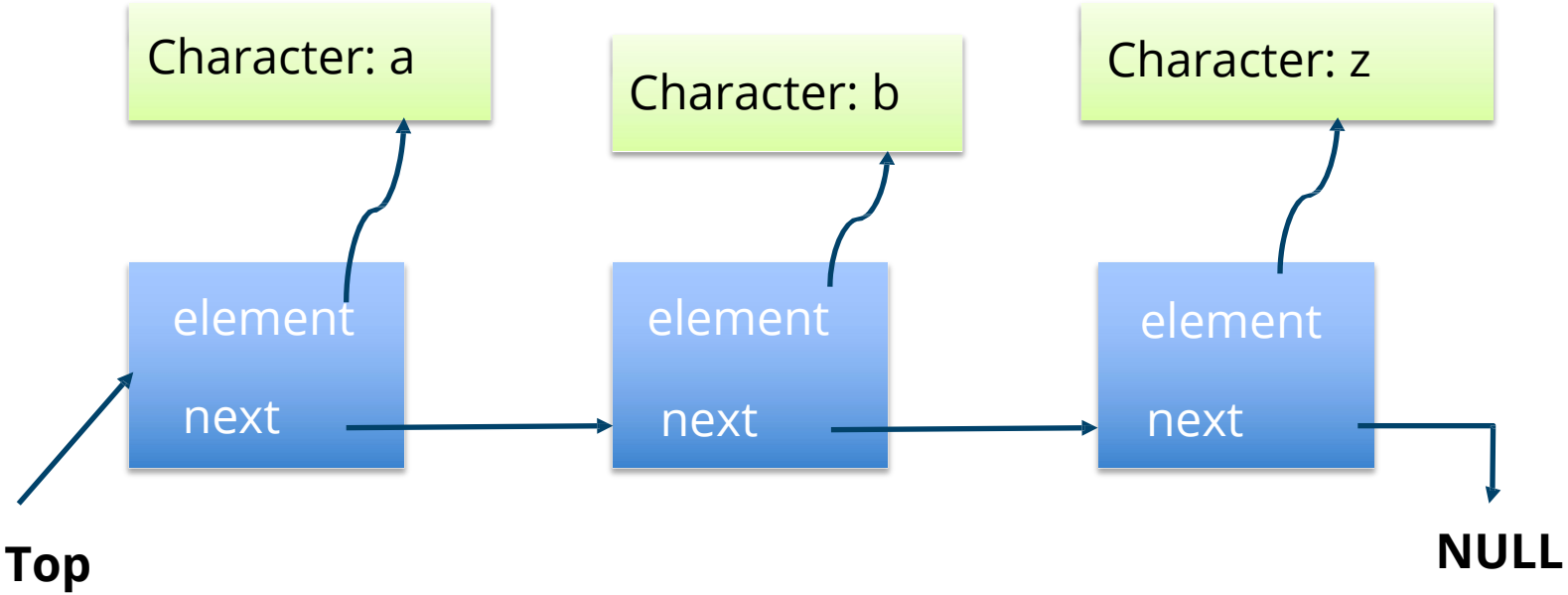
Stack Implementation
LinkedStack

COMP128 Data Structures

Linked Structures use a Node class: LinearNode

```
/*LinearNode represents a node in a linked list.*/
public class LinearNode<E> {
   private LinearNode<E> next;
   private E element;
   /**
    * Creates a node storing the specified element.
    * @param elem the element to be stored within the new
      node
    */
   public LinearNode(E elem){
    next = null;
    element = elem;
   }

… getters and setters
```



The typical Stack class contains a declaration of the LinearNode class and private instance variables of type LinearNode for top, along with methods for all of the operations.

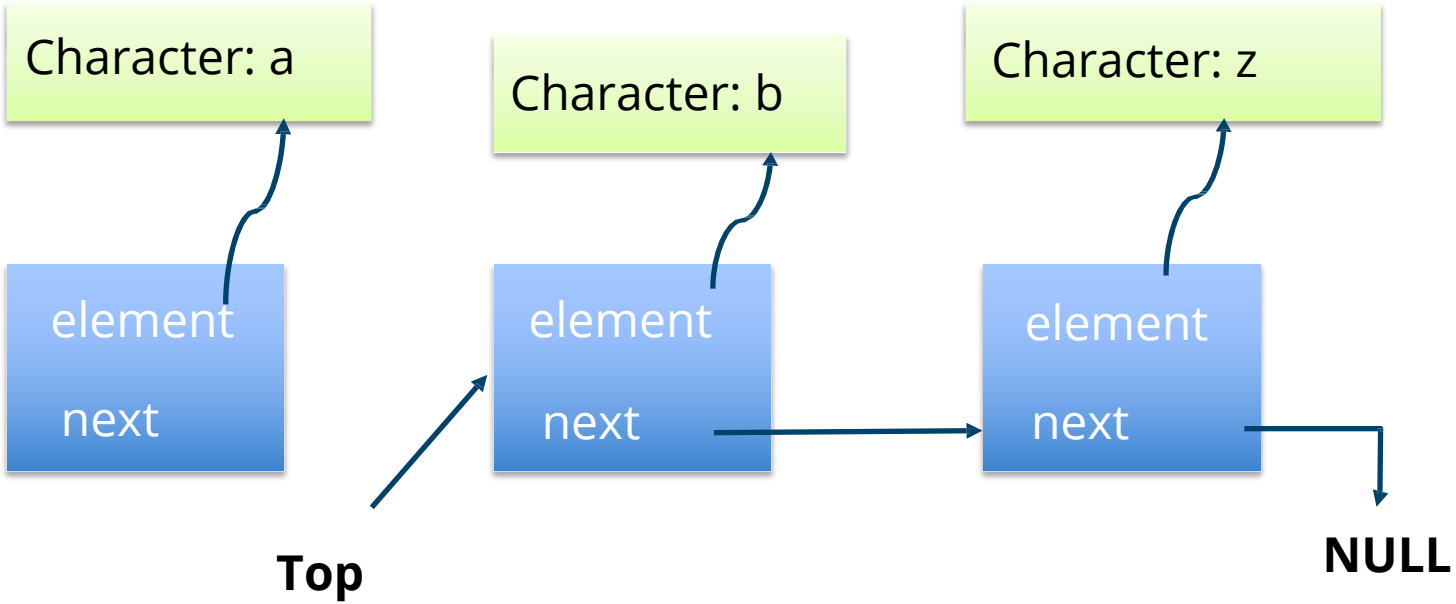
```
/**
 * Represents a linked implementation of a stack.
 */
public class LinkedStack<T> implements StackADT<T>{

    private int count;
    private LinearNode<T> top;
    /**
     * Creates an empty stack.
     */
    public LinkedStack(){
        count = 0;  top =
        null;
    }
```

The typical Stack class contains a declaration of the LinearNode class and private instance variables of type LinearNode for top, along with methods for all of the operations.

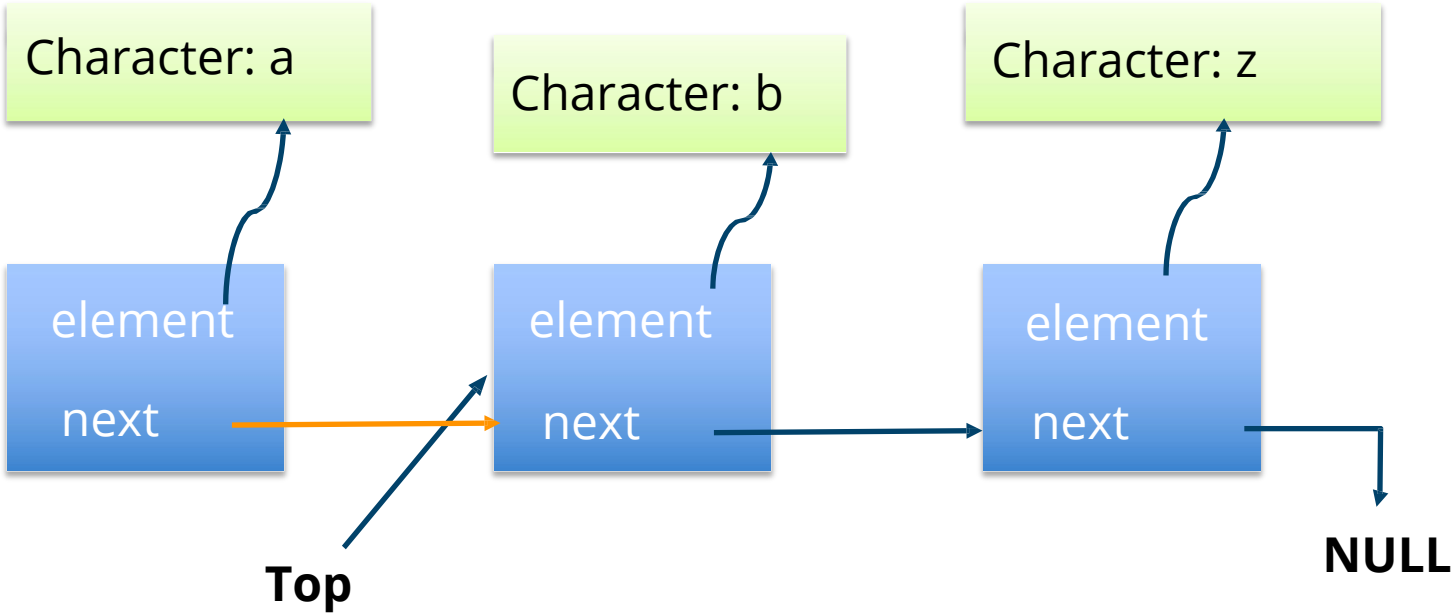
… additional methods: push(E element), pop(), peek(), size(), isEmpty()

public void push(T elem)



Step 1:
Create a new node object and assign its element variable to elem

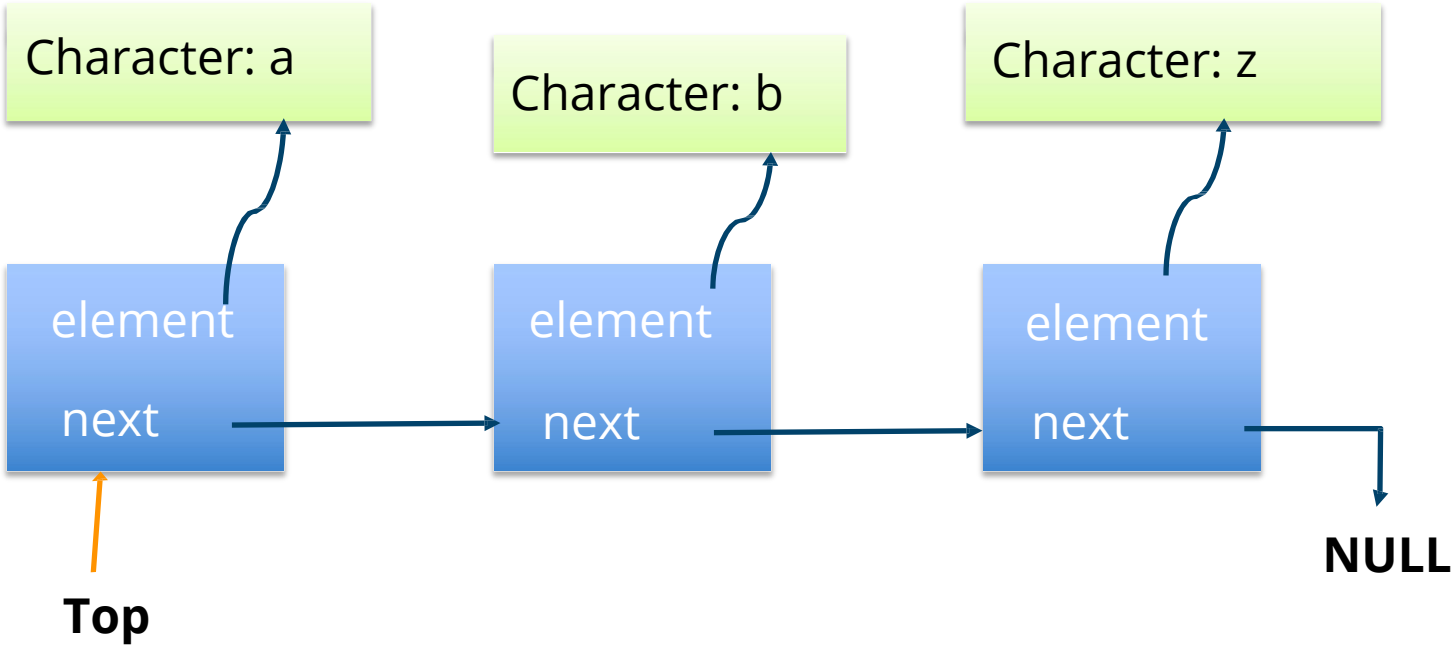
public void push(T elem)



Step 2:

Assign the node's next variable to the top element

public void push(T elem)



Step 3:

Reassign the top variable and increment count

Time Complexity

Method	Linked Node Stack	Array Stack
push(T elem)	O(1)	Amortized O(1)
pop()	O(1)	O(1)
peek()	O(1)	O(1)
isEmpty()	O(1)	O(1)

In-class Activity

LinkedStack Implementation Activity