new - allocate memory

delete thing
delete[] array

Classes have fixed numbers of fields.

Array Operations

- Update contents by index a[2] = 4;
- Access contents by index a[2].

Can't

Ask size

Grow or shrink

Automatically check indices

ADTs (abstract data types)

templatization (template)

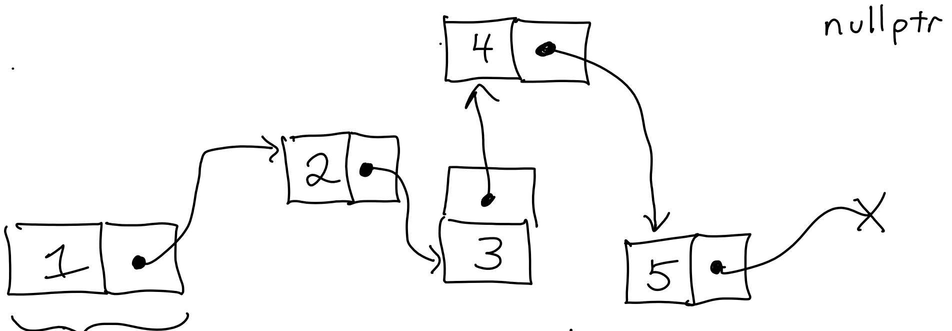
List<T>

- int size() : how many elements are in the list?
- void insertAtTail(T s) : adds an element to the end of the list
- int find(T s)
- void remove(int index)
- void insertAtIndex(int idx, T s) ["c" | "x" | "w"] → (2, "a") → ["c" | "x" | "a" | "w"]
- string get(int idx)
- void set(int idx, T s)
- void insertAtHead
- void removeTail

interface: how you use the ADT

check indices!

LinkedList is a kind of List
Poodle is a kind of Dog



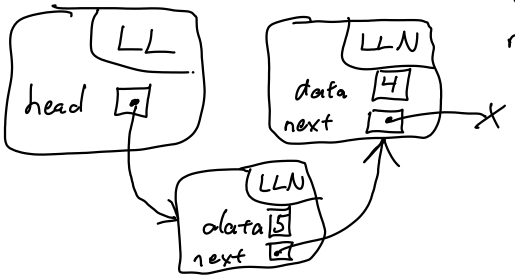
value
ptr to next

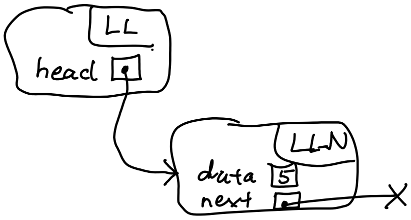
```
template ....
class LinkedList<T> : public List<T>{
    public:
        ......

    private:
        LinkedListNode<T>* head;
```

```
template <typename T>
class LinkedListNode<T> {
private:
    T data;
    LinkedListNode<T>* next;
public:
    T getData()
        .
}
```

```
Method insertAtHead(T value):
    LinkedListNode* nh ← new LLN
    nh→setData(value);
    nh→setNext(this→head);
    this→head ← nh;
```

make sure: works both w/ empty & non-empty





```
Method getSize():
    acc ← 0
    nodeptr ← head
    While nodeptr ≠ null:
        acc ← acc + 1
        nodeptr ← nodeptr→next
    EndWhile
    Return acc
End Method
```

O(n)

Alternative version of size:

LinkedList fields

Node* head
int size

insert: does insertion
& size++

getSize $O(1)$

Invariant — statement which is <u>always</u> true

size = length of the chain starting at head