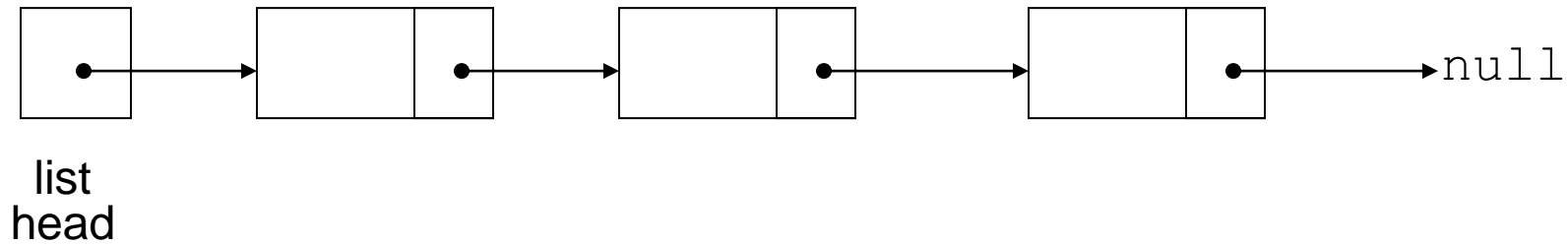


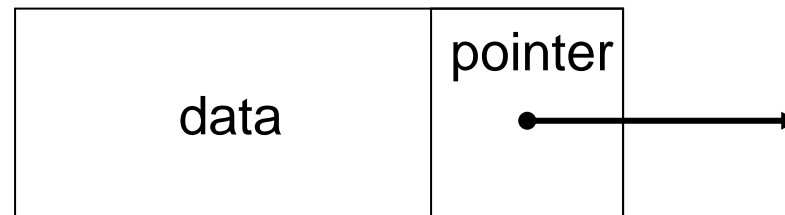
Linked Lists



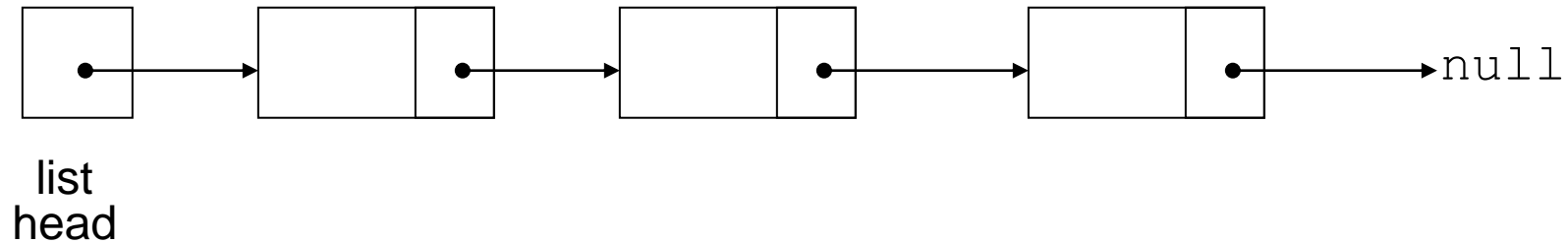
- Linked list: set of data structures (nodes) that contain links to other nodes

Node Organization

- A node contains:
 - **data**: one or more data fields – may be organized as structure, object, etc.
 - **pointer**: that can point to another node



Linked List Organization

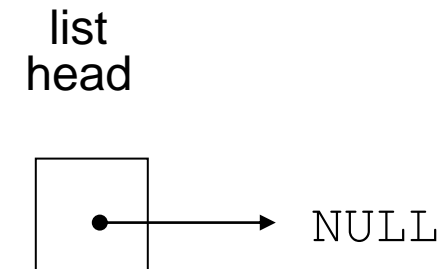


Linked lists may contain 0 or more nodes:

- Has a list head to point to first node
- Last node points to `null`

An empty list contains 0 nodes,

- The list head points to `null`



Declaring a Node

```
#include <iostream>
using namespace std;

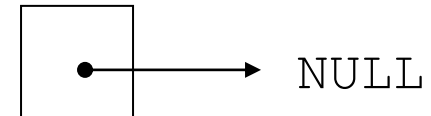
struct Node {
    Node(int data = 0, Node* next = nullptr) : data(data), next(next) {}
    int data;
    Node* next;
};

int main() {
    Node* myList = nullptr;
}
```

No memory is allocated at this time

A list is a pointer to first node, which doesn't really exist in an empty list

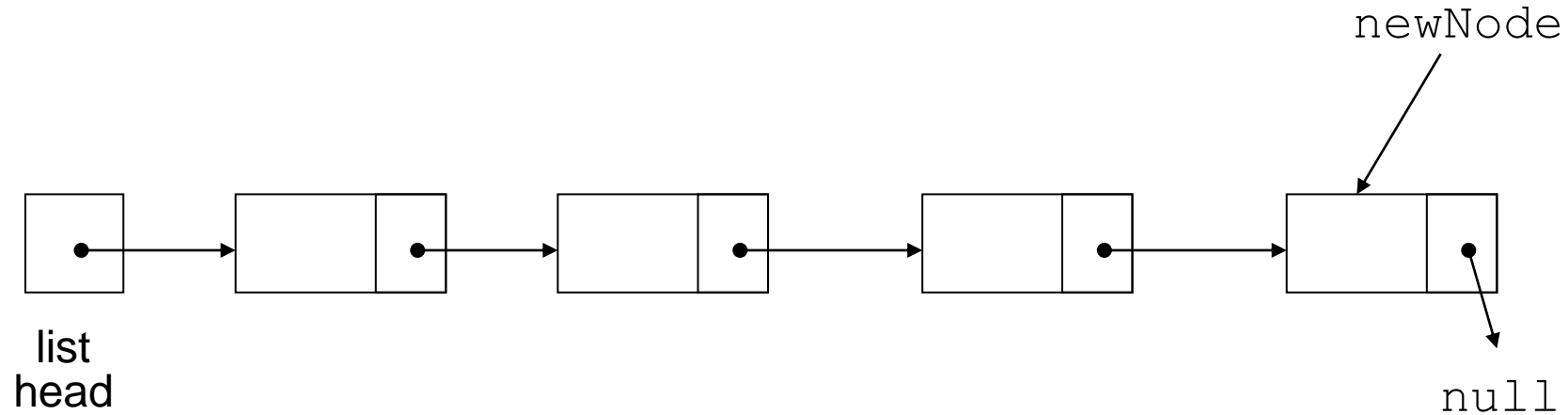
list
head



Linked List – Basic Operations

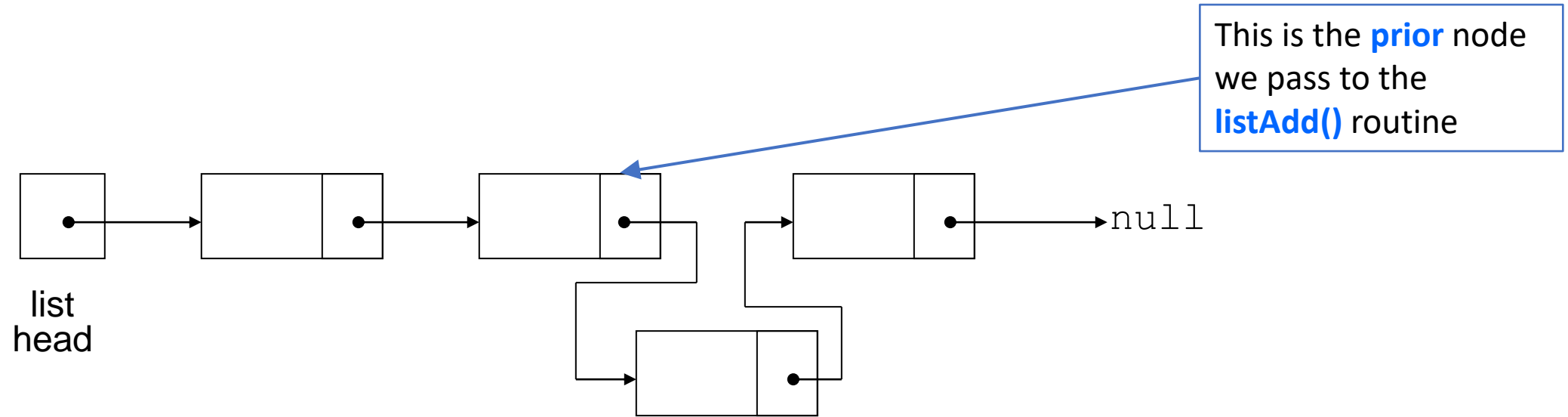
- Append a node to the tail (end) of the list
- Append a node to the head of the list
- Insert a node within the list
- Traverse the linked list
 - To display nodes perhaps, and to
 - Get the list size
- Delete a node
- Delete/destroy the list

Linked Lists – inserting a node at tail



- Nodes may be added to (or removed) the end of the list
- When adding to the tail, is the list's head pointer changing?

Linked Lists – inserting a node in the middle



- Nodes may be inserted between other nodes