

Linked List Implementation: Concept Challenge 2



This work is licensed under a [Creative Commons Attribution-ShareAlike 4.0 International License](https://creativecommons.org/licenses/by-sa/4.0/)
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

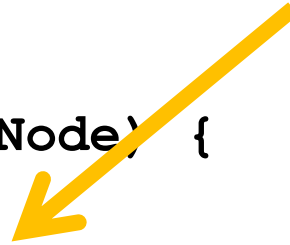


```
class SLLNode<E> {
    SLLNode<E> next;
    E data;

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

What about if we swap these two lines ?



```

class SLLNode<E> {
    SLLNode<E> next;
    E data;

    public SLLNode() {
        this.next = null;
        this.data = null;
    }
    public SLLNode(E theData) {
        this.data = theData;
    }
    public SLLNode(E theData,
                   SLLNode<E> prevNode) {
        this(theData);
        prevNode.next = this;
        this.next = prevNode.next;
    }
}

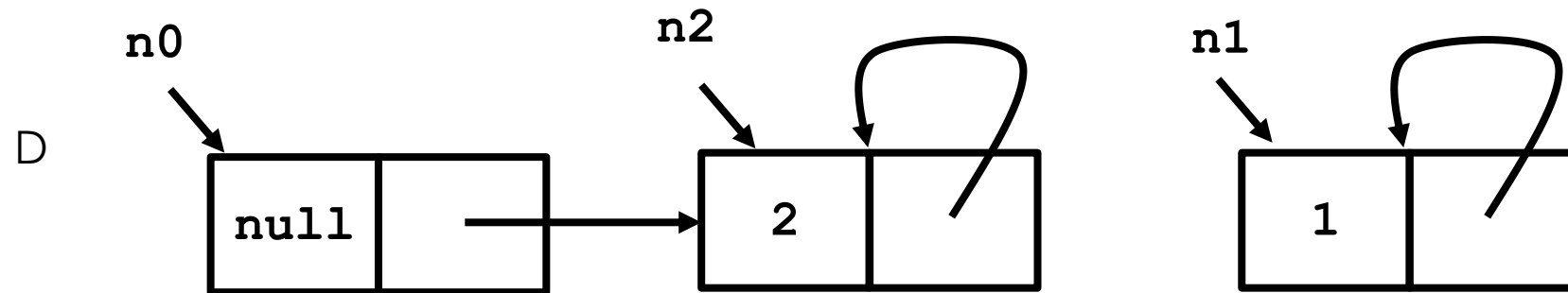
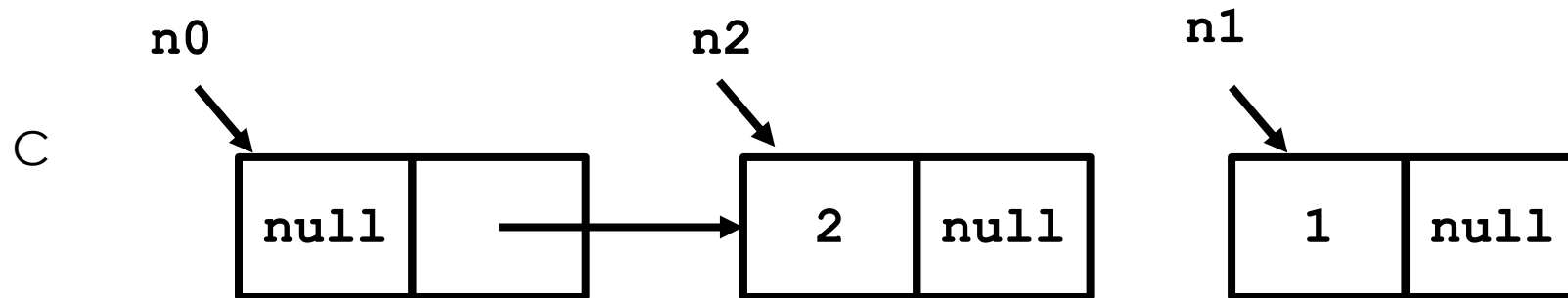
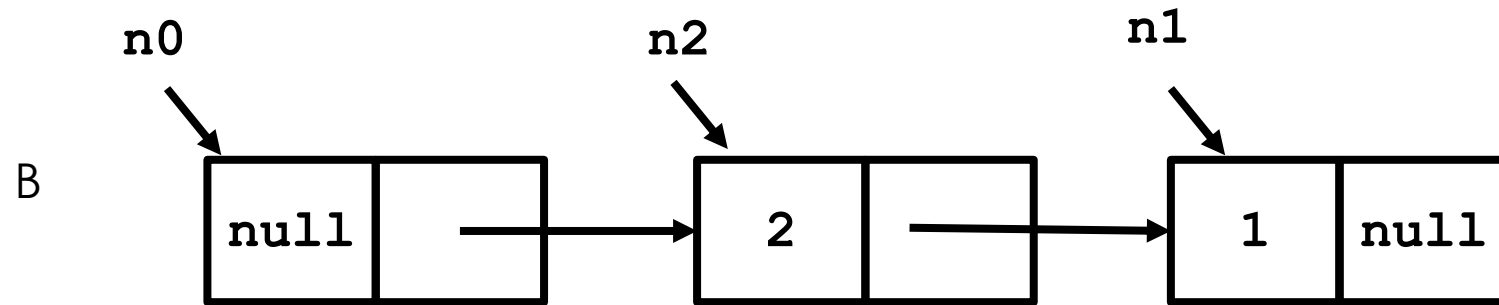
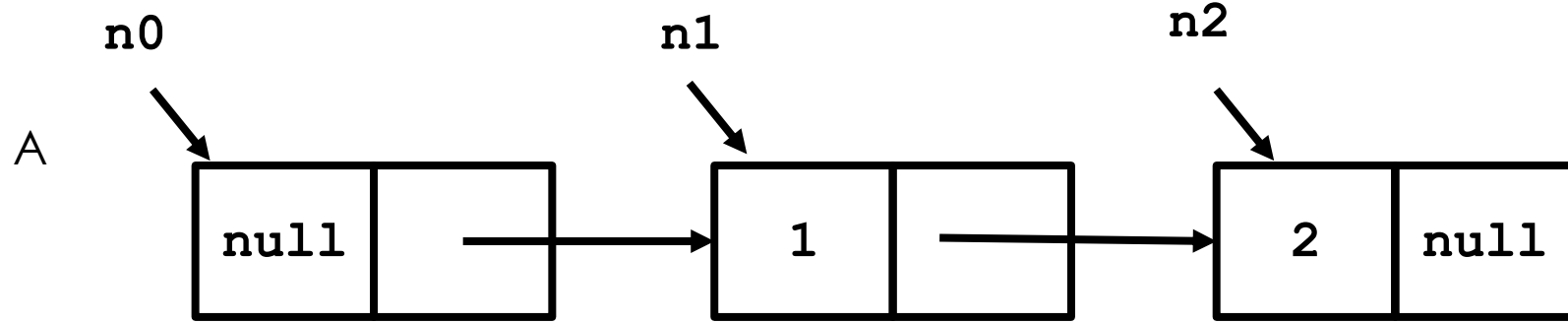
```

```

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);
}

```

What does the list of nodes look like after main runs?



E Null pointer exception

```

class SLLNode<E> {
    SLLNode<E> next;
    E data;

    public SLLNode() {
        this.next = null;
        this.data = null;
    }
    public SLLNode(E theData) {
        this.data = theData;
    }
    public SLLNode(E theData,
                   SLLNode<E> prevNode) {
        this(theData);
        this.next = prevNode.next;
        prevNode.next = this;
    }
}

```

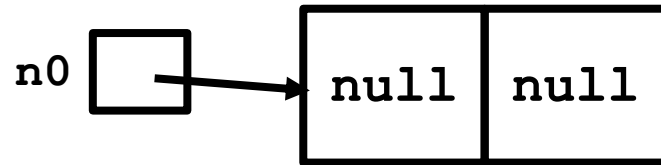
```

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);
}

```

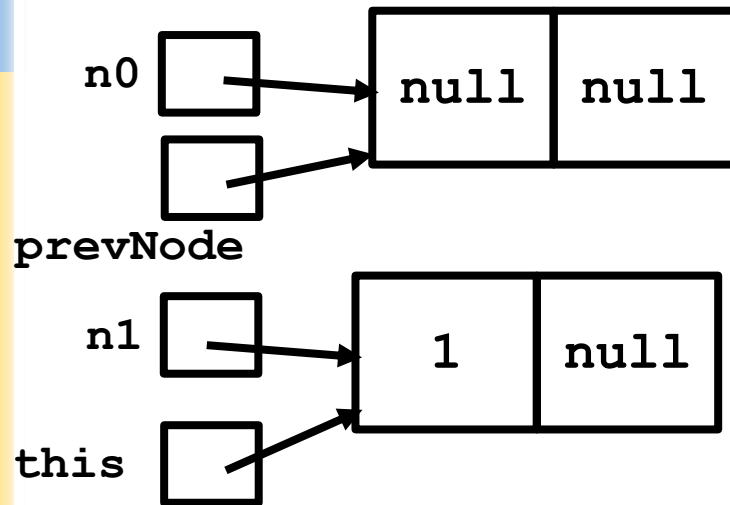
Step-by-step walkthrough ...

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



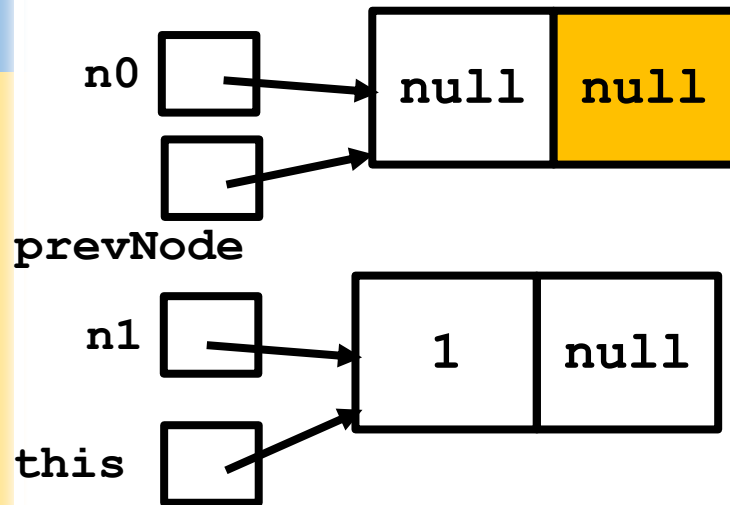
```
SLLNode<Integer> n1 = new SLLNode(1, n0);
```

```
public SLLNode(E theData,  
               SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



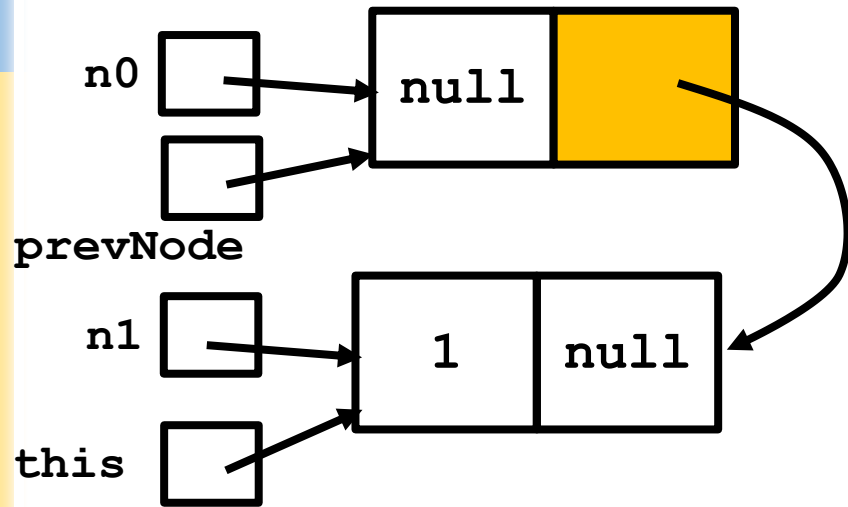

```
SLLNode<Integer> n1 = new SLLNode(1, n0);
```

```
public SLLNode(E theData,  
               SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



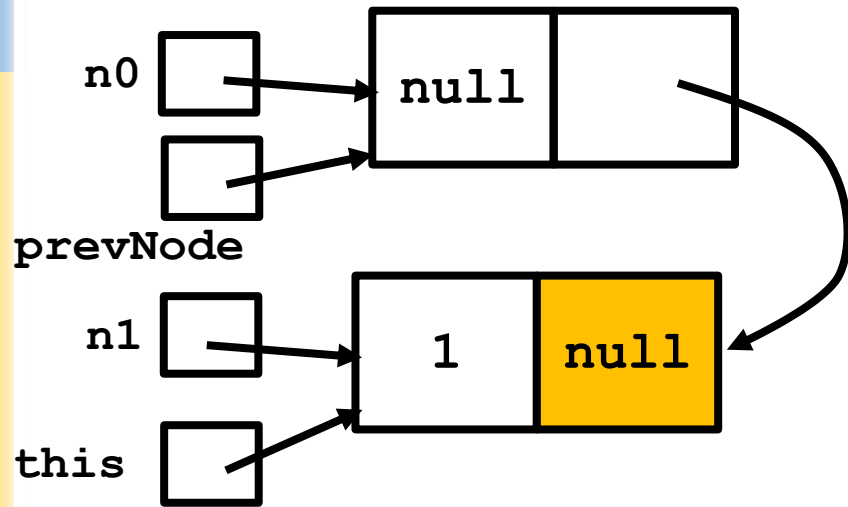
```
SLLNode<Integer> n1 = new SLLNode(1, n0);
```

```
public SLLNode(E theData,  
               SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



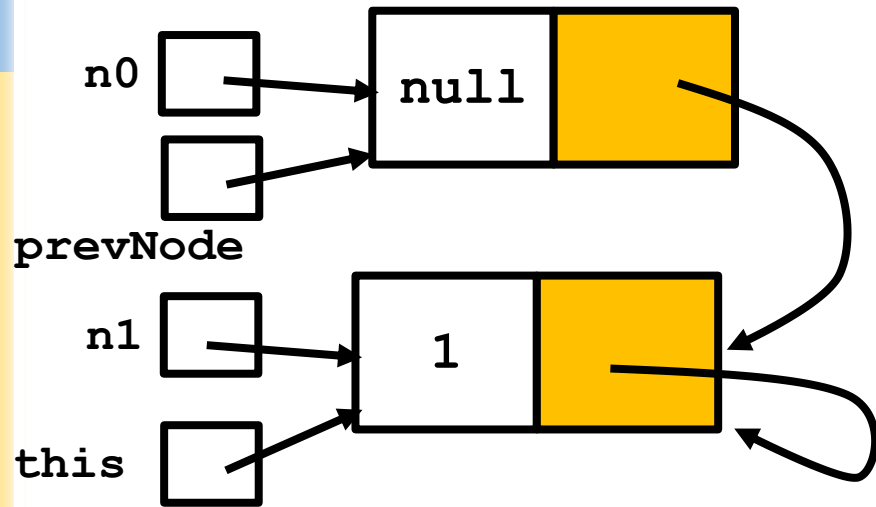
```
SLLNode<Integer> n1 = new SLLNode(1, n0);
```

```
public SLLNode(E theData,  
                SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



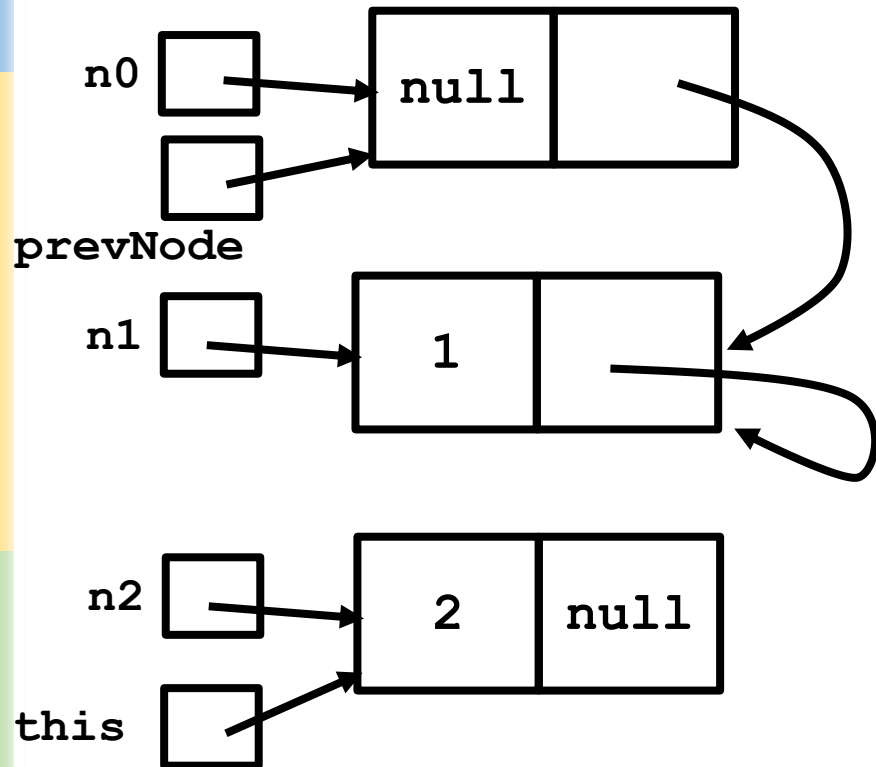
```
SLLNode<Integer> n1 = new SLLNode(1, n0);
```

```
public SLLNode(E theData,  
                SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



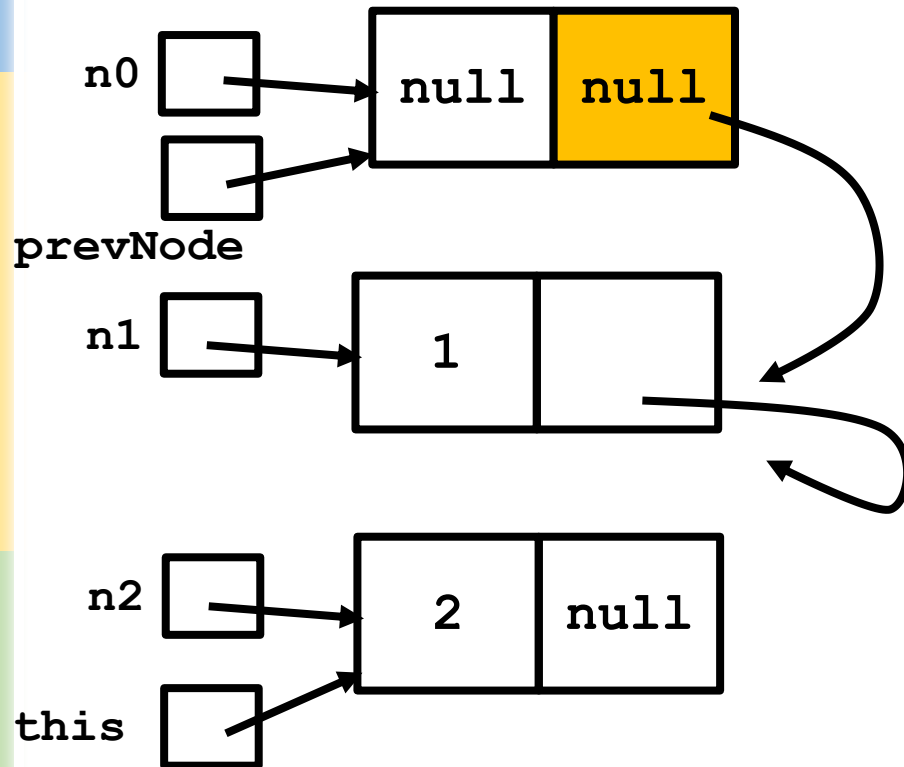
```
SLLNode<Integer> n2 = new SLLNode(2, n0);
```

```
public SLLNode(E theData,  
                SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



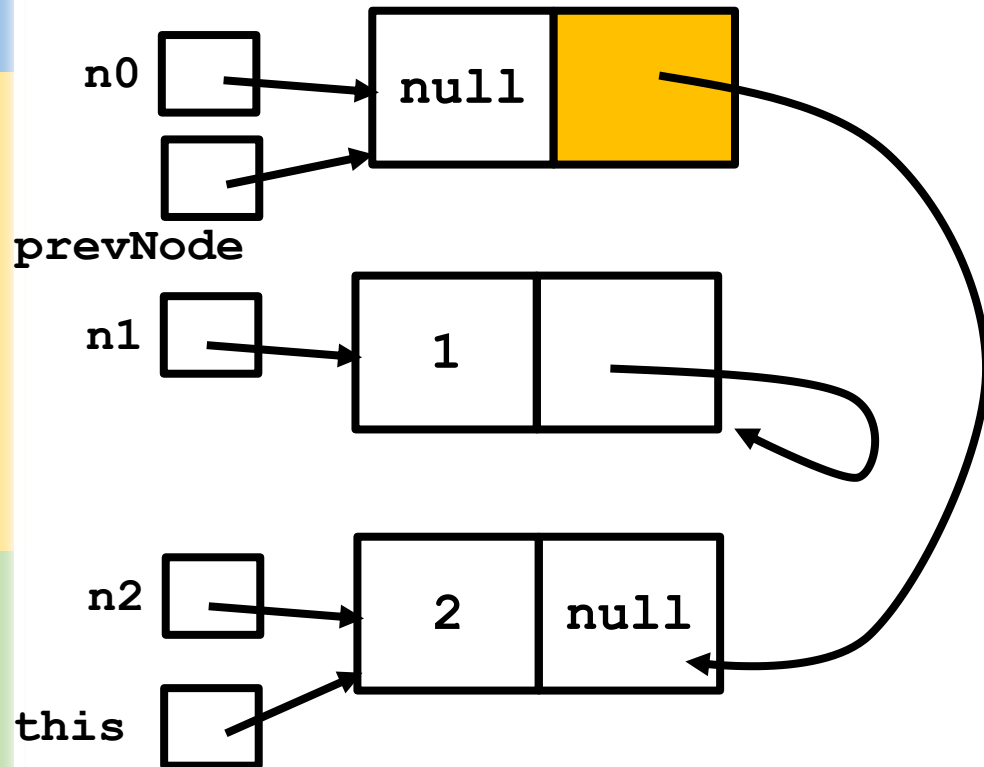
```
SLLNode<Integer> n2 = new SLLNode(2, n0);
```

```
public SLLNode(E theData,  
               SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



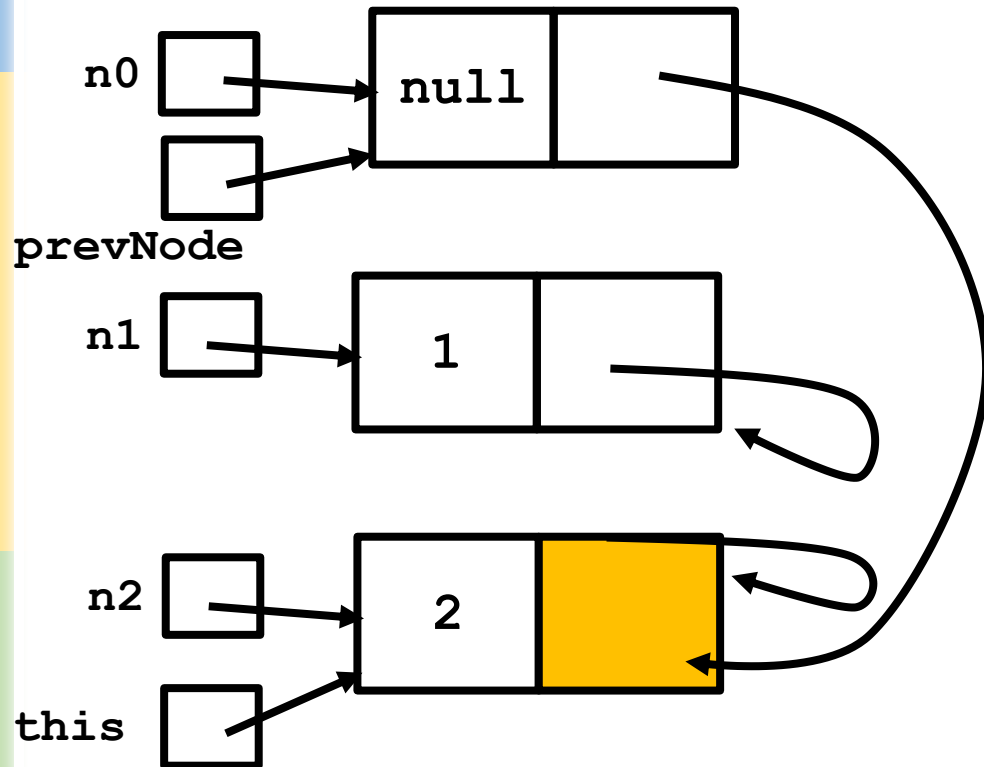
```
SLLNode<Integer> n2 = new SLLNode(2, n0);
```

```
public SLLNode(E theData,  
                SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```

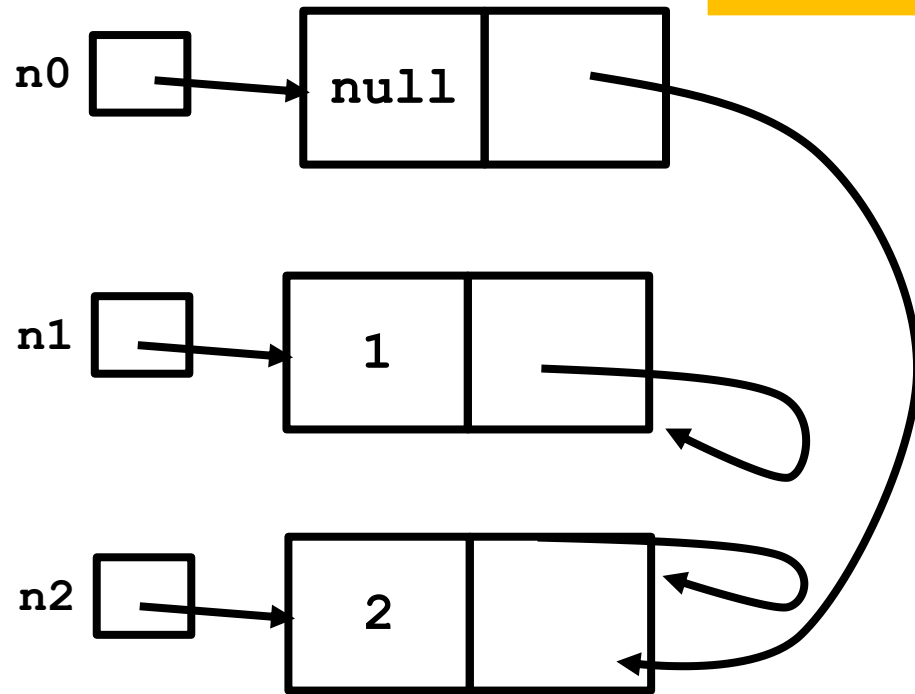


```
SLLNode<Integer> n2 = new SLLNode(2, n0);
```

```
public SLLNode(E theData,  
                SLLNode<E> prevNode) {  
    this(theData);  
    prevNode.next = this;  
    this.next = prevNode.next;  
}
```



Head of list (Sentinel)



Head of list (Sentinel)

