Topic 12 – Abstract Data Types – P1

Stacks

Stacks

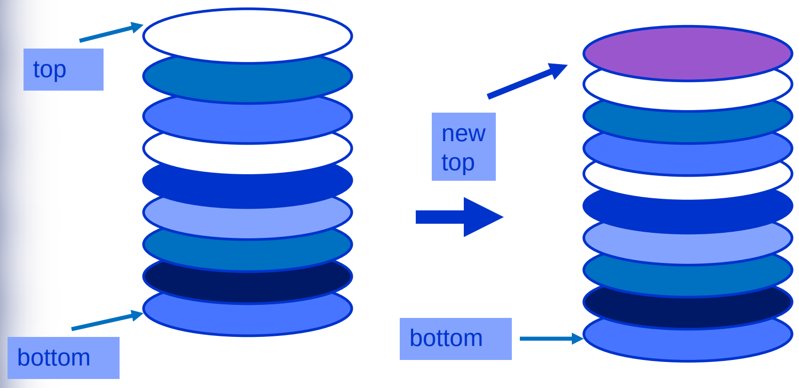
Adding to the front creates a stack

- A stack is an Abstract Data Type (ADT)
- Insertions & Deletions follow the LIFO (or FILO) scheme
  - Last In First Out
  - First In Last Out
- Conceptually: A spring loaded plate mechanism

Conceptual View

Adding an element

New object is added as the new top element of the stack



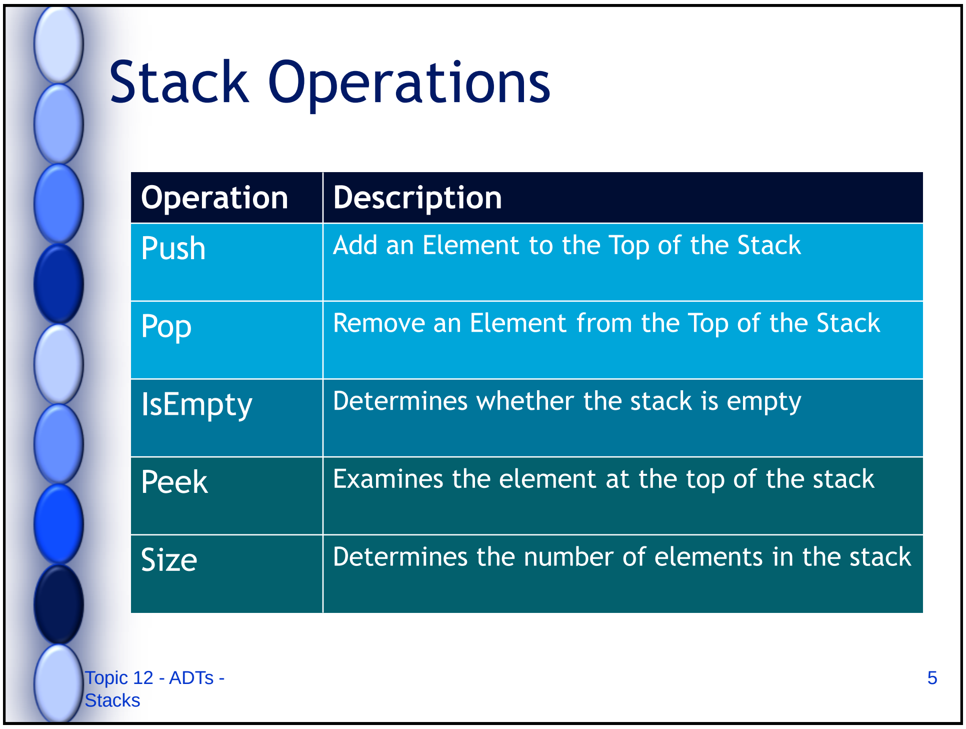
When would you use a stack?

For any application where you need to view the data in reverse order

For Example

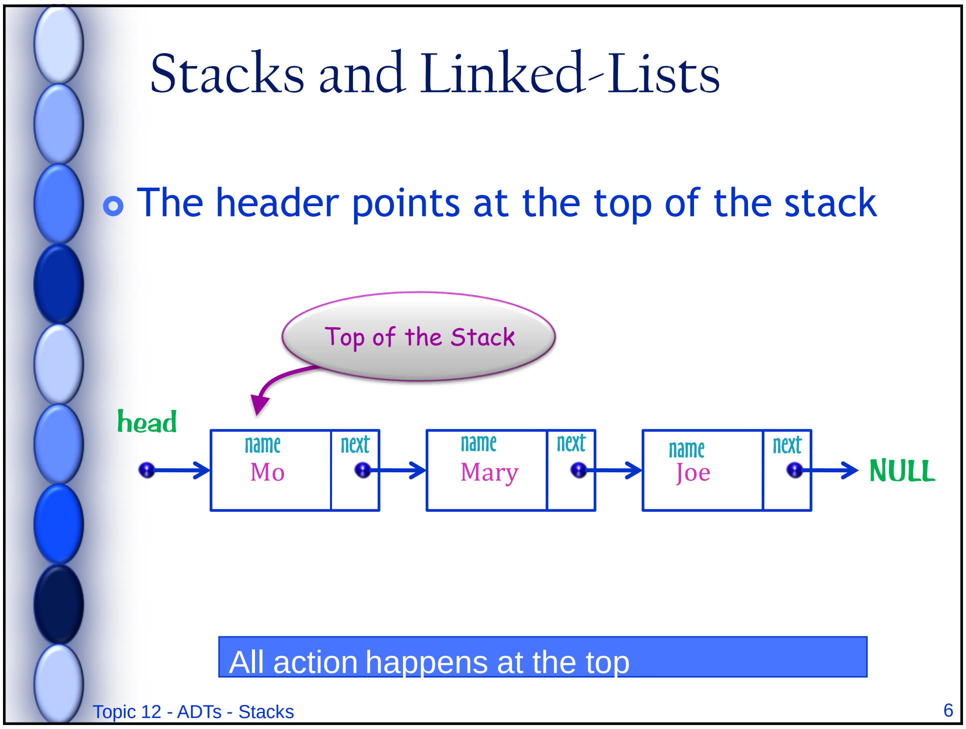
Web Browsers: recently visited sites

Text Editors: undo sequences



Stack Operations

Operation	Description
Push	Add an Element to the Top of the Stack
Pop	Remove an Element from the Top of the Stack
IsEmpty	Determines whether the stack is empty
Peek	Examines the element at the top of the stack
Size	Determines the number of elements in the stack



Stacks and Linked-Lists

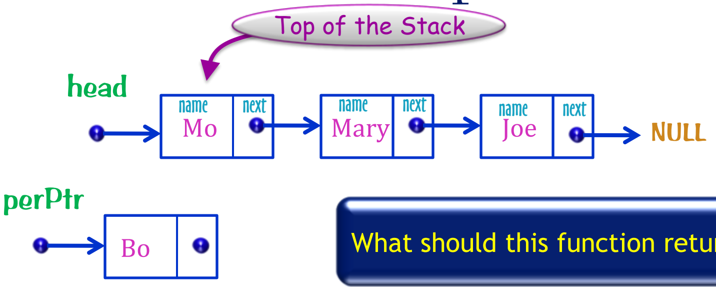
- The header points at the top of the stack

All action happens at the top

Stack Operations using a Linked List

- Push
  - → add an element to the front of the list
- Pop
  - → remove an element from the front of the list
- IsEmpty
  - → check if head points to NULL
- Peek
  - → Examine the element at the top of the stack
- Size
  - → either count each element
  - → OR keep track as you push and pop

Push → Add to the Top of the Stack



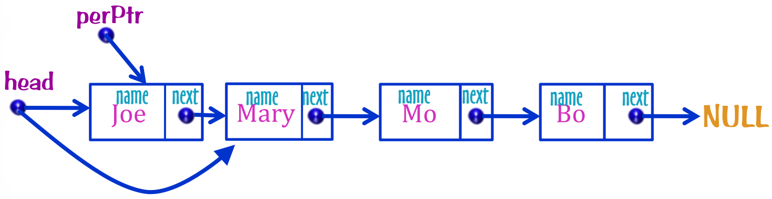
What should this function return?

- This is what we've been doing thus far

Stack Operations using a Linked List

- Push
  - → add an element to the front of the list
- Pop
  - → remove an element from the front of the list
- IsEmpty
  - → check if head points to NULL
- Peek
  - → Examine the element at the top of the stack
- Size
  - → either count each element
  - → OR keep track as you push and pop

Pop - Remove from the Front

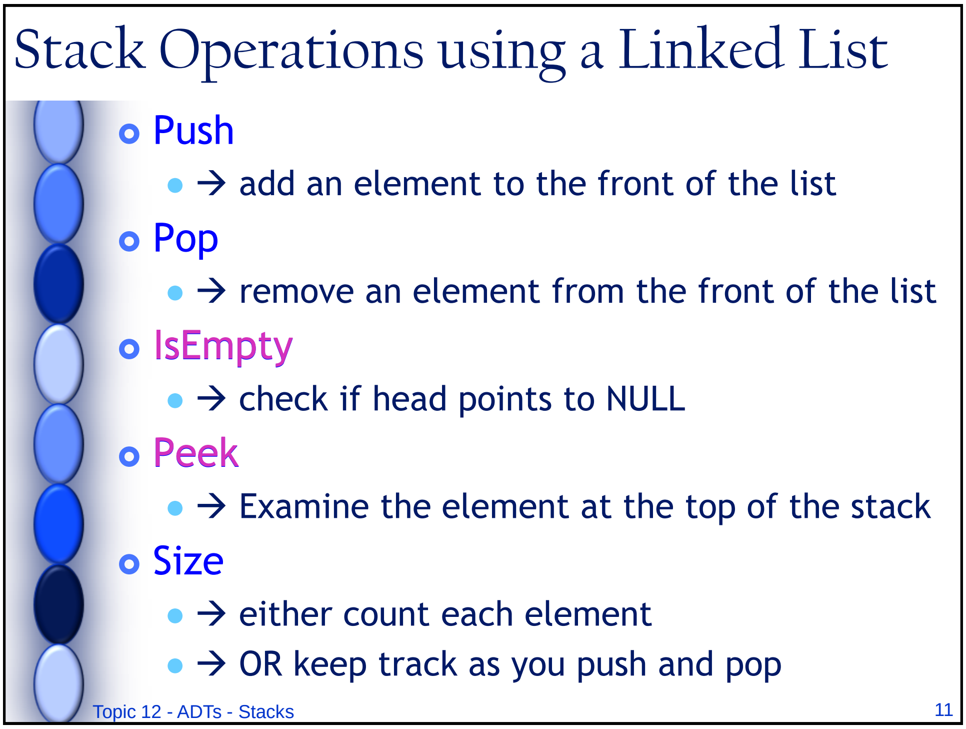


1. We need a 2nd pointer (perPtr)
2. Point perPtr to head
3. Re-assign head to the next element (perPtr->next)
4. De-allocate the node
5. to be safe… set perPtr to NULL

What if the list is empty?

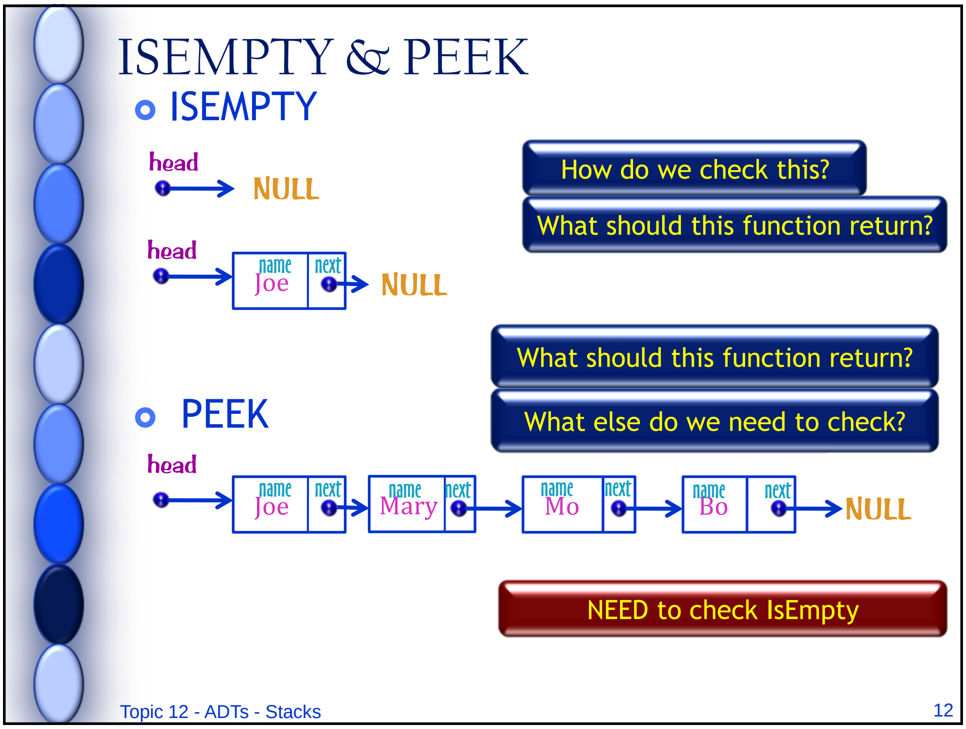
NEED to check IsEmpty

What should this function return?



Stack Operations using a Linked List

- Push
  - → add an element to the front of the list
- Pop
  - → remove an element from the front of the list
- IsEmpty
  - → check if head points to NULL
- Peek
  - → Examine the element at the top of the stack
- Size
  - → either count each element
  - → OR keep track as you push and pop



ISEMPTY & PEEK

- ISEMPTY

How do we check this?

What should this function return?

What should this function return?

What else do we need to check?

- PEEK

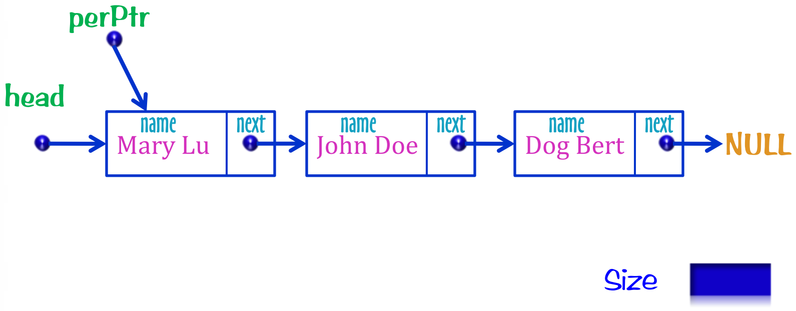
NEED to check IsEmpty

Stack Operations using a Linked List

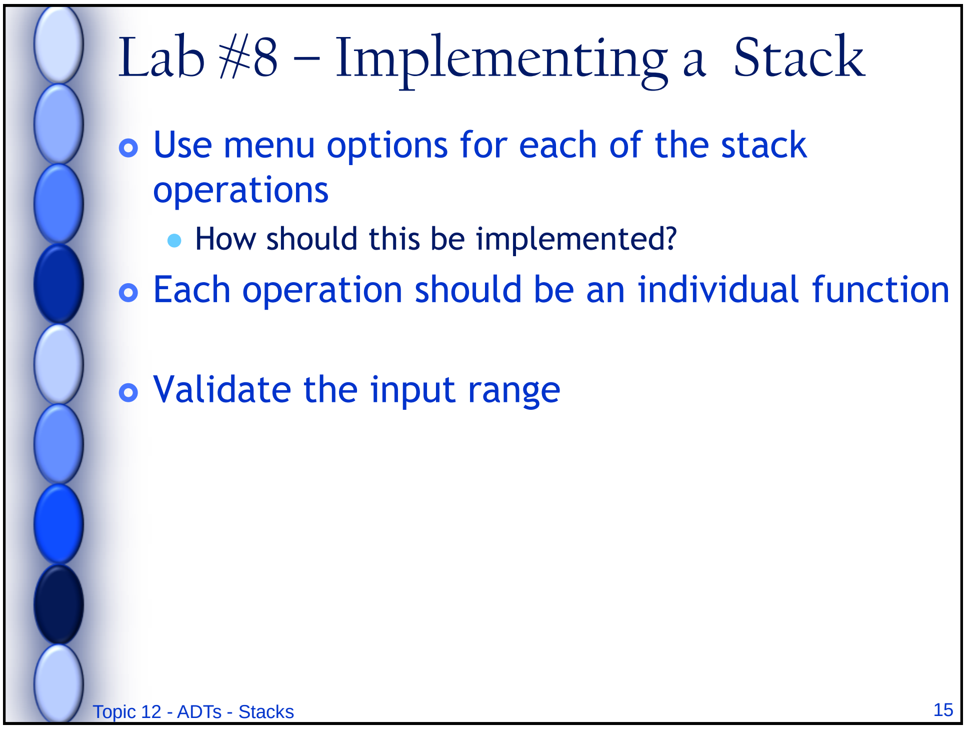
- Push
  - → add an element to the front of the list
- Pop
  - → remove an element from the front of the list
- IsEmpty
  - → check if head points to NULL
- Peek
  - → Examine the element at the top of the stack
- Size
  - → either count each element
  - → OR keep track as you push and pop

Finding the Size

- We can maintain a variable that tracks it as we go… but we can easily create a function to determine the size



What should this function return?



Lab #8 – Implementing a Stack

- Use menu options for each of the stack operations
  - How should this be implemented?
- Each operation should be an individual function
- Validate the input range