CS 225
Data Structures

Sept. 25 – Array Resize
```cpp
#include "Stack.h"

template <class T>
void Stack::push(T & t) {
    // If we are about to overflow, double the size of the array:
    if (count_ + 1 == size_) {
        size_ *= 2;
        T * newArray = new T[size_];
        for (unsigned i=0; i < count_; i++) { newArray[i] = arr_[i]; }
        delete arr_;
        arr_ = newArray;
    }

    // Insert (push) the element into the array-backed stack:
    arr_[ count_++ ] = t;
}

template <class T>
T & Stack::pop() {
    return arr[ --count_ ];
}
```
```cpp
#ifndef STACK_H
#define STACK_H

template <class T>
class Stack {

    public:
        Stack();
        Stack(const Stack &other);
        ~Stack();
        Stack& operator=(const Stack &other);
        void push(T & t);
        T & pop();
        bool isEmpty() const;

    private:
        T * arr_{};
        unsigned size_{}, count_{};

};

#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
Implications of Design

1. ```
   struct ListNode {
   T & data;
   ListNode * next;
   ListNode(T & data) : data(data), next(NULL) { }
   }
   ```

2. ```
   T ** arr;
   ```

3. ```
   T * arr;
   ```
## Implications of Design

<table>
<thead>
<tr>
<th>Who manages the lifecycle of the data?</th>
<th>Storage by Reference</th>
<th>Storage by Pointer</th>
<th>Storage by Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Is it possible for the data structure to store NULL?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>If the data is manipulated by user code while in our data structure, is the change reflected in our data structure?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Is it possible to store literals?</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Speed</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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:

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

Storage by pointer:

```c
T ** arr;
```

Storage by value:

```c
T * arr;
```
Data Modifications

<table>
<thead>
<tr>
<th>Line</th>
<th>Code</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Sphere s(1);</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>myStack.push(s);</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s.setRadius(42);</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Sphere r = myStack.pop();</td>
<td>// What is r’s radius?</td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Code</td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td><code>Stack&lt;int&gt; myStack;</code></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td><code>myStack.push(1);</code></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td><code>myStack.push(2);</code></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td><code>myStack.push(3);</code></td>
<td></td>
</tr>
</tbody>
</table>
Speed
 ifndef QUEUE_H
  #define QUEUE_H

  template <class T>
  class Queue {
    public:

    private:

  }

  #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)