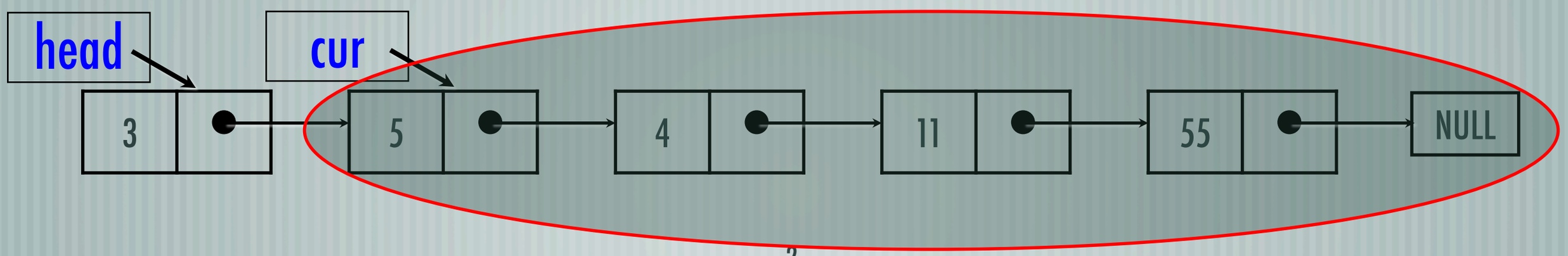
Secret Weapons About Recursion

Yun Tian (Tony)
Department of Computer Science
Eastern Washington University
CSCD300

Secret Weapons About Recursion

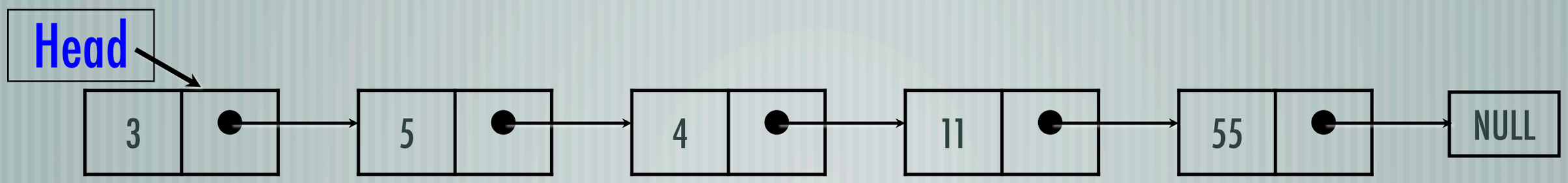
- Silver Bullets
  - You make the recursive call as if you are calling a regular method or an existing API.
  - You HAVE TO trust that the recursive call will surely do the work for you!



Print List in reverse order

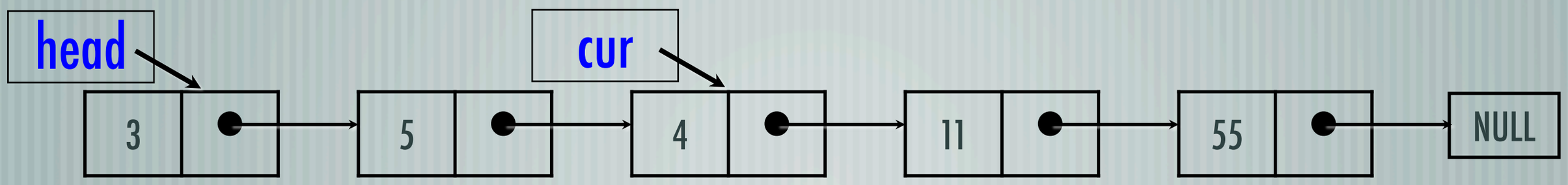
```
public class ListNode{
        private Object   data;
        private ListNode   next;

        public ListNode(Object  data, Node  next){
                this.data = data;
                this.next = next;
        }
        // other methods
}
```

Figure, ListNode class definition



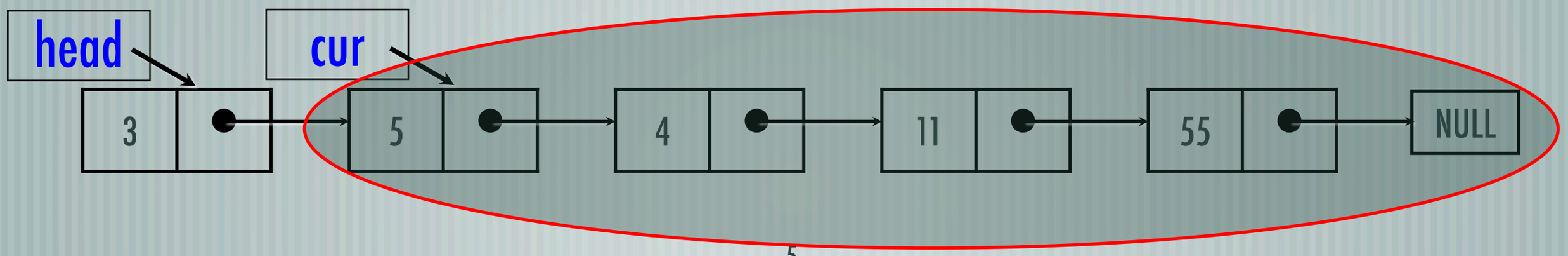
Print List in reverse order

- How to print out data in linked list in reverse order?
- Another way to think about linked list,
- A linked list headed at number 3 contains {3 ,5, 4, 11, 55}.
  - With reference *Head pointing to this list.*
- A sublinked list headed at number 4 contains {4, 11, 55}.
  - With reference *Cur pointing to this sublist.*
- *The head node uniquely determines a linked list (or sublist).*



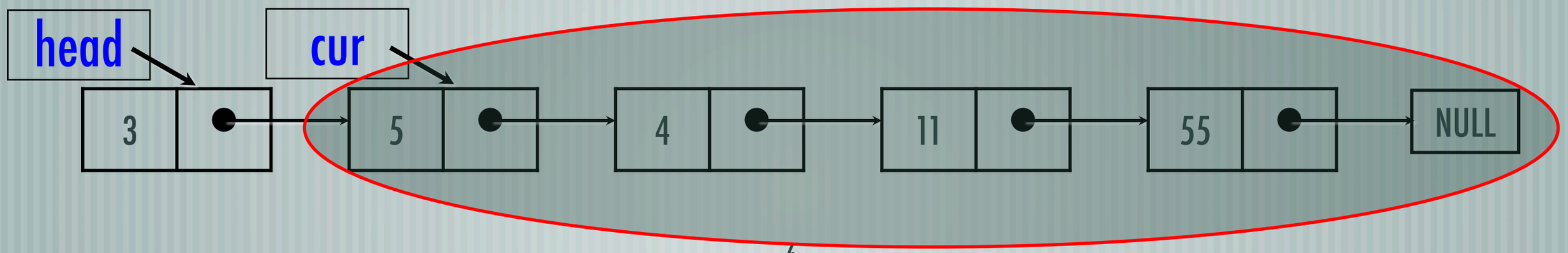
Print List in reverse order

- We first design our function interface
- void printReverse(Node head)
  - Summarize the functionality of the method,
  - printReverse(Node head) will print out all nodes in reverse order in the linked list headed at **head**,
- Base Cases
  - if size of linked list is zero. i.e. head is null(without dummy).
- Recursive Pattern(Definition)
  - first print out in reverse order the sublist that is headed at head.next, then print out the data in head node of this list.



Print List in reverse order

- Silver Bullets
  - You make the recursive call as if you are calling a regular method.
  - You HAVE TO trust that the recursive call will surely do the work for you!



Print Data in LinkedList in Reverse Order

```
// Singly Linked List without Dummny Node
private void printReverse(Node head) {
        if(head == null)
                return;                         //base case
        else
                printReverse(head.next);        //recursive pattern

        System.out.println(head.data);

}

public void printReverse() {
        printReverse(this.head);
}
```

Print Data in LinkedList ... er

```
// Singly Linked List without Dummny Node
private void printReverse(Node head) {
    if(head == null)
            return;                          //base case
    else
            printReverse(head.next);    //recursive pattern

    System.out.println(head.data);

}

public void printReverse() {
    printReverse(this.head);
}
```