



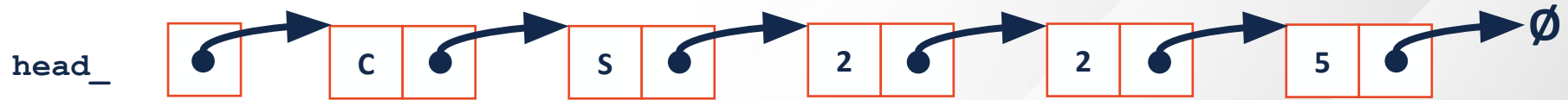
Doubly Linked List

Learning Objectives

1. Understand the differences between singly and doubly linked lists
2. Implement list functions with doubly linked lists



Singly Linked List



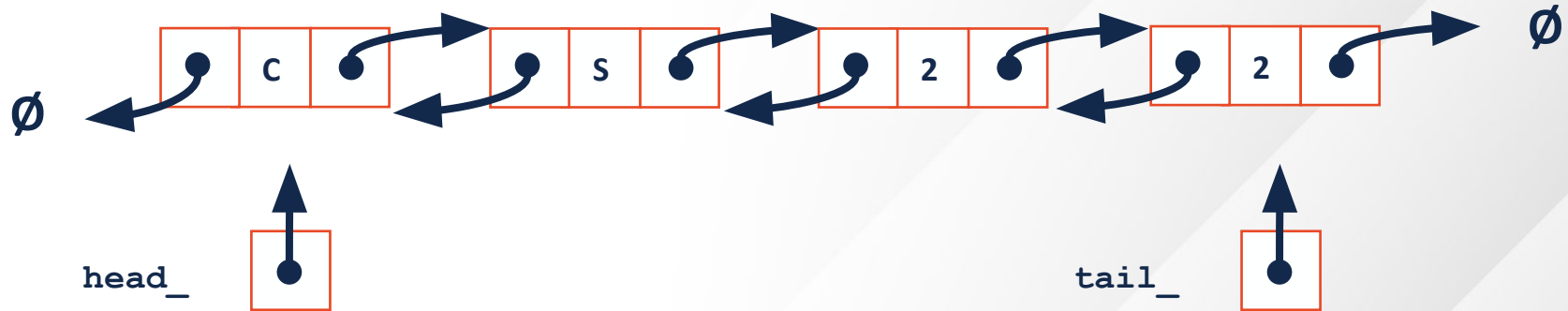
List.h

```
1 #pragma once
2
3 template <class T>
4 class List {
5     public:
6         /* ... */
7
8         insertAtFront(const T &data);
9     private:
10         struct ListNode {
11             T data;
12             ListNode * next;
13             ListNode(const T & data) :
14                 data(data), next(NULL) { }
15         };
16
17         ListNode *head_;
18
19
20 };
21
22 #include "List.hpp"
```

List.hpp

```
1
2 template <class T>
3 void List<T>::insertAtFront(const T &data) {
4     ListNode *tmp = new ListNode(data);
5     tmp->next = head_;
6     head_ = tmp;
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 }
```

Doubly Linked List



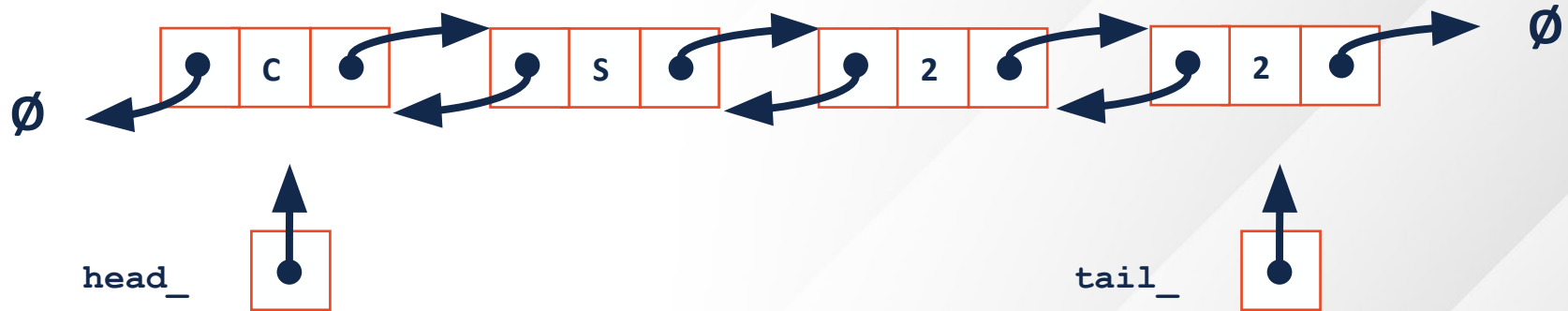
List.h

```
1 #pragma once
2
3 template <class T>
4 class List {
5     public:
6         /* ... */
7
8         insertAtFront(const T &data);
9     private:
10         struct ListNode {
11             T data;
12             ListNode * next;
13             ListNode * prev;
14             ListNode(const T & data) :
15                 data(data), next(NULL),
16                 prev(NULL) { }
17         };
18
19         ListNode *head_;
20         ListNode *tail_;
21 };
22
23 ...
24
25 #include "List.hpp"
```

List.hpp

```
1
2 template <class T>
3 void List<T>::insertAtFront(const T &data) {
4     ListNode *tmp = new ListNode(data);
5     tmp->next = head_;
6     head_ = tmp;
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22 }
```

Doubly Linked List



List.h

```
1 #pragma once
2
3 template <class T>
4 class List {
5     public:
6         /* ... */
7
8         insertAtFront(const T &data);
9     private:
10         struct ListNode {
11             T data;
12             ListNode * next;
13             ListNode * prev;
14             ListNode(const T & data) :
15                 data(data), next(NULL),
16                 prev(NULL) { }
17
18         };
19
20         ListNode *head_;
21         ListNode *tail_;
22     };
23
24 ...
25
26 #include "List.hpp"
```

List.hpp

```
1
2 template <class T>
3 void List<T>::insertAtFront(const T &data) {
4     ListNode *tmp = new ListNode(data);
5     tmp->next = head_;
6     tmp->prev = NULL;
7
8     if (head_ != NULL){
9         head-> prev = tmp;
10    }
11
12    if (tail_ == NULL){
13        tail = tmp;
14    }
15
16    head_ = tmp;
17
18
19
20
21
22 }
```

Doubly Linked List

