

C++ Templates:

- 1.
- 2.
- 3.

Templated Functions:

```
functionTemplatel.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.

A Linked List implementation of a List:

```
List.cpp
1 #pragma once
2
3 template <typename T>
4 class List {
5     public:
6         /* ... */
7
8
9
10
11
12
13
14
15
16     private:
17         class ListNode {
18             public:
19                 const T data;
20                 ListNode * next;
21                 ListNode(T & data) :
22                     data(data), next(nullptr) { }
23
24         };
25
26     ListNode *head_;
27
28     /* ... */
29 };
```

Implementing a basic List operation:



Implementing a basic List operation:

```

List.hpp
9  #include "List.h"
...
14  template <typename T>
15  void List<T>::insertAtFront(const T & d) {
16
17
18
19
20
21
22  }
    
```

Finding in a list:



```

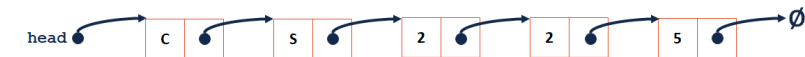
List.hpp
57  template <typename T>
58  typename List<T>::ListNode *&
    List<T>::_index(unsigned index) {
59
60
61  }
62
63
64
65
    
```

What is the return type of `_index`?

Building functionality with `_index()`:

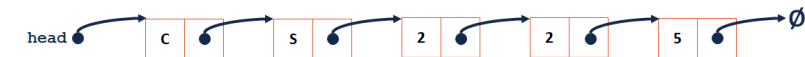
```

List.hpp
48  template <typename T>
49  T & List<T>::operator[](unsigned index) {
50
51
52
53  }
54
    
```



```

List.hpp
103  template <typename T>
104  T List<T>::remove(unsigned index) {
105
106
107
108
109  }
    
```



CS 225 – Things To Be Doing:

1. lab attendance for extra credit starts tomorrow
2. Daily POTDs