

CS 225

Data Structures

Sept. 25 – Array Resize

```
1 #include "Stack.h"
2
3 template <class T>
4 void Stack::push(T & t) {
5     // If we are about to overflow, double the size of the array:
6     if (count_ + 1 == size_) {
7         size_ *= 2;
8         T * newArray = new T[size_];
9         for (unsigned i=0; i < count_; i++) { newArray[i] = arr_[i]; }
10        delete arr_;
11        arr_ = newArray;
12    }
13
14    // Insert (push) the element into the array-backed stack:
15    arr[ count_++ ] = t;
16 }
17
18 template <class T>
19 T & Stack::pop() {
20     return arr[ --count_ ];
21 }
```

Stack.h

```
1 #ifndef STACK_H
2 #define STACK_H
3
4 template <class T>
5 class Stack {
6     public:
7         Stack();
8         Stack(const Stack &other);
9         ~Stack();
10        Stack& operator=(const Stack &other);
11
12        void push(T & t);
13        T & pop();
14        bool isEmpty() const;
15
16    private:
17        T * arr_;
18        unsigned size_, count_;
19
20 };
21
22 #endif
```

Exam Updates

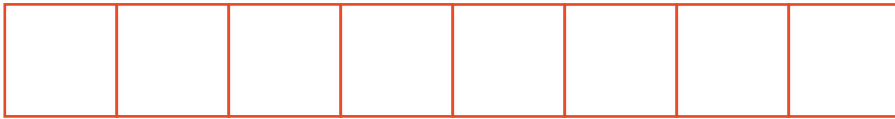
Right Now: Exam 3 – Theory Exam (Advanced C++)

Next Week: Exam 4 – Programming Exam (MP2-focused)

- Inheritance
- Linked memory



Resize Strategy – Details



Resize Strategy – Details



Implications of Design

1.

```
struct ListNode {  
    T & data;  
    ListNode * next;  
    ListNode(T & data) : data(data), next(NULL) { }  
};
```

2.

```
T ** arr;
```

```
struct ListNode {  
    T * data;  
    ...  
};
```

3.

```
T * arr;
```

```
struct ListNode {  
    T data;  
    ...  
};
```


Implications of Design

| | Storage by Reference | Storage by Pointer | Storage by Value |
|---|----------------------|--------------------|------------------|
| Who manages the lifecycle of the data? | | | |
| Is it possible for the data structure to store NULL? | | | |
| If the data is manipulated by user code while in our data structure, is the change reflected in our data structure? | | | |
| Is it possible to store literals? | | | |
| Speed | | | |

Data Lifecycle

Storage by reference:

```
1 Sphere s ;  
2 myStack.push(s) ;
```

Storage by pointer:

```
1 Sphere s ;  
2 myStack.push(s) ;
```

Storage by value:

```
1 Sphere s ;  
2 myStack.push(s) ;
```

Possible to store NULL?

Storage by reference:

```
struct ListNode {  
    T & data;  
    ListNode * next;  
    ListNode(T & data) : data(data), next(NULL) { }  
};
```

Storage by pointer:

```
T ** arr;
```

Storage by value:

```
T * arr;
```

Data Modifications

```
1 Sphere s(1);  
2 myStack.push(s);  
3  
4 s.setRadius(42);  
5  
6 Sphere r = myStack.pop();  
7 // What is r's radius?
```

Literals

```
1 Stack<int> myStack;  
2  
3 myStack.push(1);  
4 myStack.push(2);  
5 myStack.push(3);
```



Speed

Queue.h

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

CS 225 – Things To Be Doing

Exam 3 (Theory, C++) starts Monday

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

MP2: Week #2

Nightly reports are in mp2/grades/

Lab: lab_gdb – Due Sunday, 11:59pm

One of the hardest labs of the semester, important to work with gdb

POTD

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