ECE 2574: Data Structures and Algorithms - List: Linked Implementations

C. L. Wyatt

Today we will look at a doubly link-based implementation of the List ADT.

- Review of AbstractList
- Memory management for LinkedList
- Implementing AbstractList methods

Recall our AbstractList interface

```
virtual bool isEmpty() const noexcept = 0;
virtual std::size_t getLength() const noexcept = 0;
virtual void insert(std::size_t position, const T& item)
virtual void remove(std::size_t position) = 0;
virtual void clear() = 0;
virtual T getEntry(std::size_t position) const = 0;
virtual void setEntry(std::size_t position, const T& newV
```

Memory management for LinkedList

Like for ArrayList we need to take care to do proper memory management and implement

- Destructor
- Copy Constructor
- Copy-Assignment

see in-class code

Implementing AbstractList Methods

Now we have to implement the methods in AbstractList
see in-class code

Next Actions and Reminders

- Read CH pp. 289-295 and Appendix E
- Warmup due before noon on We 10/11