

DYNAMIC MEMORY ALLOCATION LINKED LISTS

Problem Solving with Computers-I

C++

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

int main(){
    cout<<"Hola Facebook\n";
    return 0;
}
```



Dynamic memory management

- To allocate memory on the heap use the 'new' operator
- To free the memory use delete

```
int *p= new int;  
delete p;
```

Creating a small list

- Define an empty list
- Add a node to the list with data = 10

```
struct Node {  
    int data;  
    Node* next;  
};
```

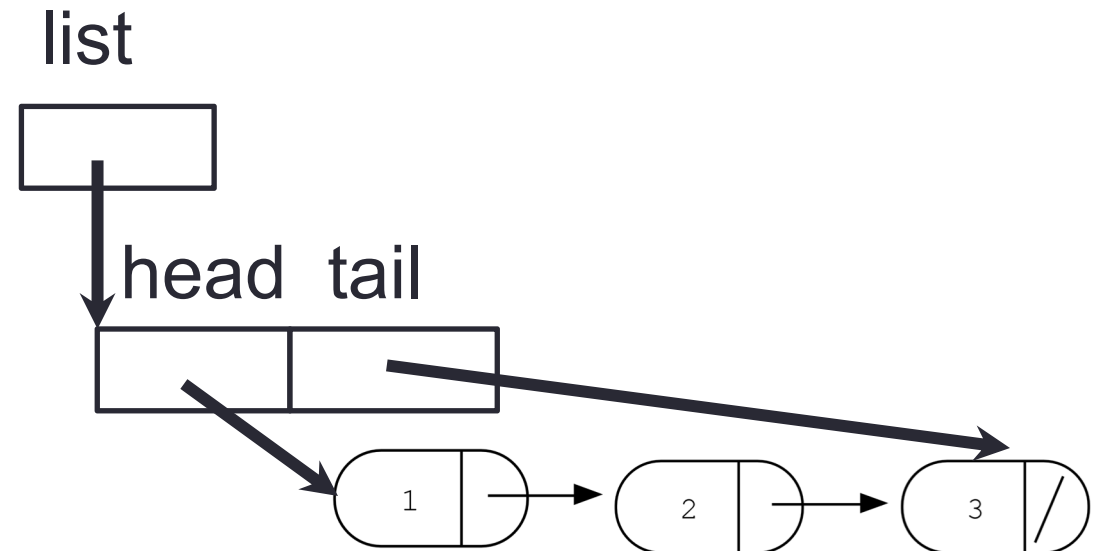
```
struct LinkedList {  
    Node* head;  
    Node* tail;  
};
```

Inserting a node in a linked list

```
Void insertNode(LinkedList* h, int value) ;
```

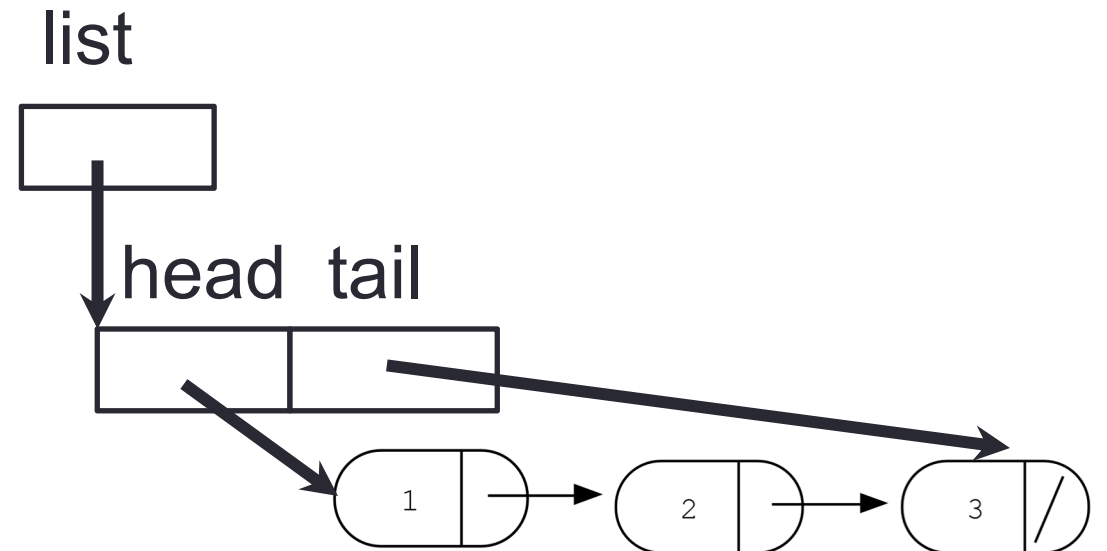
Iterating through the list

```
int lengthOfList(LinkedList* list) {  
    /* Find the number of elements in the list */  
}
```



Deleting the list

```
int freeLinkedList(LinkedList * list) {  
    /* Free all the memory that was created on the heap*/  
}
```



}

Next time

- More linked lists
- Dynamic arrays
- Dynamic memory pitfall