CS 225

Data Structures

Sept. 22 – Stack
# Array Implementation

<table>
<thead>
<tr>
<th></th>
<th>Singly Linked List</th>
<th>Array</th>
</tr>
</thead>
<tbody>
<tr>
<td>Insert/Remove at <strong>front</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert at <strong>given</strong> location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove at <strong>given</strong> location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Insert at <strong>arbitrary</strong> location</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Remove at <strong>arbitrary</strong> location</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
#ifndef STACK_H
#define STACK_H

template <class T>
class Stack {
    public:

    private:

};
#endif
Stack Implementations

1.

2.
Linked Memory

head → C → S → 2 → 2 → 5 → ∅
Stack - Linked List

Diagram showing a linked list with nodes containing numbers and a head pointer.
#include "Stack.h"

void Stack::push(T & t) {
}

T & Stack::pop() {
#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:

};

#endif
Array Implementation

T* arr: | C | S | 2 | 2 | 5 | [0] | [1] | [2] | [3] | [4]
```cpp
#include "Stack.h"

void Stack::push(T & t) {
}

T & Stack::pop() {
}
```
Resize Strategy – Details
Resize Strategy – Details
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)