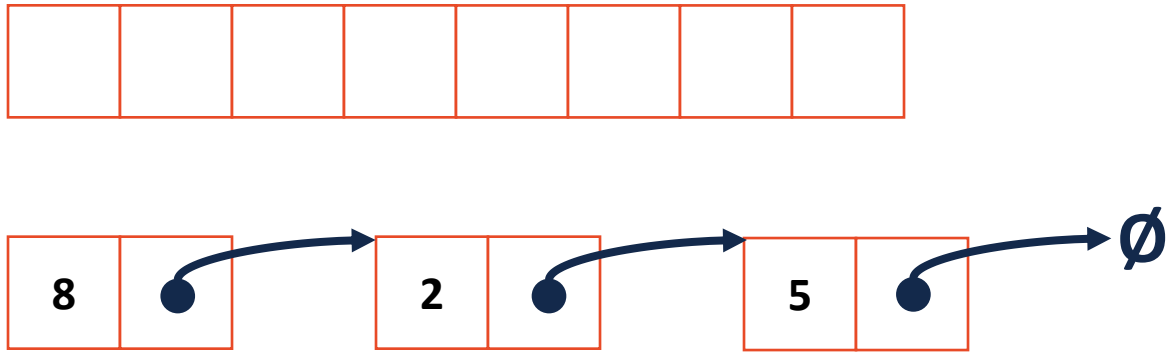
CS 225

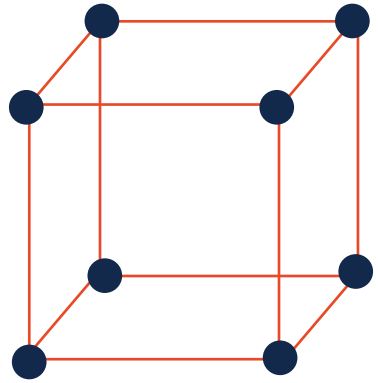
Data Structures

Sept. 29 – Functors

Iterators

Iterators give client code access to traverse the data!





Operators:

- operator++
- operator==
- operator!=
- operator=
- operator*

Types of iterators:

- Forward
- Backward
- Bidirectional

QueueIter.h

```
 1 #ifndef QUEUE_H
 2 #define QUEUE_H
 3
 4 template <class QE>
 5 class Queue {
 6   public:
 7
 8
 9
10
11
12
13
14
15
16   private:
17
18
19
20 };
21
22 #endif
```

Where does the iterator go?

What additional functions are needed for an iterator?

stlList.cpp

```
 1 #include <list>
 2 #include <string>
 3 #include <iostream>
 4
 5 struct Animal {
 6   std::string name, food;
 7   bool big;
 8   Animal(std::string name = "blob", std::string food = "you", bool big = true) :
 9     name(name), food(food), big(big) { /* none */ }
10 }
11
12 int main() {
13   Animal g("giraffe", "leaves", true), p("penguin", "fish", false), b("bear");
14   std::list<Animal> zoo;
15
16   zoo.push_back(g);
17   zoo.push_back(p);   // std::list's insertAtEnd
18   zoo.push_back(b);
19
20   for ( std::list<Animal>::iterator it = zoo.begin(); it != zoo.end(); it++ ) {
21     std::cout << (*it).name << " " << (*it).food << std::endl;
22   }
23
24   return 0;
25 }
```

Queue.h

```
4  template <class QE>
5  class Queue {
6    public:
7      class QueueIterator : public std::iterator<std::bidirectional_iterator_tag, T> {
8        public:
9          QueueIterator(unsigned index);
10         QueueIterator& operator++();
11         bool operator==(const QueueIterator &other);
12         bool operator!=(const QueueIterator &other);
13         QE& operator*();
14         QE* operator->();
15       private:
16
17     }
18
19
20
21
22   private:
23     T* arr_; unsigned capacity_, count_, entry_, exit_;
24
25
26 };
```

Function Objects (aka “Functors”)

Functors are objects that can be called like a function.

```
1
2
3
4
```

Doubler.h

```
1  #ifndef DOUBLER_H
2  #define DOUBLER_H
3
4  class Doubler {
5    public:
6      int operator()(int value);
7  };
8
9  #endif
```

```
1  #include "Doubler.h"
2
3  int Doubler::operator()(int value) {
4    return value * 2;
5  }
```

```
 1 template<class Value, class Modifier>
 2 Value modify(Value value, Modifier modifier) {
 3    return modifier(value);
 4 }
```

```
11 Doubler d;
12 Tripler t;
13 int value = 4;
14 modify<int, Doubler>(value, d);
15 modify<int, Tripler>(value, t);
```

```
1  template<class Iter, class Formatter>
2  void print(Iter first, Iter second, Formatter printer) {
3    while ( first != second ) {
4      printer( *first );
5      first++;
6    }
7  }
```

This is a function called _____ whose inputs are two _____ and a _____.

This function appears to:

stlList.cpp

```
 1 #include <list>
 2 #include <string>
 3 #include <iostream>
 4
 5 struct Animal {
 6   std::string name, food;
 7   bool big;
 8   Animal(std::string name = "blob", std::string food = "you", bool big = true) :
 9     name(name), food(food), big(big) { /* none */ }
10 }
11
12 int main() {
13   Animal g("giraffe", "leaves", true), p("penguin", "fish", false), b("bear");
14   std::list<Animal> zoo;
15
16   zoo.push_back(g);
17   zoo.push_back(p);   // std::list's insertAtEnd
18   zoo.push_back(b);
19
20   for ( std::list<Animal>::iterator it = zoo.begin(); it != zoo.end(); it++ ) {
21     std::cout << (*it).name << " " << (*it).food << std::endl;
22   }
23
24   return 0;
25 }
```

Trees

CS 225 – Things To Be Doing

Exam 4 (Programming/MP2) starts tomorrow!

More Info: https://courses.engr.illinois.edu/cs225/fa2017/exams/

MP3: Available now!

Up to +7 extra for submission by Monday, Oct. 2!

Lab: lab_quacks due Sunady, Oct 1!

Fun lab!

POTD

Every Monday-Friday – *Worth +1 Extra Credit /problem (up to +40 total)*