#8: Stacks and Queues (and Iterators)

Stack ADT

Function Name	Purpose

Queue ADT

Function Name	Purpose

Stack and Queue Implementations

Stack.h

```
 1  #pragma once
 2
 3  #include <vector>
 4
 5  template <typename T>
 6  class Stack {
 7    public:
 8      void push(const T & d);
 9      T pop();
10      bool isEmpty();
11
12    private:
13      std::vector<T> list_;
14  };
15
16  #include "Stack.hpp"
```

Stack.hpp

```
 3  template <typename T>
 4  void Stack<T>::push(const T & d) {
 5    list_.push_back(d);
 6  }
 7
 8  template <typename T>
 9  T Stack<T>::pop() {
10    T data = list_.back();
11    list_.pop_back();
12    return data;
13  }
```

Example 1

```
Queue<int> q;
q.enqueue(3);
q.enqueue(8);
q.enqueue(4);
q.dequeue();
q.enqueue(7);
q.dequeue();
q.dequeue();
q.enqueue(2);
q.enqueue(1);
q.enqueue(3);
q.enqueue(5);
q.dequeue();
q.enqueue(9);
```

Example 2

```
Queue<char> q;
q.enqueue('m');
q.enqueue('o');
q.enqueue('n');
…
q.enqueue('d');
q.enqueue('a');
q.enqueue('y');
q.enqueue('i');
q.enqueue('s');
q.dequeue();
q.enqueue('h');
q.enqueue('a');
```

Accessing Every Element in Our Data Collection
Suppose we want to look through every element in our data structure.
What if we don't know what our data structure even looks like?

What are the necessary member functions for a class implementing an iterator?

What are the necessary functions for that class's iterator?