Exam 0 Starts Monday

Covers prerequisite knowledge details on course website

https://courses.engr.illinois.edu/cs225/sp2023/exams/exam0/

Practice on PrairieLearn

Register on PrairieTest
Contacting the Course

cs225admin@lists.cs.illinois.edu
What about C++

Lectures from Previous Semesters Covering C++ Available Here
https://mediaspace.illinois.edu/playlist/dedicated/177553201/1_s10ctiib/1_z2cz05fi
Material from CS 128
https://learncpp.online/lessons
Encapsulation - Classes
Memory Management - Ownership
The “Rule of Three/Five”

If it is necessary to define any one of these three functions in a class, it will be necessary to define all three of these functions:

1.

2.

3.
The “Rule of Zero”

Corollary to Rule of Five

Classes that declare custom destructors, copy/move constructors or copy/move assignment operators should deal exclusively with ownership. Other classes should not declare custom destructors, copy/move constructors or copy/move assignment operators

–Scott Meyers
List ADT
What types of “stuff” do we want in our list?
Templates
T maximum(T a, T b) {
    T result;
    result = (a > b) ? a : b;
    return result;
}
List Implementations

1.

2.
Linked Memory

C → S → 2 → 2 → 5 → Ø
struct ListNode {
    T data;
    ListNode * next;
    ListNode(const T & data) : data(data), next(NULL) { }
};
Linked Memory

head

C → S → 2 → 2 → 5 → ∅
```cpp
#pragma once

class List {
    public:

    private:

};
```
Linked Memory: `insertAtFront`

head → C → S → 2 → 2 → 5 → \(\emptyset\)
# List.h

```cpp
#pragma once

template <class T>
class List {
 public:
  /* ... */

 private:
  struct ListNode {
    T data;
    ListNode * next;
    ListNode(const T & data) :
      data(data), next(NULL) { }
  }
};
```
Running Time of Linked List `insertAtFront`
Linked Memory: \texttt{insertAtNth}
ListNode * & List<T>::_index(int index) {
}

List.hpp