

Introduction to Computer Science II

CS112-2008S-19

Linked Lists II

David Galles

Department of Computer Science
University of San Francisco

19-0: Midterm on Wednesday

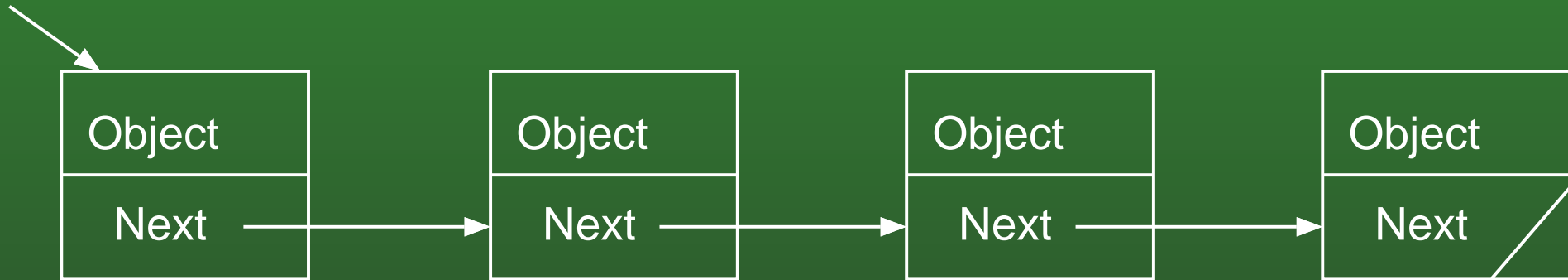
- Trace through code
 - What does this code output?
 - Basic constructs (loops / function calls / arrays)
 - Polymorphism
 - Exceptions
 - Recursion
 - (all code tested on a compiler!)

19-1: Midterm on Wednesday

- Write code
 - Basic constructs (loops / function calls)
 - Arrays (1D / 2D)
 - Simple Recursion

19-2: Linked Lists

head



19-3: **Linked Lists**

- Review:
 - Finding the nth element in a list
 - Finding an element in a list
 - `boolean find(ListNode list, Object o)`

19-4: Linked Lists

```
public boolean find(ListNode list, Object o)
{
    while (list != null)
    {
        if (list.data.equals(o))
        {
            return true;
        }
        list = list.next;
    }
    return false;
}
```

- Recursive?

19-5: Linked Lists

```
public boolean find(ListNode list, Object o)
{
    if (list == null)
        return false;
    if (list.data.equals(o))
        return true;
    return find(list.next, o);
}
```

19-6: Linked Lists

```
public class LinkedList
{
    private ListNode head;

    public LinkedList()
    {
        head = null;
    }

    boolean find(Object o) { ... }
}
```

19-7: Linked Lists

```
public class LinkedList
{
    private ListNode head;

    public LinkedList()
    {
        head = null;
    }

    boolean find(Object o)
    {
        ListNode tmp = head;
        while (tmp != null)
        {
            if (tmp.data.equals(o))
                return true;
            tmp = tmp.next;
        }
        return false;
    }
}
```

19-8: Linked Lists

```
public class LinkedList
{
    private ListNode head;

    public LinkedList()
    {
        head = null;
    }

    boolean find(Object o)
    {
        return find(head, o); // Find as defined above
    }
}
```

19-9: Linked Lists

- Linked List Methods
 - `insertAt(Object o, int index)` Insert an element at a specific index
 - `removeAt(int index)` Remove an element at a specified index
 - `remove(Object o)` Remove an element `o` from the list, if it exists.

19-10: Linked Lists

- `void insertAt(Object o)`

```
public class LinkedList
{
    private ListNode head;

    public LinkedList()
    {
        head = null;
    }

    public void insertAt(Object o, int index) { ... }
}
```

19-11: Linked Lists

```
public class LinkedList
{
    private ListNode head;
    public void insertAt(Object o, int index)
    {
        if (index == 0)
        {
            head = new ListNode(o, head);
        }
        else
        {
            ListNode tmp = head;
            for (int i = 0; i < index - 1; i++)
            {
                tmp = tmp.next;
            }
            ListNode newElem = new ListNode(o);
            newElem.next = tmp.next;
            tmp.next = newElem;
        }
    }
}
```

19-12: Linked Lists

```
public class LinkedList
{
    private ListNode head;
    public void insertAt(Object o, int index)
    {
        if (index == 0)
        {
            head = new ListNode(o, head);
        }
        else
        {
            ListNode tmp = head;
            for (int i = 0; i < index - 1; i++)
            {
                tmp = tmp.next;
            }
            tmp.next = new ListNode(o, tmp.next);
        }
    }
}
```

19-13: Linked Lists

- Linked List Methods
 - `removeAt(int index)` Remove an element at a specified index
 - Much like `insertAt`
 - Need a special case for deleting the first element in the list
 - (why?)
 - See code on Friday, create code today!

19-14: Linked Lists

- Linked List Methods
 - `remove(Object o)` Remove the first occurrence of an object from a list
 - If we advance a pointer until we find the element to remove, we've gone too far
 - Still need a special case for deleting the first element in the list

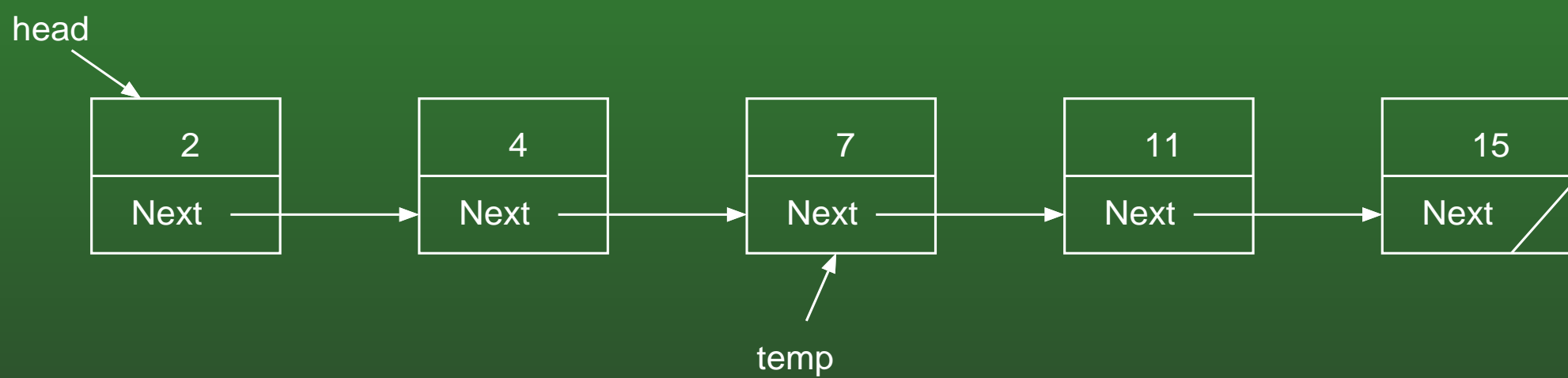
19-15: Linked Lists

Deleting the 7 from the list



19-16: Linked Lists

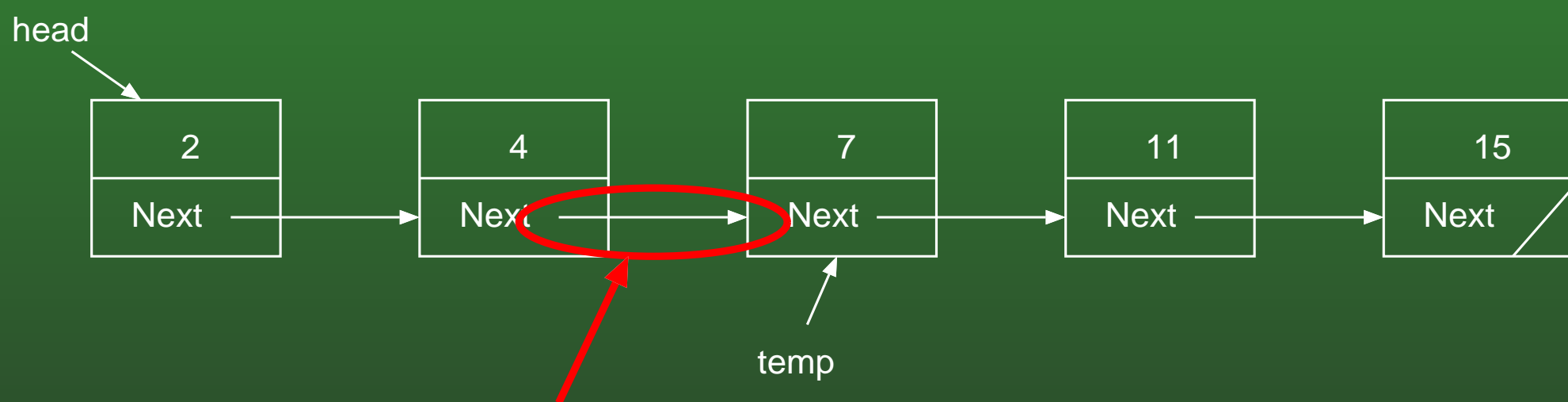
Deleting the 7 from the list



Advance a temp pointer until it points to the element to be deleted

19-17: Linked Lists

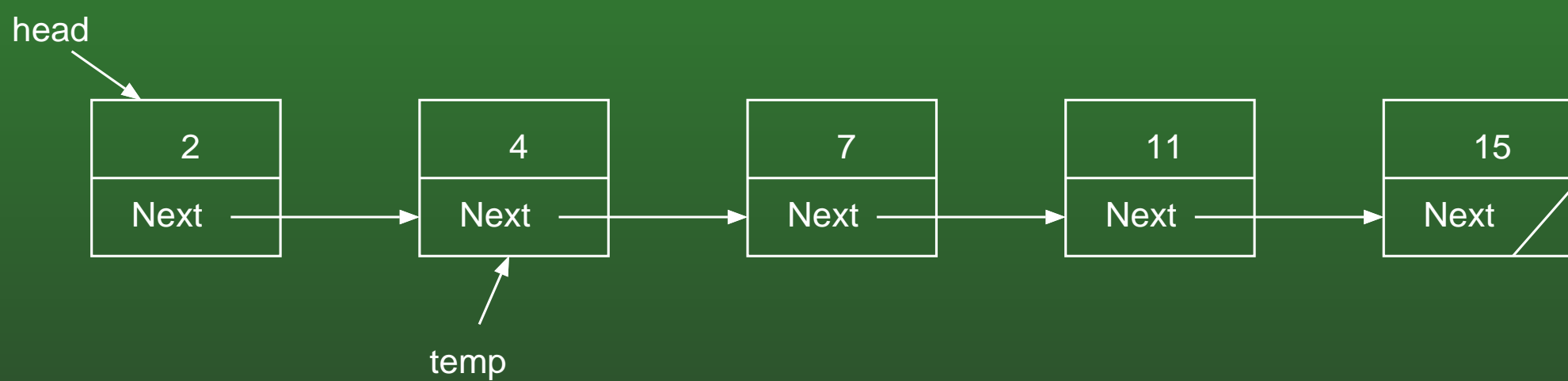
Deleting the 7 from the list



Need to change this pointer to remove the 7,
no way to get to it from temp

19-18: Linked Lists

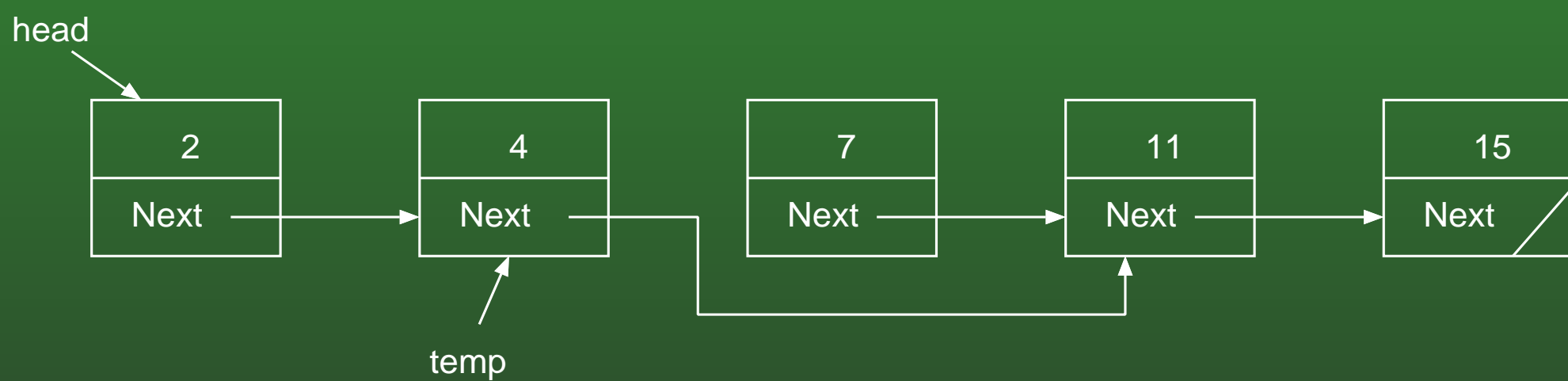
Deleting the 7 from the list



If we stop one step before we get to the 7,
we can remove the 7 from the list

19-19: Linked Lists

Deleting the 7 from the list



If we stop one step before we get to the 7,
we can remove the 7 from the list

19-20: Practical Experience

- Get ListNode, LinkedList classes from website
- Fill in missing classes
- Write a main program that prompts the user to enter strings. Each string is inserted into the list, until an empty string is added – at which point the program prints out the list using the toString method