

# Linked List Solutions

- Describe the structure of a linked list
  - A linked list consists of a series of nodes
  - Each node contains a single element and a pointer to the following node
- How does this differ from other containers we have used?
  - Each element has its own memory allocation, as opposed to being stored with other elements in a block of memory

- Describe how to
  - a) Add an element to a linked list
    - Allocate memory for the node
    - Set the node's pointer to the address of the node that will follow it in the list
    - Find the node that will precede it in the list and set its pointer to the address of the new node
  - b) Remove an element from a linked list
  - c) Iterate through the elements of a linked list

- Describe how to
  - a) Add an element to a linked list
  - b) Remove an element from a linked list
    - Find the node that precedes our node and set its pointer to the address of the node that follows our node
    - Release the memory allocated to the removed node
  - c) Iterate through the elements of a linked list

- Describe how to
  - a) Add an element to a linked list
  - b) Remove an element from a linked list
  - c) Iterate through the elements of a linked list

Start with the first node in the list

Dereference its pointer

This will give us the second node in the list

We dereference its pointer to get the next node

This continues until we encounter a node where the pointer is null

This is the last node in the list, so we stop here