CMSC202 Computer Science II for Majors

Lecture 12 – Linked Lists

Dr. Katherine Gibson

Last Class We Covered

- Inheritance
- Object relationships
  - is-a (Inheritance)
  - has-a (Composition and Aggregation)

Any Questions from Last Time?

Today’s Objectives

- To cover linked lists in detail
  - Traversal
  - Creation
  - Insertion
  - Deletion

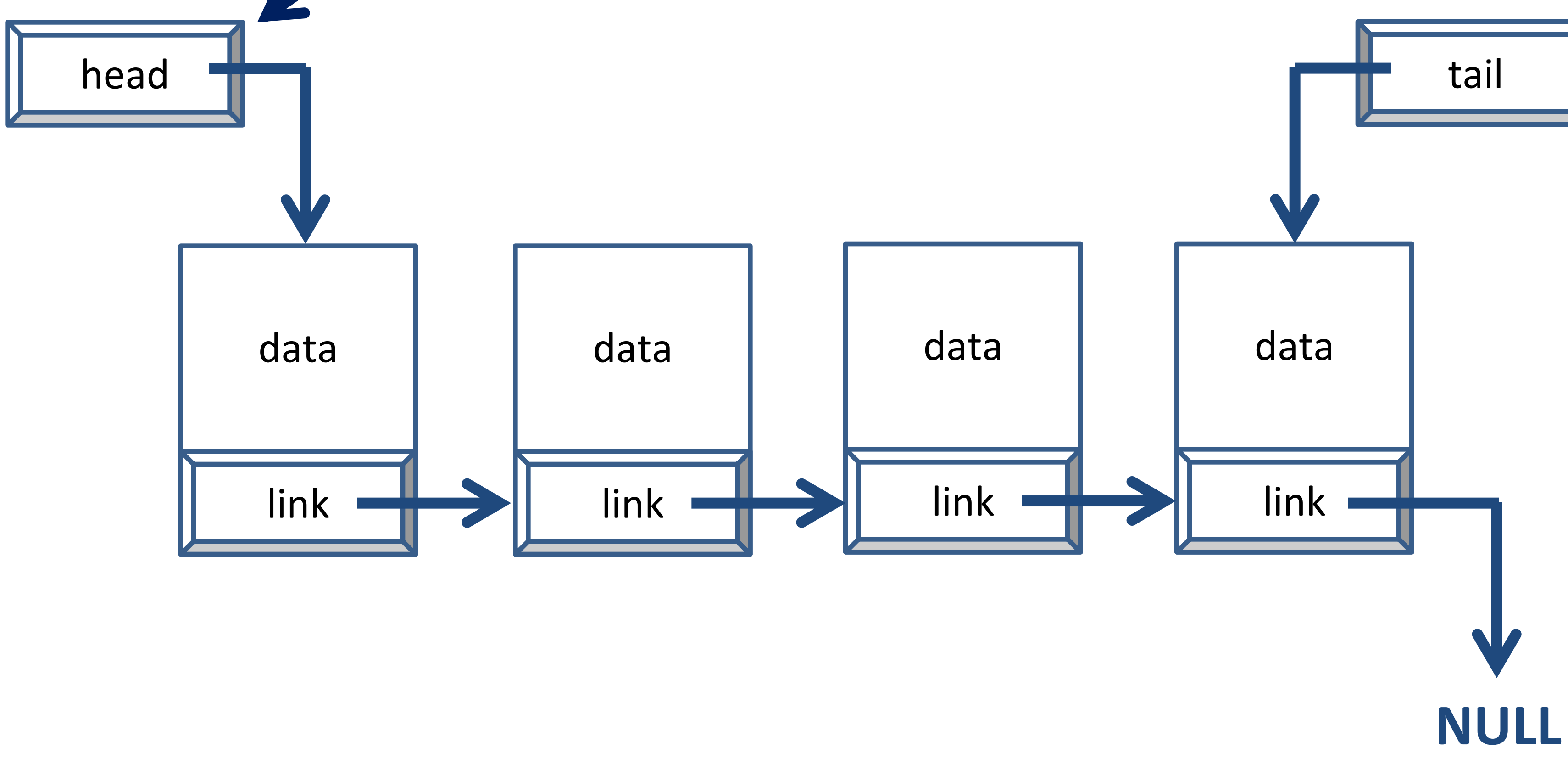
Linked Lists vs Vectors

What is a Linked List?

- Data structure
  - Dynamic
  - Allow easy insertion and deletion
- Uses nodes that contain
  - Data
  - Pointer to next node in the list

Example Linked List

In these diagrams, a doubly-outlined box indicates a pointer.

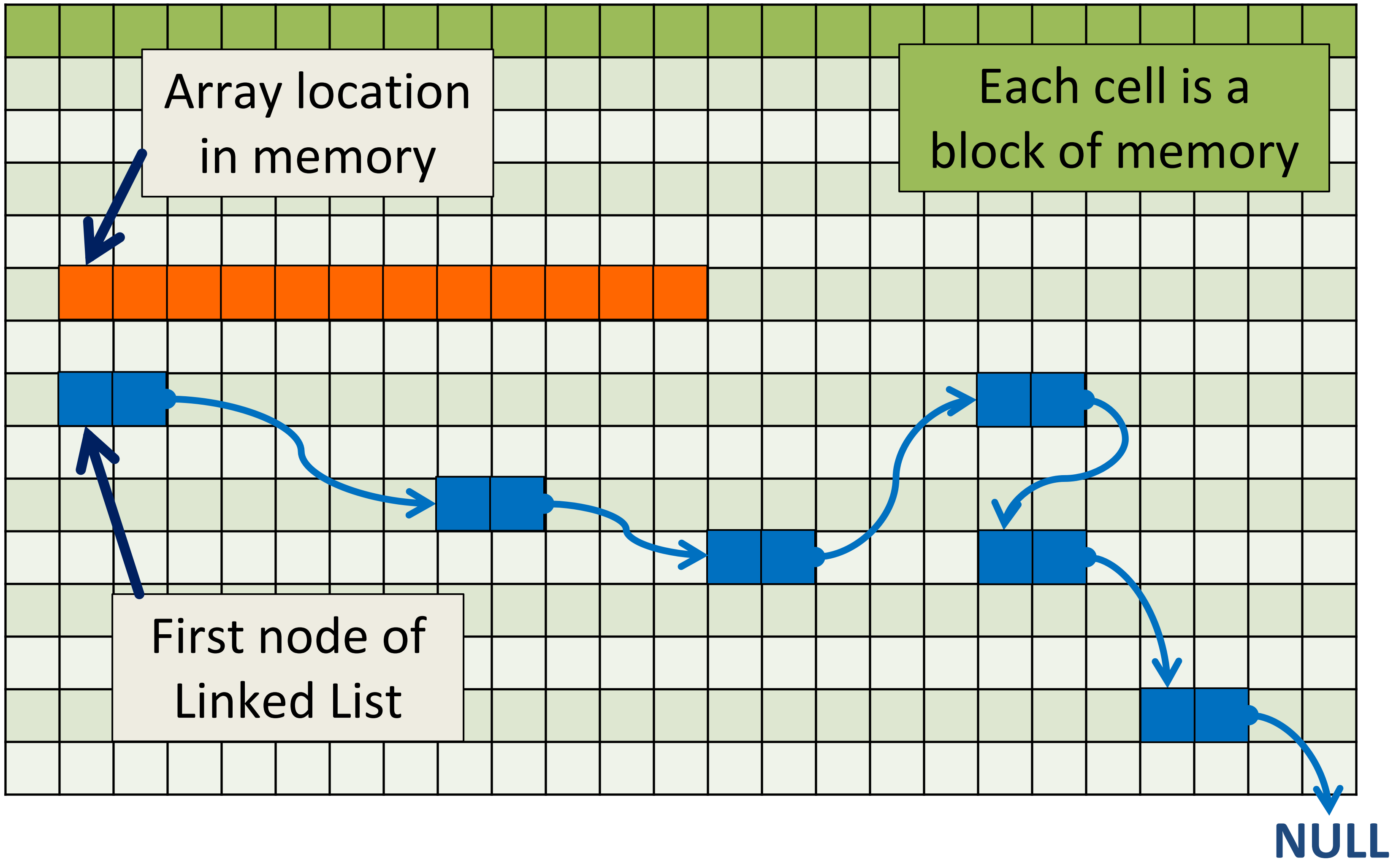


Why Use Linked Lists?

- We already have vectors!
- What are some disadvantages of an vectors?
  - Inserting in the middle of an array takes time
  - Deletion as well
  - Sorting
  - Requires a *contiguous* block of memory

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Representation in Memory



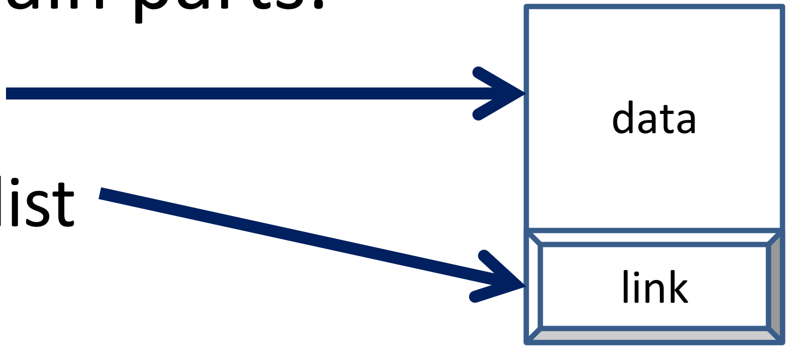
(Dis)Advantages of Linked Lists

- Advantages:
  - Change size easily and constantly
  - Insertion and deletion can easily happen anywhere in the Linked List
  - Only one node needs to be contiguously stored
- Disadvantages:
  - Can’t access by index value
  - Requires management of memory
  - Pointer to next node takes up more memory

Nodes

Nodes

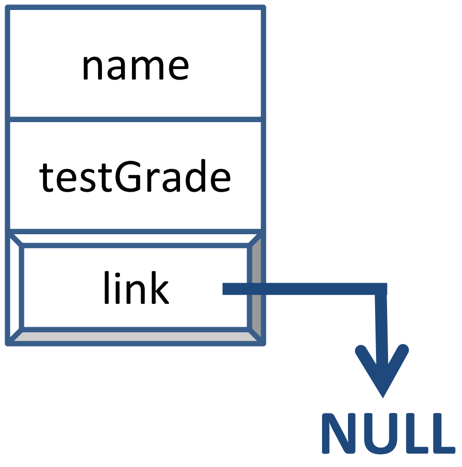
- A node is one element of a Linked List
- Nodes consist of two main parts:
  - Data stored in the node
  - Pointer to next node in list
- Often represented as classes



Code for Node Class

```
class Node
{
    String name;
    int    testGrade;
    Node   *link;

    // constructor
    // accessors
    // mutators
};
```



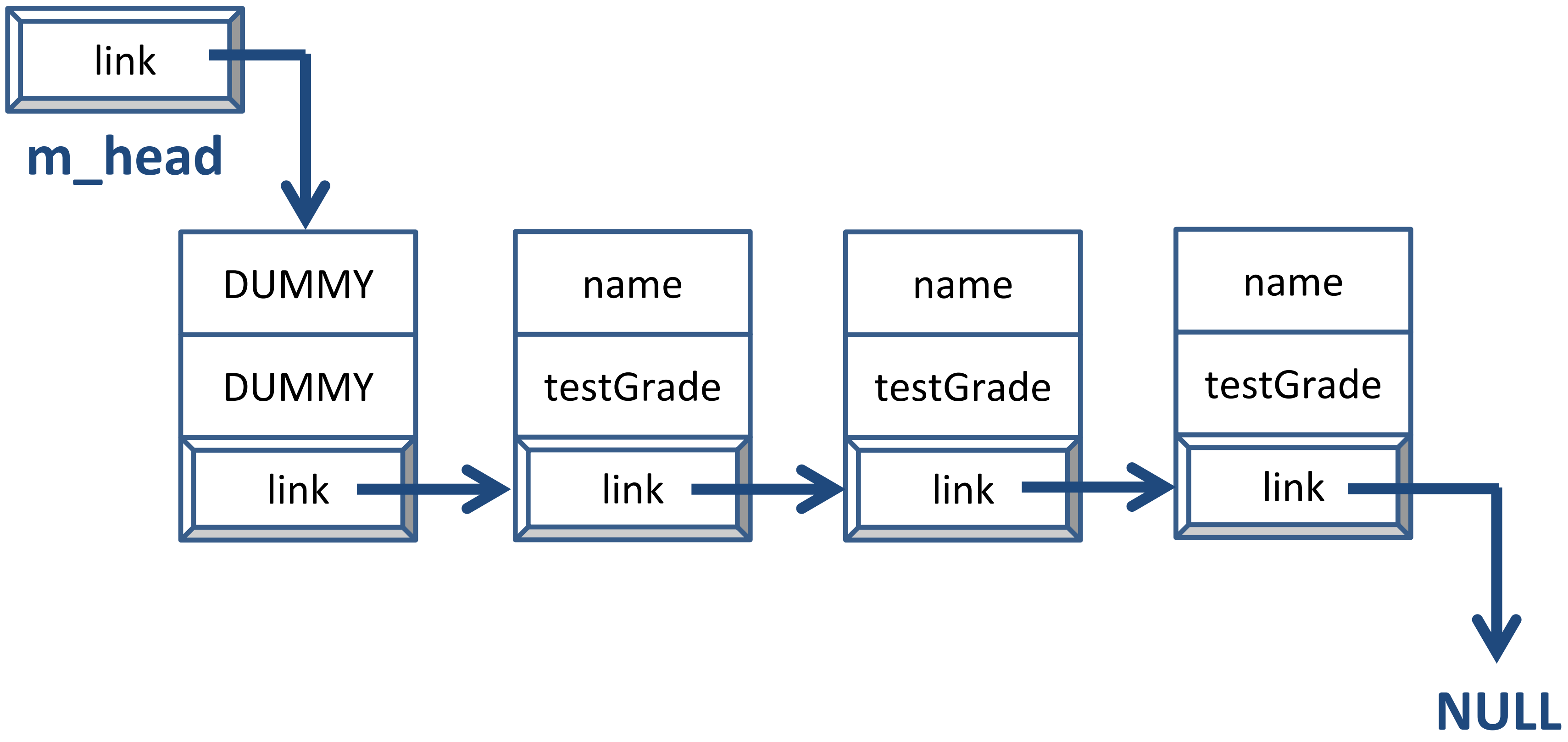
link can point to other nodes

two options:
1. another Node
2. NULL

Linked List Overview

UMBC
AN HONORS UNIVERSITY IN MARYLAND

Example Linked List



Important Points to Remember

- Last node in the Linked List points to **`NULL`**
- Each node points to either another node in the Linked List, or to **`NULL`**
  - Only one link per node

Managing Memory with LLs

- Hard part of using Linked Lists is ensuring that none of the nodes go “missing”

- Think of Linked List as a train
  - (Or as a conga line of Kindergarteners)

- Must keep track of where links point to
- If you’re not careful, nodes can get lost in memory (and you have no way to find them)

UMBC

AN HONORS UNIVERSITY IN MARYLAND

Linked List Functions

- What functions does a Linked List class implementation require?

- Linked_List constructor
- `insert()`
- `remove()`
- `printList()`
- `isEmpty()`

Linked Lists' "Special" Cases

- Linked Lists often need to be handled differently under specific circumstances
  - Linked List is empty
  - Linked List has only one element
  - Linked List has multiple elements
  - Changing something with the first or last node
- Keep this in mind when you are coding
  - Dummy nodes alleviate some of these concerns

Creating a Linked List

On the Board

Traversing the List

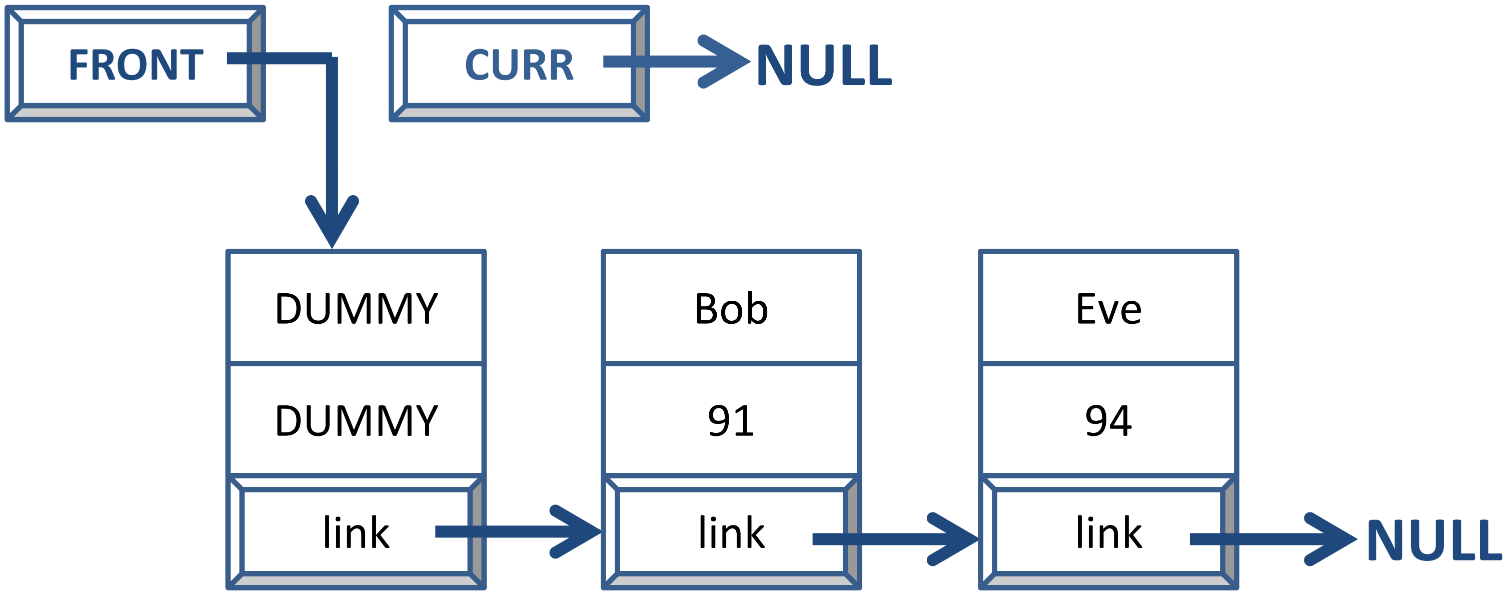
- To control our traversal, we'll use a loop
  - Initialization, Termination Condition, Modification

1. Set **`CURR`** to the first node in the list

2. Continue until we hit the end of the list (**`NULL`**)

3. Move from one node to another (using **`m_next`**)

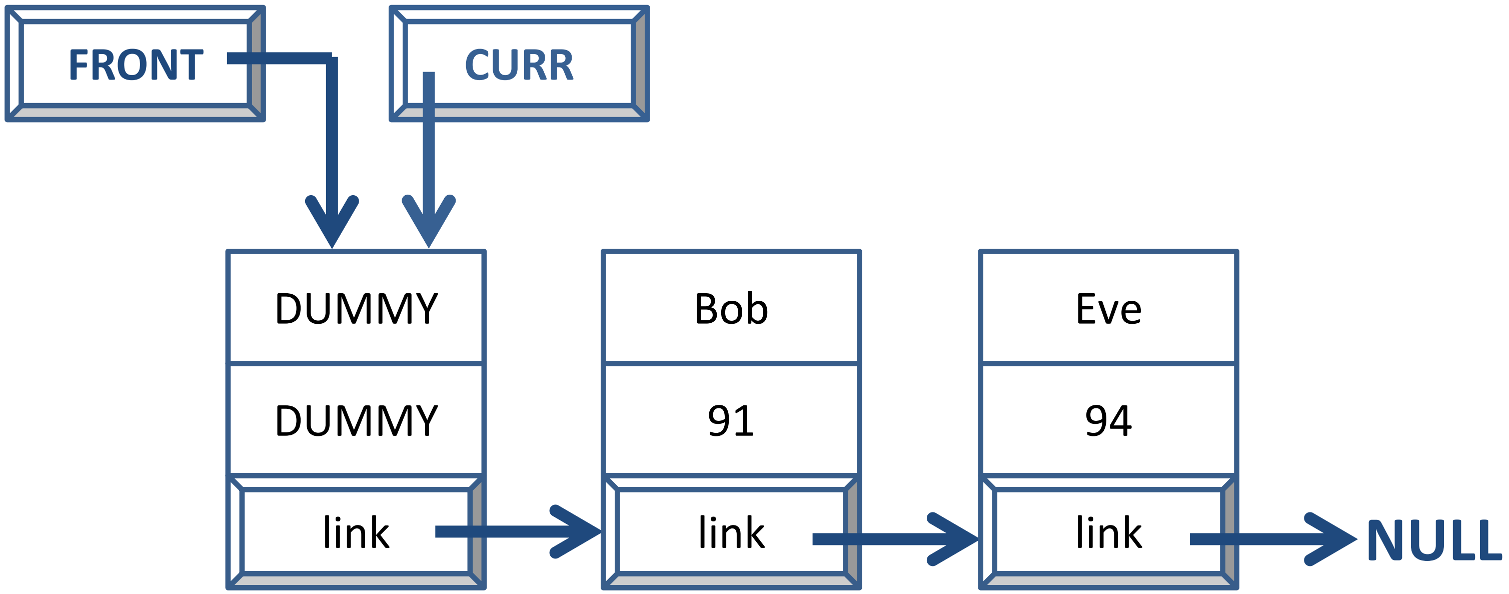
Demonstration of Traversal



```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link) {
```

UMBC AN HONORS UNIVERSITY IN MARYLAND

Demonstration of Traversal

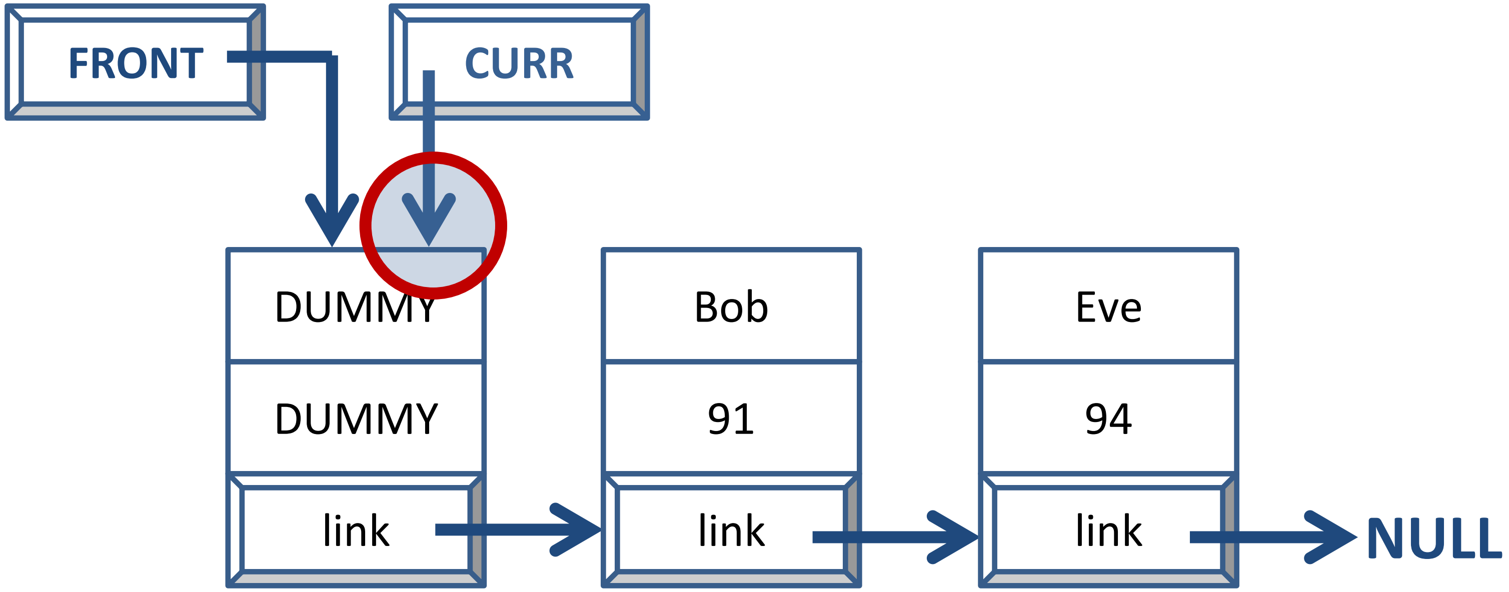


```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link) {
```

UMBC

AN HONORS UNIVERSITY IN MARYLAND

Demonstration of Traversal

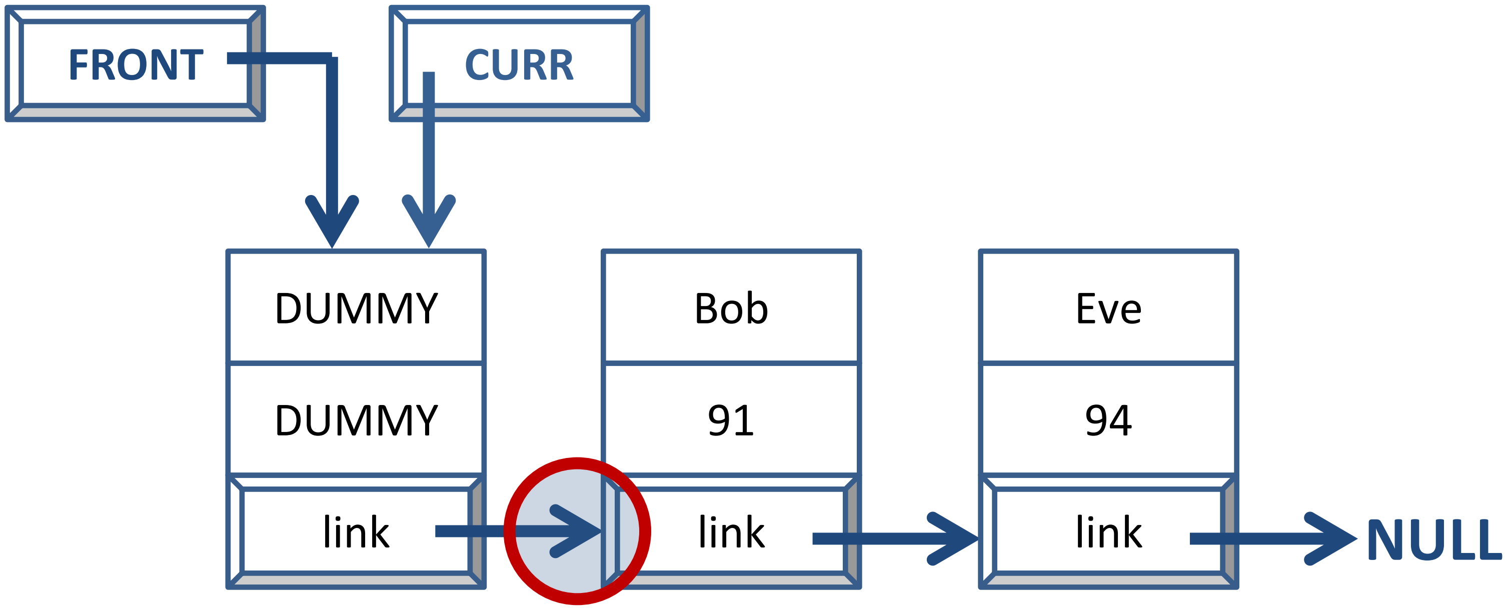


```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link) {
```

✓

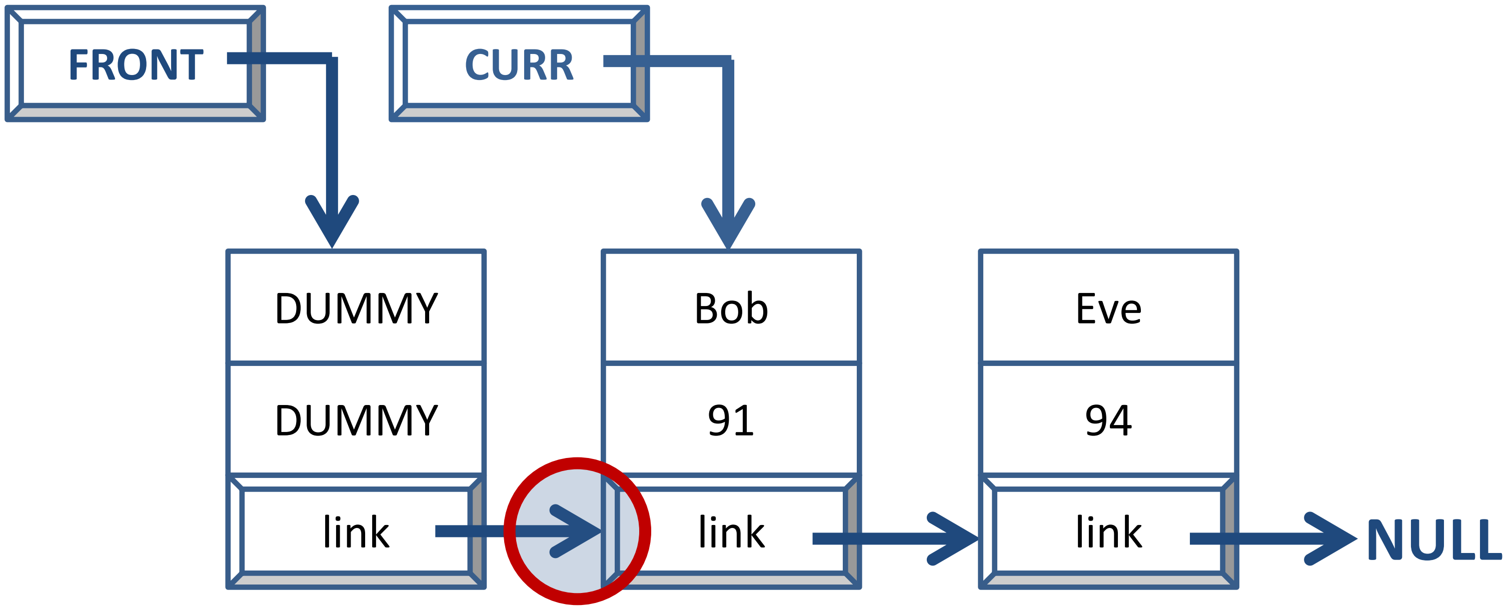
```
    // ignore, dummy node
```

Demonstration of Traversal



```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link  {
```

Demonstration of Traversal

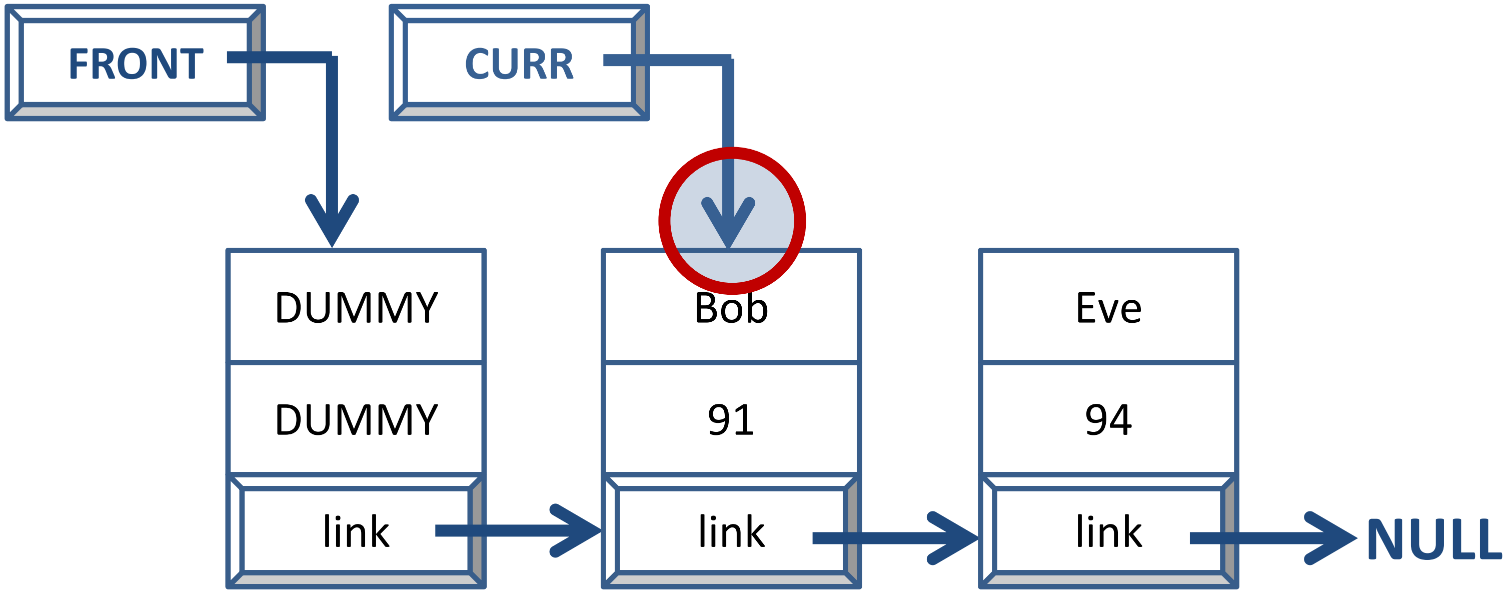


```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link {
```

UMBC

AN HONORS UNIVERSITY IN MARYLAND

Demonstration of Traversal

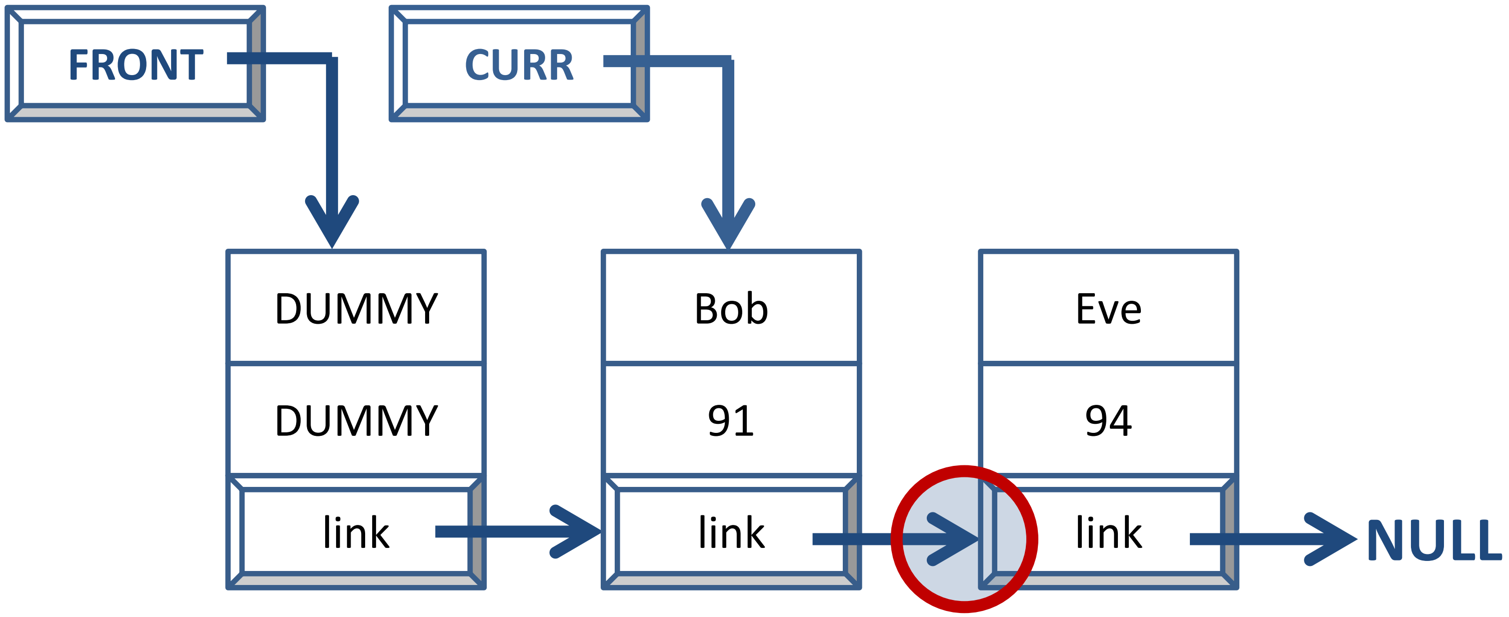


```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link) {
```

✓

```
    // print information (Bob)
```

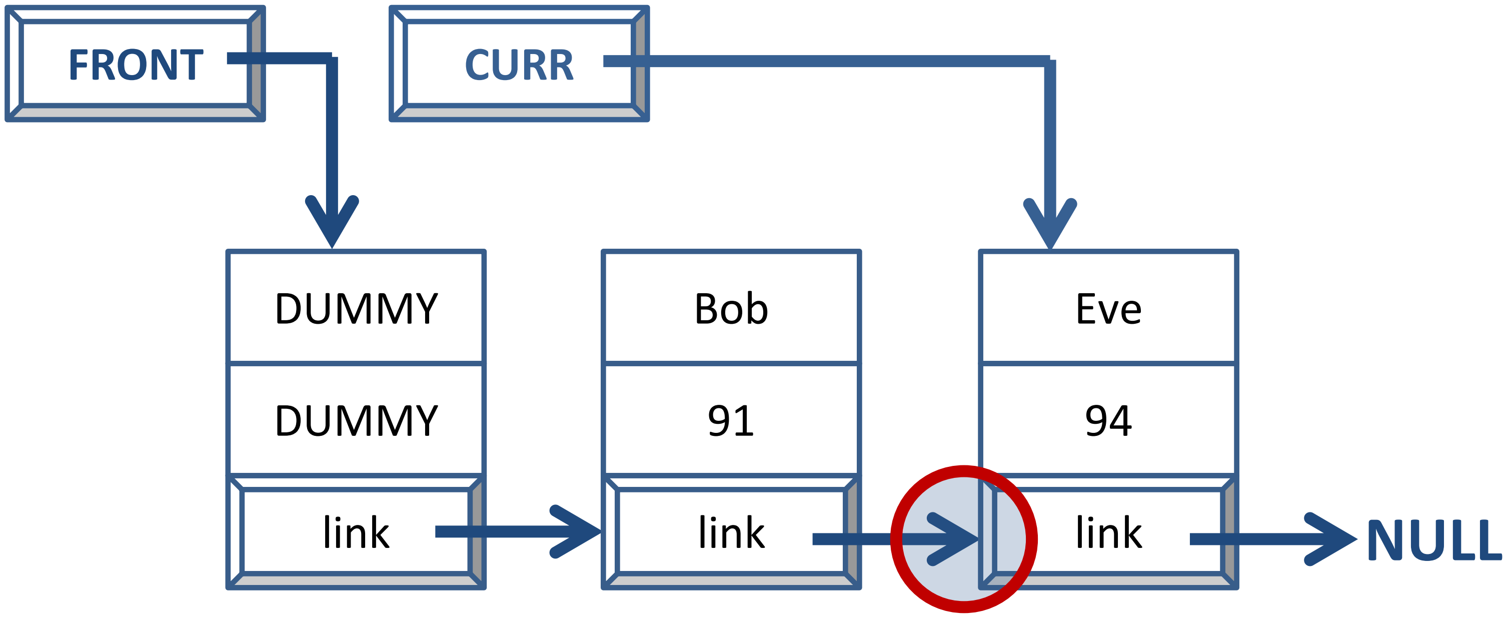
Demonstration of Traversal



```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link) {
```

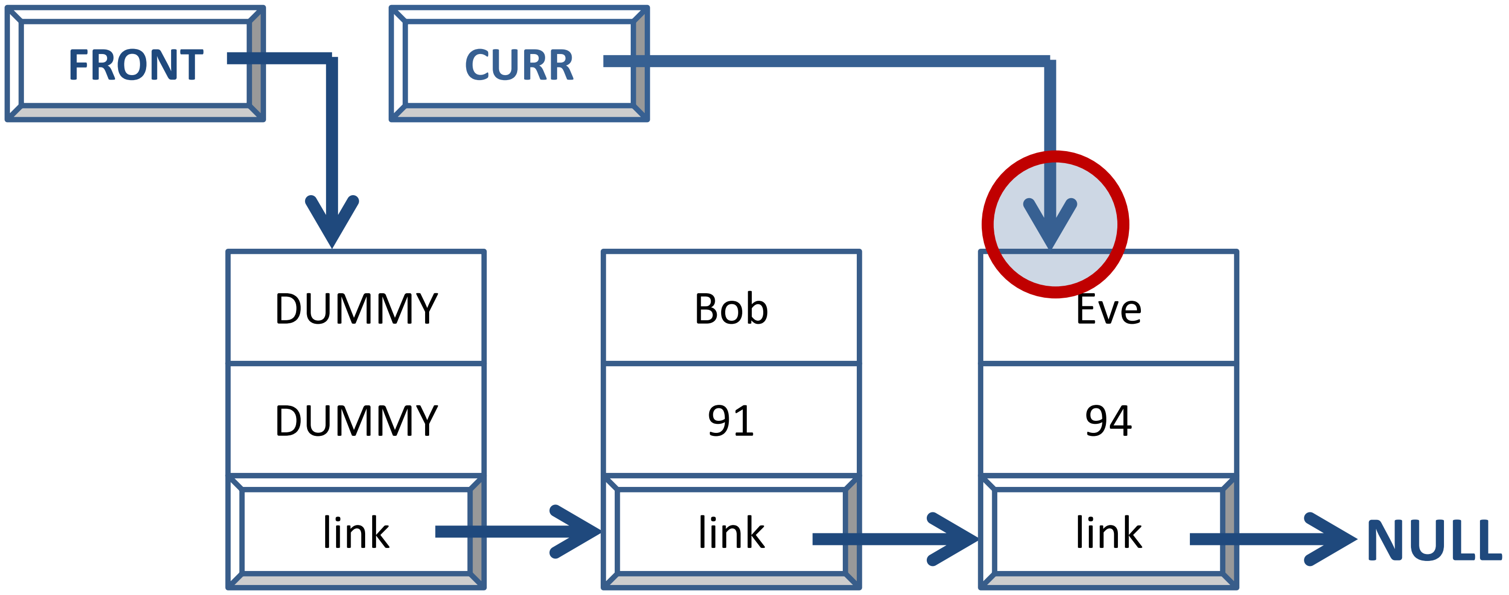
UMBC
AN HONORS UNIVERSITY IN MARYLAND

Demonstration of Traversal



```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link  {
```

Demonstration of Traversal

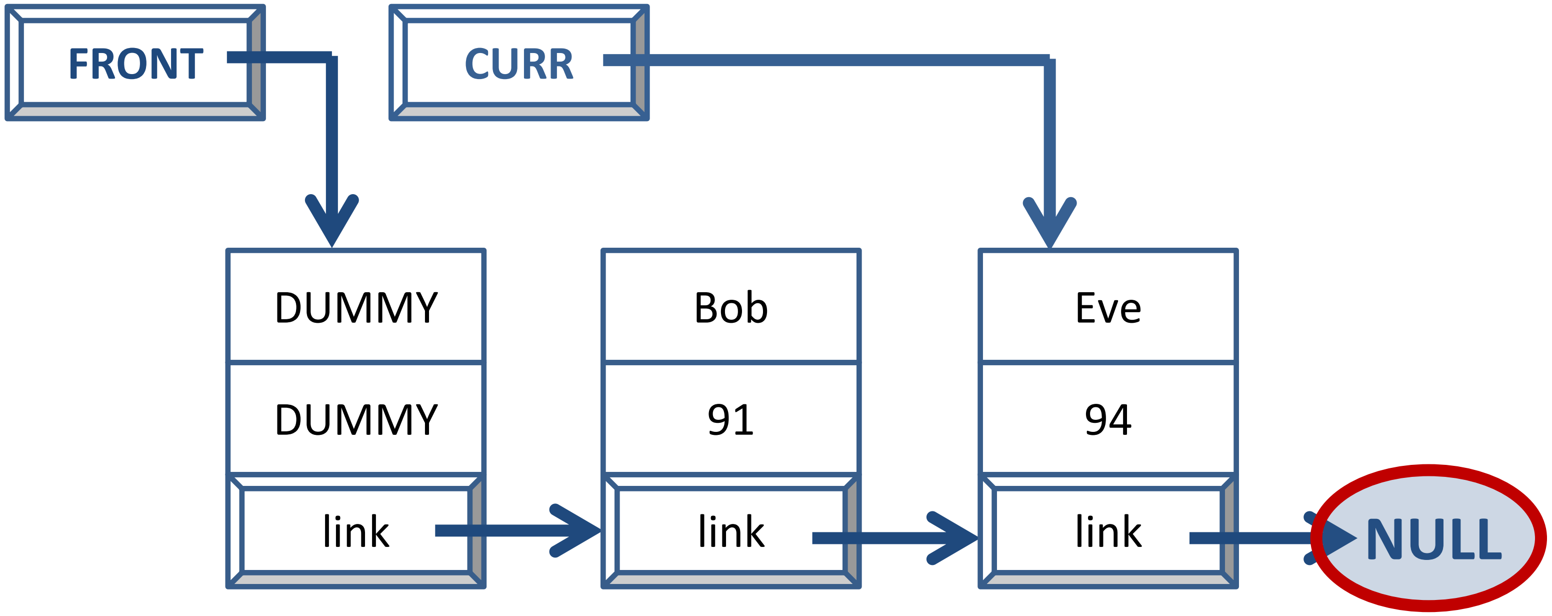


```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link) {
```

✓

```
    // print information (Eve)
```

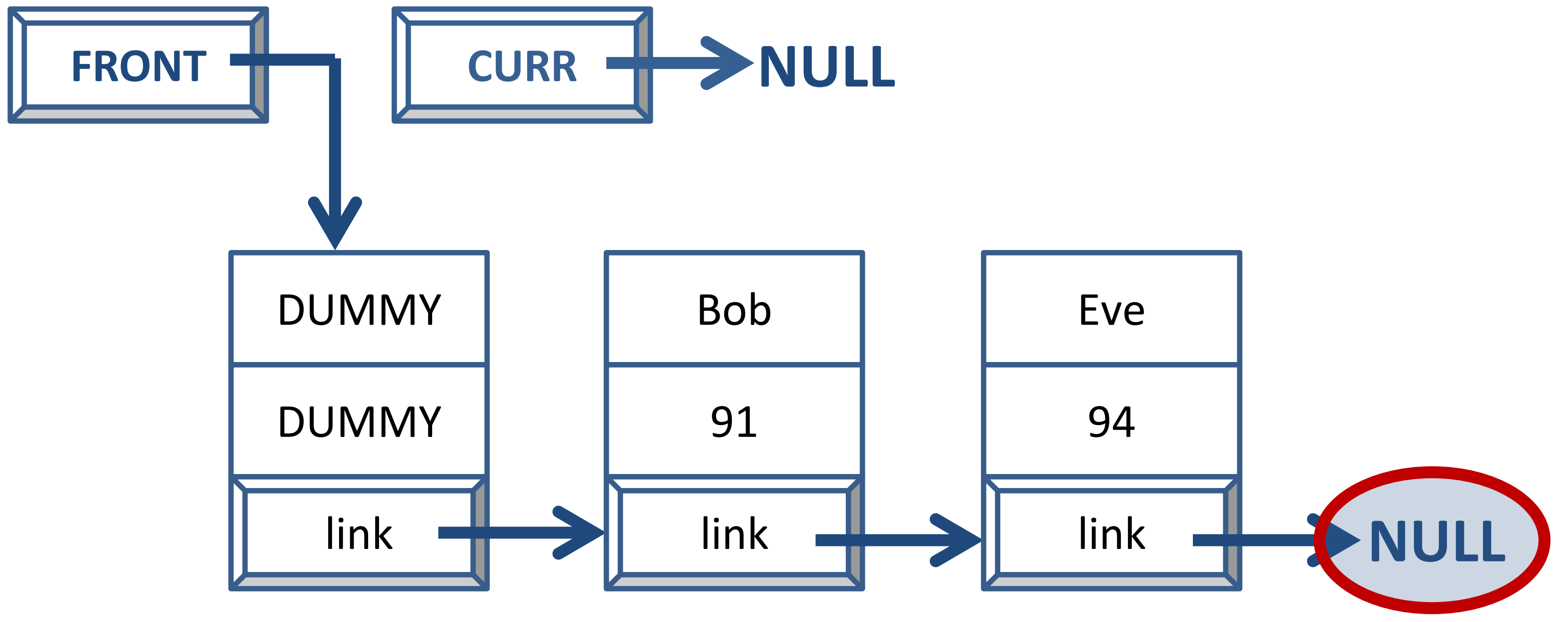
Demonstration of Traversal



```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link  {
```

UMBC AN HONORS UNIVERSITY IN MARYLAND

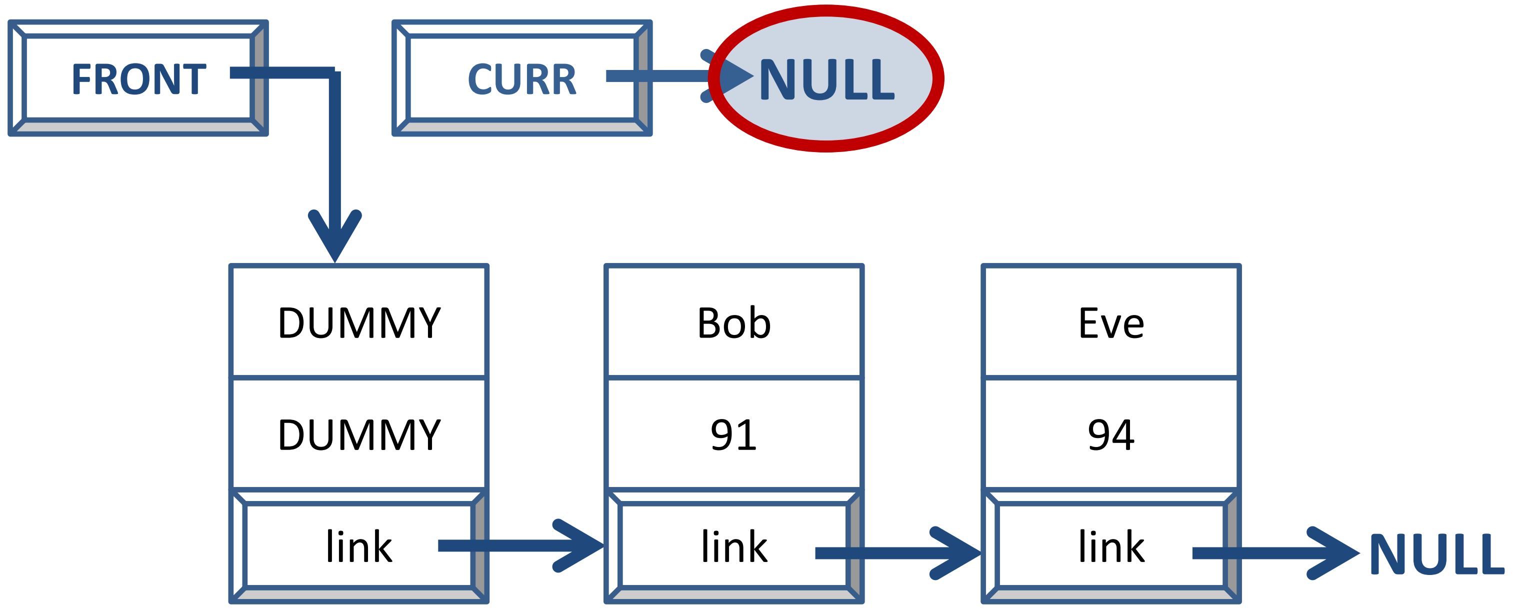
Demonstration of Traversal



```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link  {
```

UMBC AN HONORS UNIVERSITY IN MARYLAND

Demonstration of Traversal



```
for (CURR = FRONT; CURR != NULL; CURR = CURR->link) {

} // exit the loop
```

UMBC

AN HONORS UNIVERSITY IN MARYLAND

Insertion and Deletion

On the Board

Announcements

- Project 3 is out – get started now!
  - It is due Thursday, March 31st