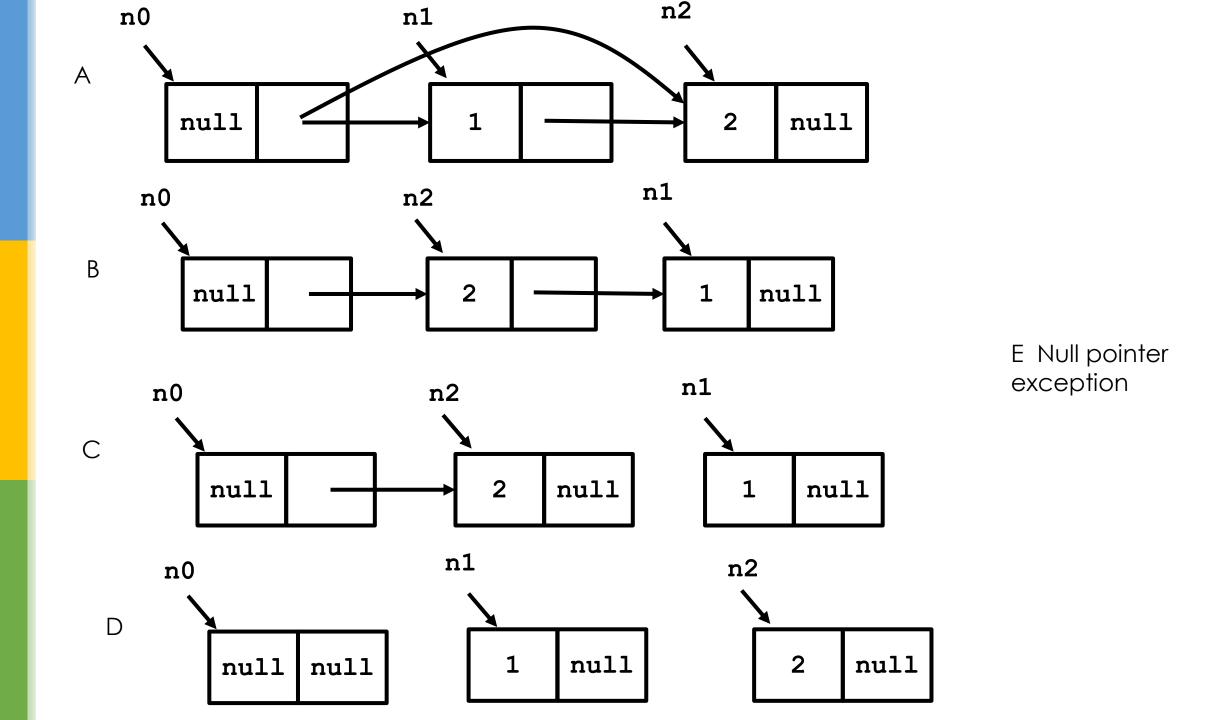
```
class SLLNode < E >
  SLLNous E mext;
 E data;
 public SLLNode(E theData) {
    this.data = theData;
 public SLLNode(E theData,
                 SLLNode<E> prevNode) {
    this (theData);
    this.next = prevNode.next;
    prevNode.next = this;
```

```
public static void main(String[] args)
{
    SLLNode<Integer> n0 =
        new SLLNode<Integer>();
    SLLNode<Integer> n1=
        new SLLNode(1,n0);
    SLLNode<Integer> n2 =
        new SLLNode(2,n0);
}
```

```
class SLLNode<E> {
  SLLNode<E> next;
 E data;
 public SLLNode() {
    THIS . ......
    this.data = null;
 public SLLNode(E theData)
    this. data the Data,
  public SLLNode (E theData,
                 SLLNode<E> prevNode)
    this (the Data):
    this.next = prevNode.next;
    prevNode.next = this;
```

```
public static void main(String[] args)
{
    SLLNode<Integer> n0 =
        new SLLNode<Integer>();
    SLLNode<Integer> n1=
        new SLLNode(1,n0);
    SLLNode<Integer> n2 =
        new SLLNode(2,n0);
}
```

```
public static void main(String[] args)
class SLLNode<E> {
                                     SLLNode<Integer> n0 =
  SLLNode<E> next;
                                        new SLLNode<Integer>();
 E data;
                                      SLLNode<Integer> n1=
                                        new SLLNode(1,n0);
 public SLLNode() {
                                      SLLNode<Integer> n2 =
    this.next = null;
                                       new SLLNode(2,n0);
    this.data = null;
 public SLLNode(E theData) {
    this.data = theData;
                             What does the list of nodes look like at the end of main?
  public SLLNode(E theData,
                  SLLNode<E> prevNode) {
    this (theData);
    this.next = prevNode.next;
    prevNode.next = this;
```



## Learner video here

```
public static void main(String[] args)
class SLLNode<E> {
                                     SLLNode<Integer> n0 =
  SLLNode<E> next;
                                       new SLLNode<Integer>();
 E data;
               Let's draw the diagram one line at a time
                                                  ger> n1=
                                     SLLN
                                                   (1,n0);
 public SLLNode() {
                                                   r> n2 =
    this.next = null;
    this.data = null
  public SLLNd
    this.data
  public SLLNode(E theData,
                 SLLNode<E> prevNode) {
    this (theData);
    this.next = prevNode.next;
    prevNode.next = this;
```

SLLNode<Integer> n0 = new SLLNode<Integer>();

n0

```
SLLNode<Integer> n0 = new SLLNode<Integer>();
```

```
n0
```

```
No-arg constructor

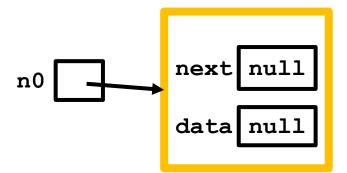
public SLLNode() {

  this.next = null;

  this.data = null;

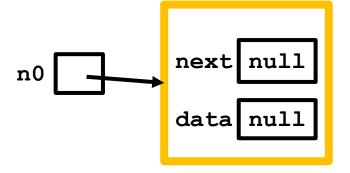
}
```

#### SLLNode<Integer> n0 = new SLLNode<Integer>();

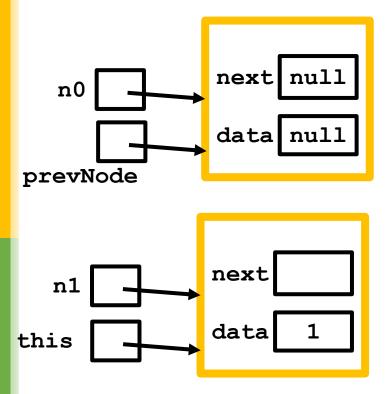


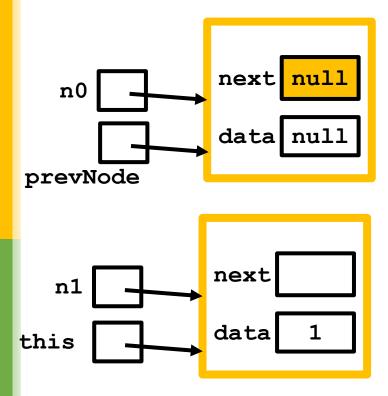
#### No-arg constructor

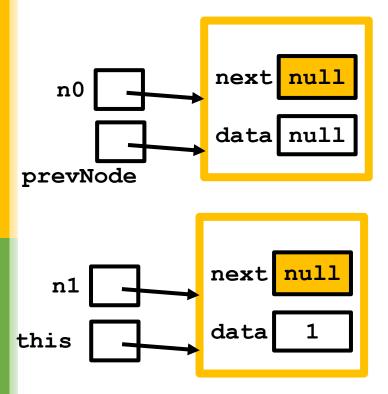
```
public SLLNode() {
  this.next = null;
  this.data = null;
}
```

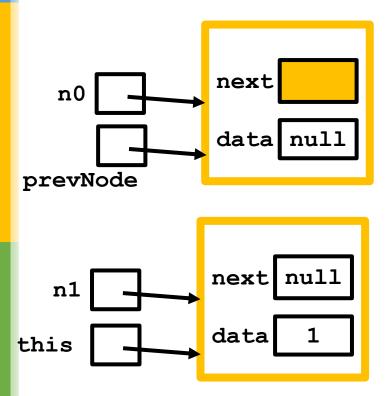


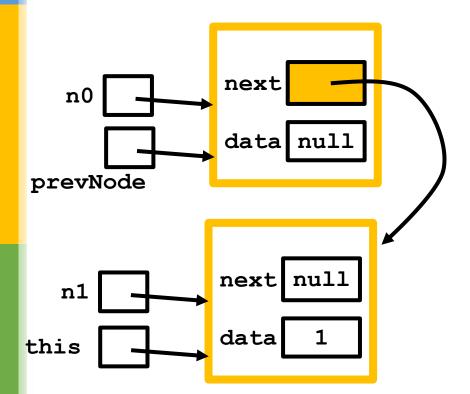
```
n1
```

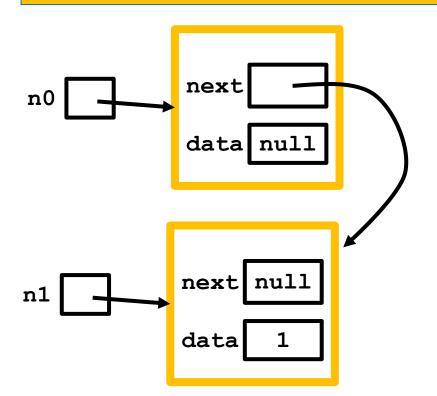


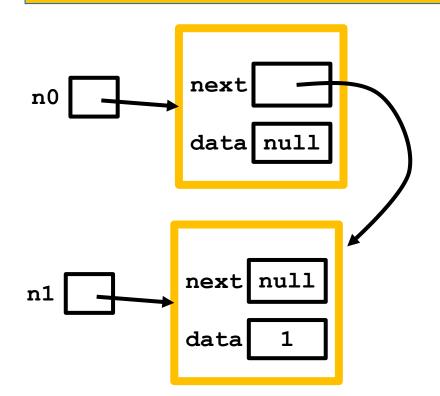












```
n2
```

