COP 3540 Data Structures with OOP

Chapter 5 - Part 2
Linked Lists

Abstract Data Types and Sorted Lists

Abstract Data Types

- A **wonderful** way to deal with data structures.
- An ADT is a way of looking at a data structure focusing on **what** it does and **NOT how** it does whatever it is supposed to do.
- We **implement** an ADT
- Stacks and Queues are examples of ADTs.
- They are ‘concepts’ ; logical data structures.
- Let’s first look how we can have a **linked list** that **implements** a stack.

A Stack Implemented by a Linked List

- Can use an array, but no chance to extend it.

- **Basic operations**: push() and pop() implemented as
  - arr[++top] = newdata; and returnedData = arr[top--];
  - Indexes are checked for stack overflow and stack underflow and then adjusted as shown.

- **For a Stack: implemented as a linked list**, we have
  - theList.insertFirst(data) to **add** an element onto the stack
    - (looks like a 'push')
  - Retrieved data = theList.deleteFirst(). (looks like a pop)

- ➔ The important things: user **calls** push() and pop() for his/her stack operations and has **no knowledge** whether the logical stack is implemented as a physical array or as a physical linked list!