CS 162
Intro to CS 2

Linked Structures-
Linked List

Links

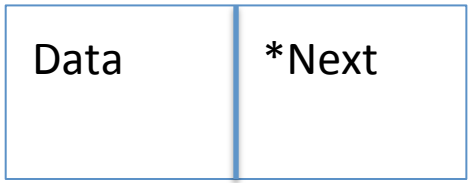
- Use a data structure to hold data and one or more pointers
  - Typically called a node
- Usually done with a struct in C/C++
- Dynamic structure so size can change during runtime
- Number of pointers and interpretation will determine how the linked structure is used

Example

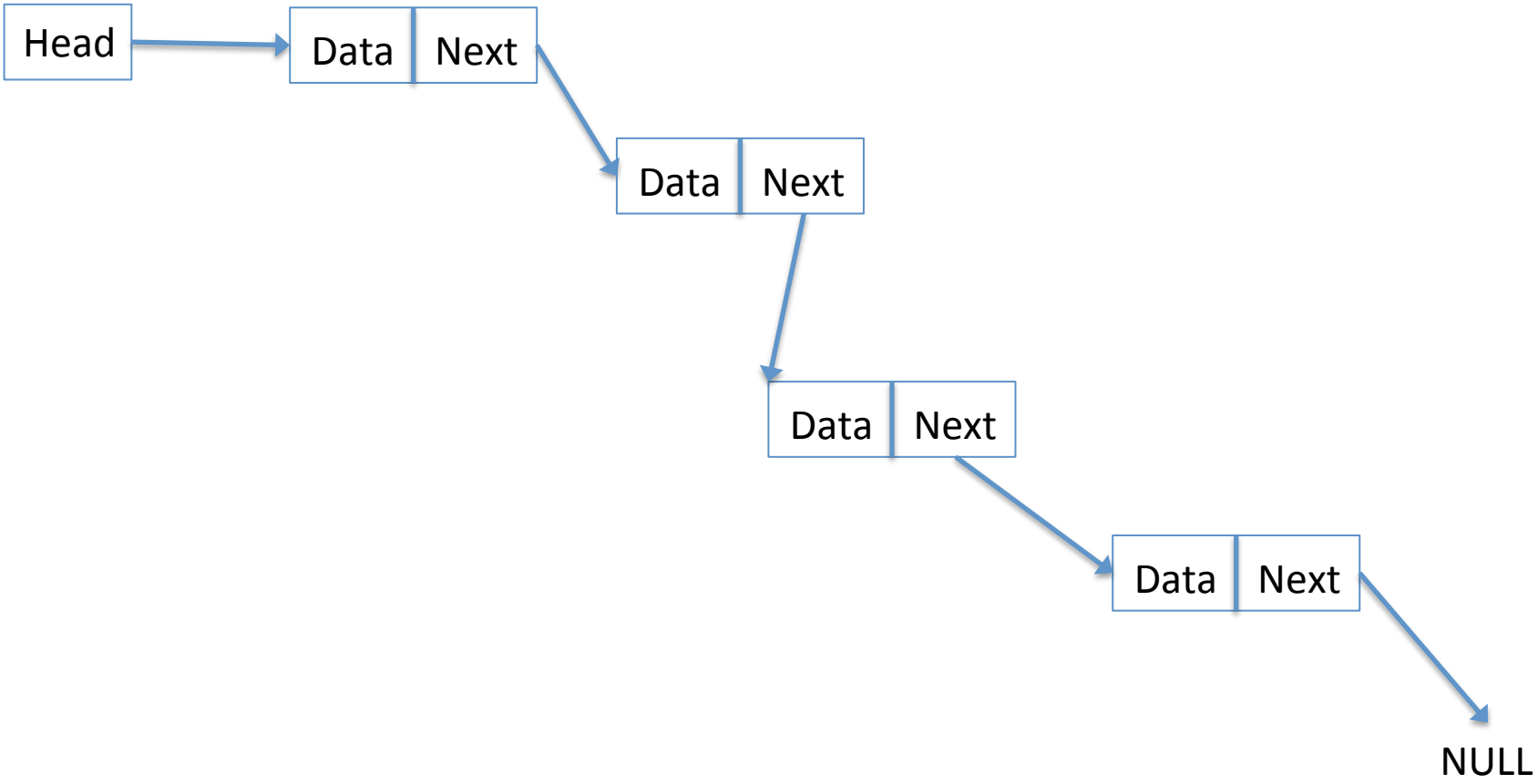
- Struct with 1 data element & 2 pointer elements
- Interpretation one-
  - Pointers viewed as next and previous
  - Used as doubly linked list (go forwards and back)
- Interpretation two-
  - Pointers viewed as left and right
  - Used as a binary tree, links to subtrees

Linked List

- Simplest example
- Each node points to the next node
- Node will have data and the next pointer
- Typically drawn as:



Graphically



Details

- You have a pointer to the beginning of the list
- Each node points to the next node
- You can only traverse one way!
- If you want a previous node you must start over
- Terminates with a NULL pointer

Using a Struct

```
Struct  IntNode
{
private:
     int data;
     IntNode *link;
 };
```

Dynamically allocate-

intNode* node4 = new IntNode;

Necessary functions are written separately

Linked List Class

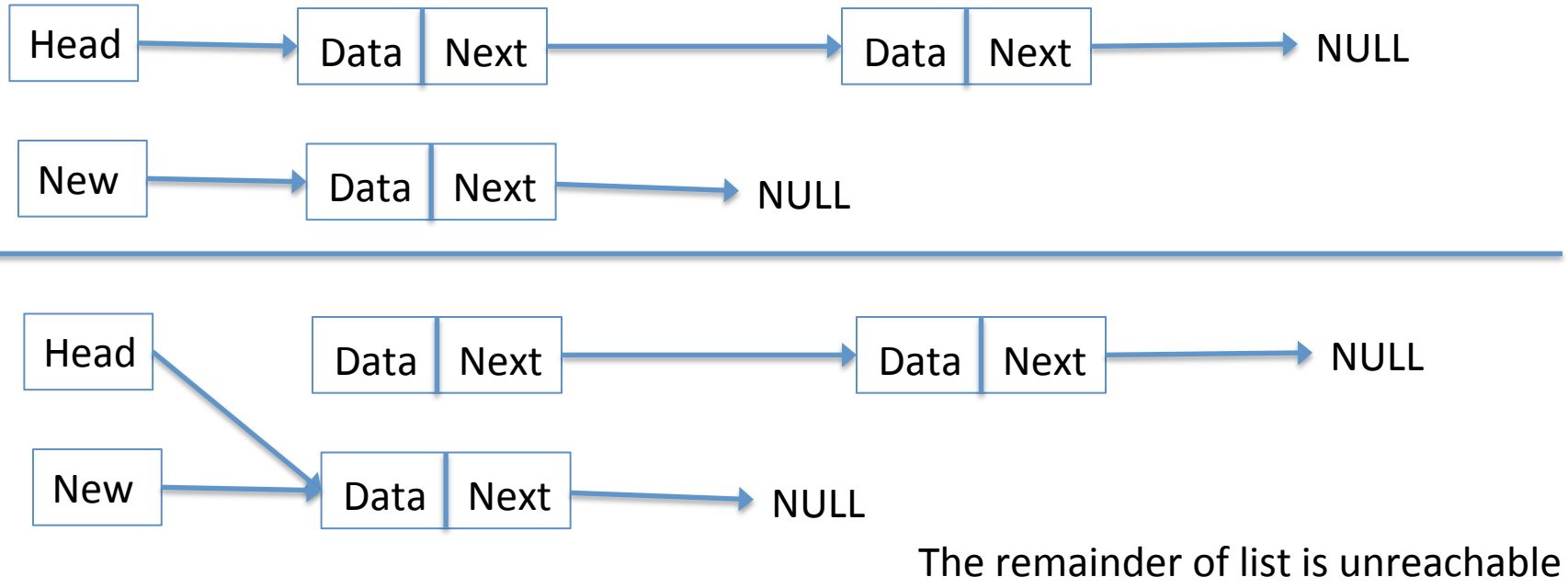
- class IntNode
  {
  public:
    IntNode() { }
    IntNode(int theData, IntNode* theLink)
        : data(theData), link(theLink) { }
    IntNode* getLink() const {return link;}
    int getData() const {return data;}
    void setData(int theData) {data = theData;}
    void setLink(IntNode* pointer) {link=pointer;}
  private:
    int data;
    IntNode *link;
  };
  typedef IntNode* IntNodePtr;

From the publisher's slides

A Key Point!

- You must always have a pointer to all nodes

Insert a new node in list

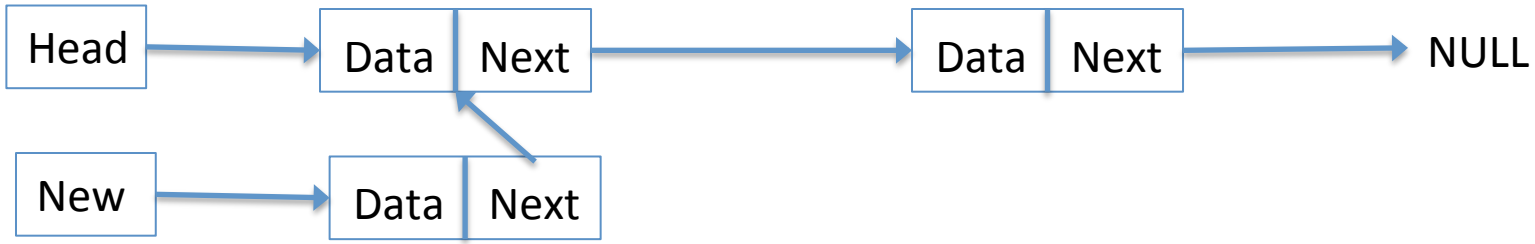


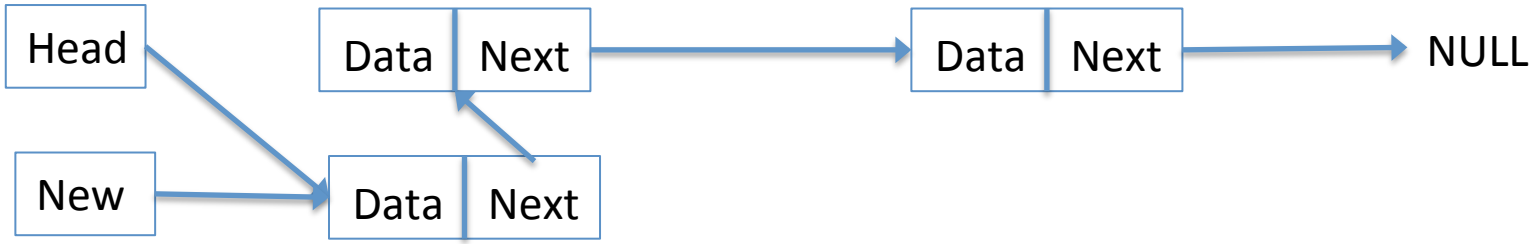
The remainder of list is unreachable

A Key Point!

- To do it correctly

Create a new node and set New->Next = Head





Head = New

Linked Structures

- Use a pointer to connect or link nodes
- Dynamic structure
  - Make as many as you need
  - Manage your memory!!
- Avoid dangling pointers!
  - Make sure you always have a pointer to each node
- It REALLY helps to draw out your linked structure with pencil and paper <<<<