Goals Today

Use pointers & dynamic memory to create a node.

Use nodes to create a linked list.

Recap:

Pointers: - allow to keep track of data in memory space
- needed for dynamic memory allocation
- can point to user defined types (structs, classes)

Dynamic Memory

- lets create variables "on the fly"
- uses heap memory
- allows for greater program flexibility
  ↳ can write a program that

works with various sized files

Nodes & Lists

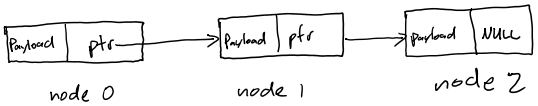
Wednesday, January 31, 2018 10:15 AM

Arrays use contiguous space in memory

Need to insert element?
- copy all the elements

Alternative Approach

1. Allocate space for elements dynamically
2. Let's link the elements via pointers
3. Use a node for each element



Singly Linked List - type dynamic data struct

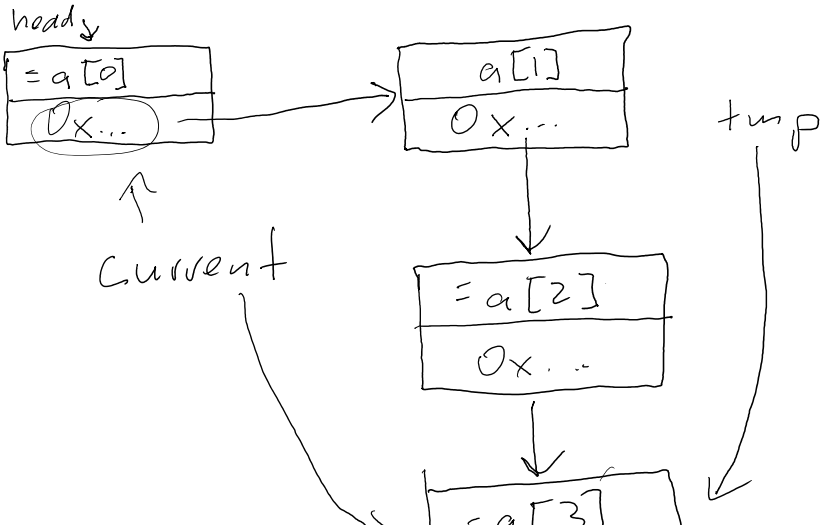
Each element is a node

```
struct    Node
{  string  item;
   ListNode  *next;
};
```

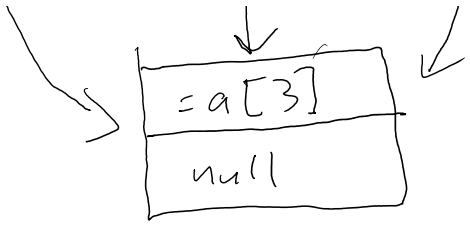
- circularly defined
- each instance of this struct will be one node of the LL.

e.g. copy the elements from an array[4] to a linked list (4 nodes).

```
string a[4] = {"uno", "dos", "tres", "cuatro"}

ListNode *head, *tmp, *current;

head = new Node;

head->item = a[0];
head->next = nullptr;
current = head;
i=1;
while (i<4)      i=3
{   tmp = new Node;
```



```
{   tmp = new Node;
    tmp->item = a[i];
    tmp->next = nullptr;
    current->next = tmp;
    current = tmp;
    i++;
}
```




```
//e.g. traverse the LL
current = head;
while (current != nullptr) {
        cout << current->item << endl;
        current = current->next;
}
```

(* Curren

current

t).next

⇒next