

### Abstract Data Types (ADT):

List ADT - Purpose	Function Definition

### List Implementation

What types of List do we want?

### C++ Templates:

- 1.
- 2.
- 3.

### Templated Functions:

```

functionTemplate1.cpp
1
2
3 T maximum(T a, T b) {
4     T result;
5     result = (a > b) ? a : b;
6     return result;
7 }

```

Where to put templated code?

### Templated Classes:

```

List.h
1 #pragma once
2
3
4 class List {
5     public:
6
7
8
9
10
11
12     private:
13
14
15 };

List.hpp
1
2
3
4
5

```

### Two Basic Implementations of List:

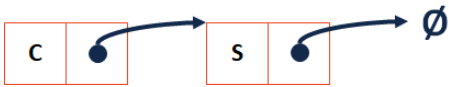
- 1.
- 2.

## Linked Memory:



```

List.h
28 class ListNode {
29     T & data;
30     ListNode * next;
31     ListNode(T & data) : data(data), next(NULL) { }
32 };
  
```



## Coding with Linked Lists: Examples

```

List.h
1 #pragma once
2
3 template <typename T>
4 class List {
5     public:
6         /* ... */
7     private:
8     ...
28     class ListNode {
29         T & data;
30         ListNode * next;
31         ListNode(T & data) : data(data), next(NULL) { }
32     };
33
34
35
36
37
38
39 };
  
```

```

List.hpp
9 #include "List.h"
10
11 template <typename T>
12 void List<T>::insertAtFront(T & t) {
13
14
15
16
17
18
19
20 }
25
26 template <typename T>
27 void List<T>::printReverse() const {
28
29 }
30
31
32
33
34
35
39 template <typename T>
40 T List<T>::operator[](unsigned index) {
41
42
43
44 }
...
48 template <typename T>
49 typename List<T>::ListNode *
50 List<T>::_index(unsigned index) {
51
52
53
54
55
56 }
  
```

## CS 225 – Things To Be Doing:

1. mp\_stickers extra credit due Feb. 15 ( 3 days)!
2. lab\_memory due Sunday.
3. Daily POTDs