

# Intro to Programming II

## Linked Lists

Chris Brooks

Department of Computer Science  
University of San Francisco

Department of Computer Science — University of San Francisco — p. 17

### 18-2: Arrays

- Previously we talked about using arrays to store sequences of data.
- Advantages:
  - Everything is stored in sequential memory locations.
  - Fast lookup of elements.
- Disadvantages:
  - Hard to resize
  - Removing or adding an element in the middle is costly.

Department of Computer Science — University of San Francisco — p. 17

### 18-3: Linked Lists

- Linked lists have the opposite advantages and disadvantages
- Advantages:
  - Easy to insert and remove
  - Easy to resize
- Disadvantages:
  - Elements are not stored sequentially
  - Finding the *n*th element is slower.

Department of Computer Science — University of San Francisco — p. 17

### 18-4: Linked Lists

- The general idea:
- Each element of the list will “know” who the next element is.
- Let’s try this as a class.

Department of Computer Science — University of San Francisco — p. 17

### 18-5: List elements

- So how do we do this in Java?
- Our Element class needs to have two components:
  - The data that we want to store in the list.
  - A pointer to the next element in the list.

Department of Computer Science — University of San Francisco — p. 17

### 18-6: List elements

### 18-7: Arranging ListItems

- ListItems hook together like a chain.
- All we need to do is keep track of the beginning of the chain.
- No need to allocate everything ahead of time.

Exercises in Computer Systems ... Security in Data Networks ... p. 17

### 18-8: The LinkedList class

- The LinkedList will be responsible for hanging onto the "head" of the list and providing methods for working with the list.
  - Insert()
  - InsertAt(index)
  - get(index)
  - remove(index)
  - find(object)

Exercises in Computer Systems ... Security in Data Networks ... p. 17

### 18-9: The LinkedList class

### 18-13: Adding

- Are there any special cases we need to worry about?

Department of Computer Science — University of San Francisco — p. 18-13

### 18-14: Adding

- Are there any special cases we need to worry about?
  - What if the list is empty?
  - What if it only has one element?
  - What if we're adding at the end?

Department of Computer Science — University of San Francisco — p. 18-14

### 18-15: Exercise

- Code the ListItem and LinkedList classes as described above.
- Write a main method that creates a list and prompts the user for strings, which are always added to the front of the list.
- Add a method to the LinkedList class called toString().
- This should walk the list and print out each of the strings in order.

Department of Computer Science — University of San Francisco — p. 18-15

### 18-16: Adding elements

- To add an element at the front, we point the new element's "next" pointer to whatever head is pointing to, then point head to point to the new element.