Northeastern University

Lecture 18: Stack Implementations - 1

Prof. Chen-Hsiang (Jones) Yu, Ph.D.

College of Engineering

Materials are edited by Prof. Jones Yu from

Data Structures and Abstractions with Java, 5th edition. By Frank M. Carrano and Timothy M. Henry.
ISBN-13 978-0-13-483169-5

Outline

- A Linked Implementation
- An Array-Based Implementation
- A Vector-Based Implementation

Outline

- A Linked Implementation
- An Array-Based Implementation
- A Vector-Based Implementation

A Linked Implementation

Stack Implementation

- Each operation involves the top of stack
  - `push`
  - `pop`
  - `peek`
- The head of linked list is easiest, fastest to access
  - Let this be the top of the stack

Linked Implementation

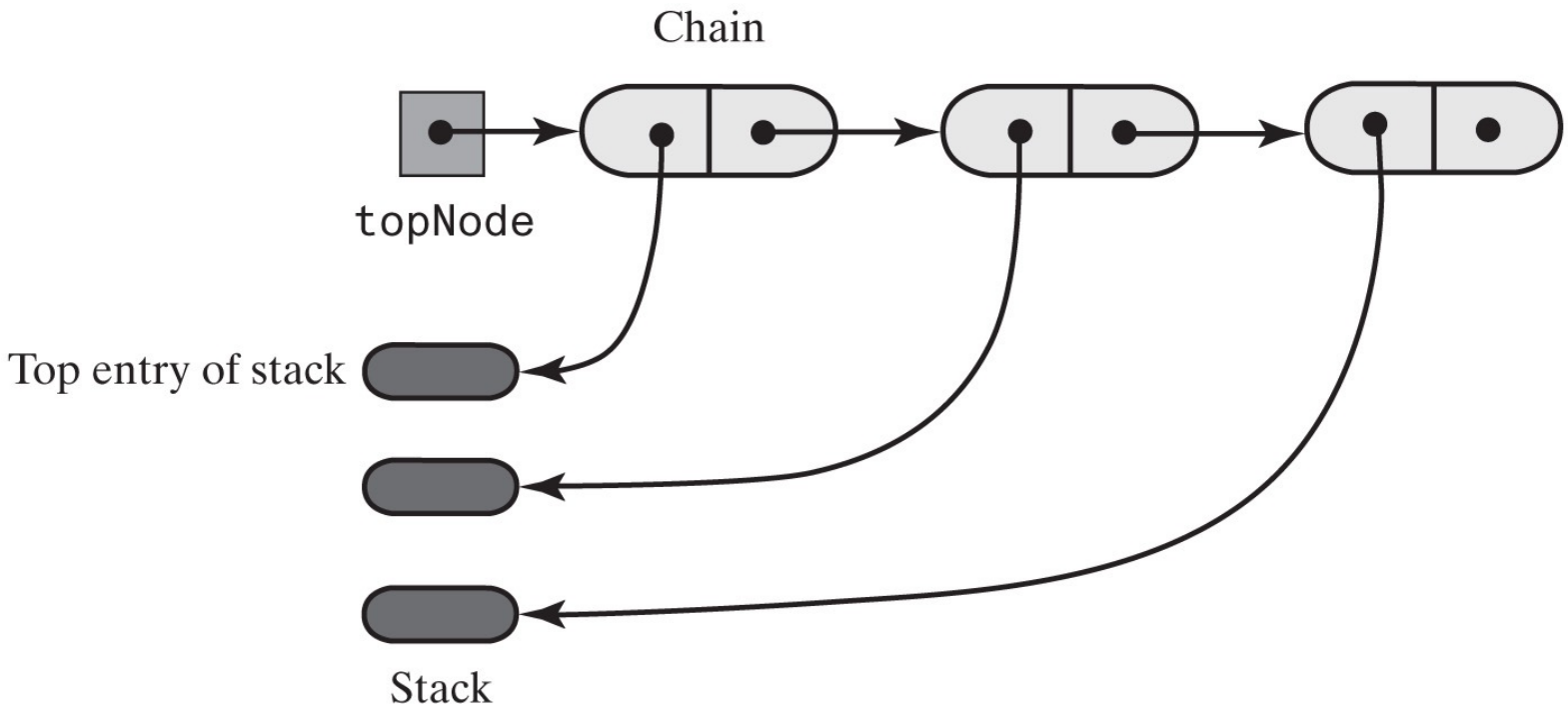


Figure 6-1: A chain of linked nodes that implements a stack

Linked Implementation

```
/** A class of stacks whose entries are stored in a chain of nodes. */
public final class LinkedStack<T> implements StackInterface<T>
{
   private Node topNode; // References the first node in the chain

   public LinkedStack()
   {
      topNode = null;
   } // end default constructor

   //  < Implementations of the stack operations go here. >
   //  . . .

   private class Node
   {
      private T    data; // Entry in stack
      private Node next; // Link to next node
      //  < Implementations of the node operations go here. >

   } // end Node
} // end LinkedStack
```

Listing 6-1: An outline of a linked implementation of the ADT stack

Linked Implementation - Push Operation

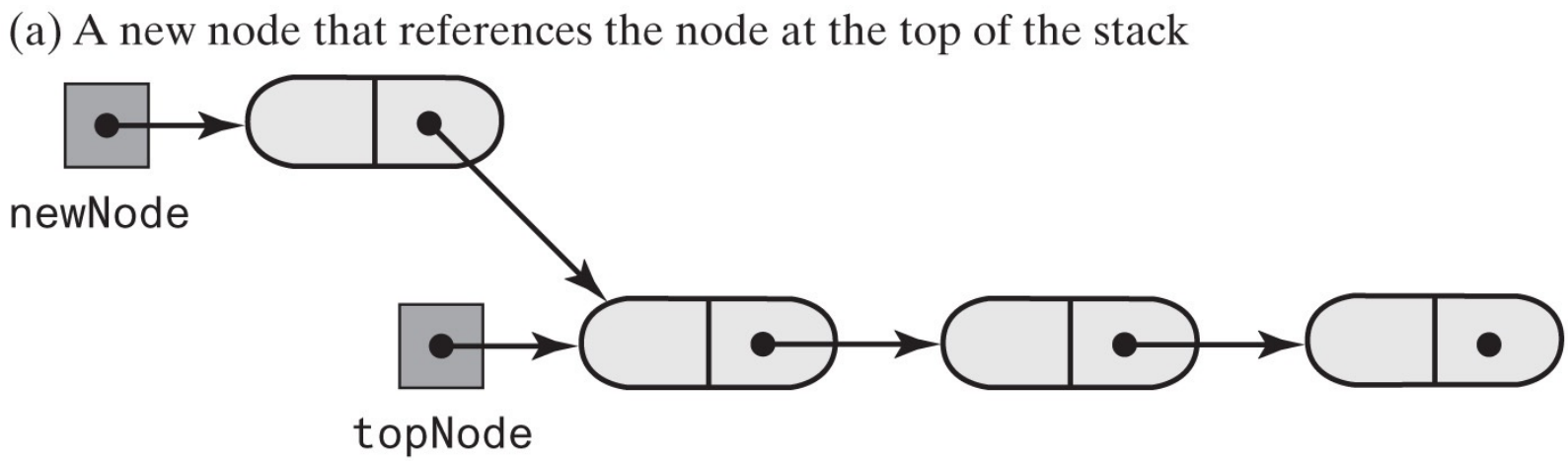


Figure 6-2: (a) A new node that references the node at the top of the stack;

Linked Implementation - Push Operation (cont.)

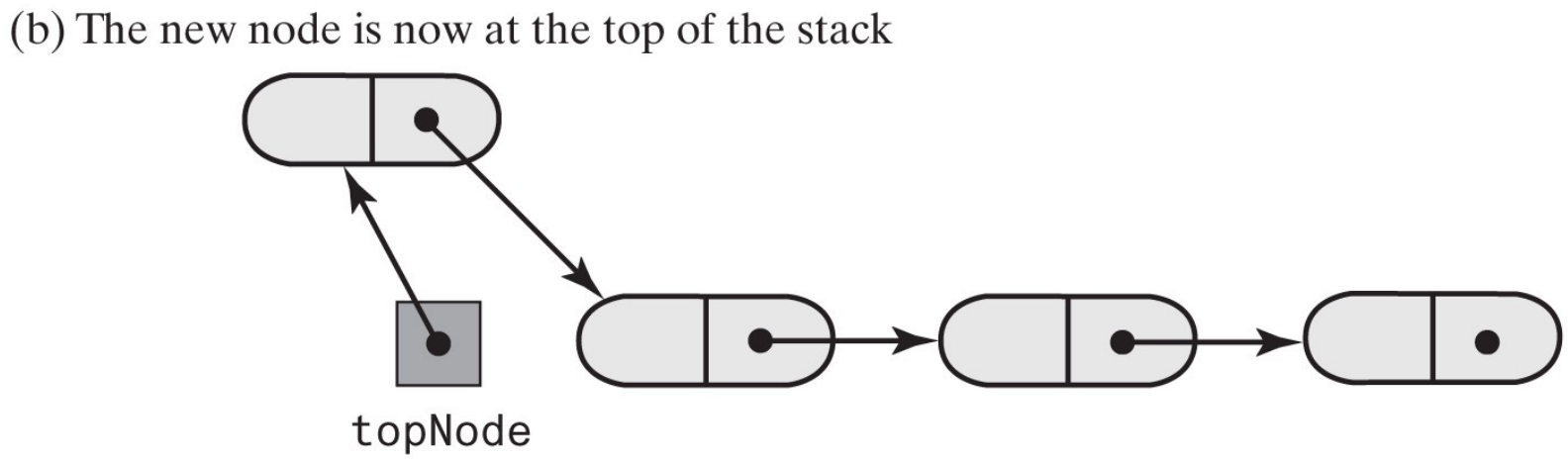
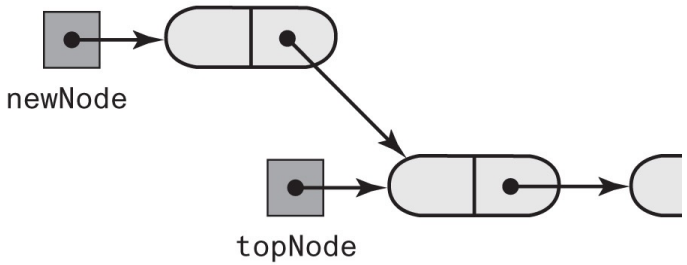


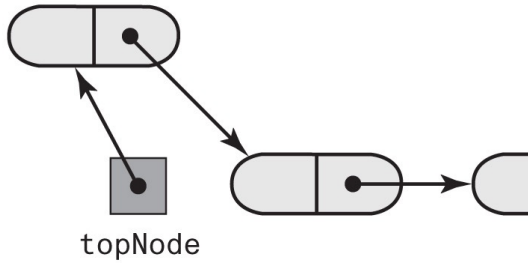
Figure 6-2: (b) the new node is now at the top of the stack

Linked Implementation - Push Operation (cont.)




```
public void push(T newEntry)
{
   Node newNode = new Node(newEntry, topNode);
   topNode = newNode;
} // end push
```

Definition of **`push`**



Linked Implementation - Pop Operation

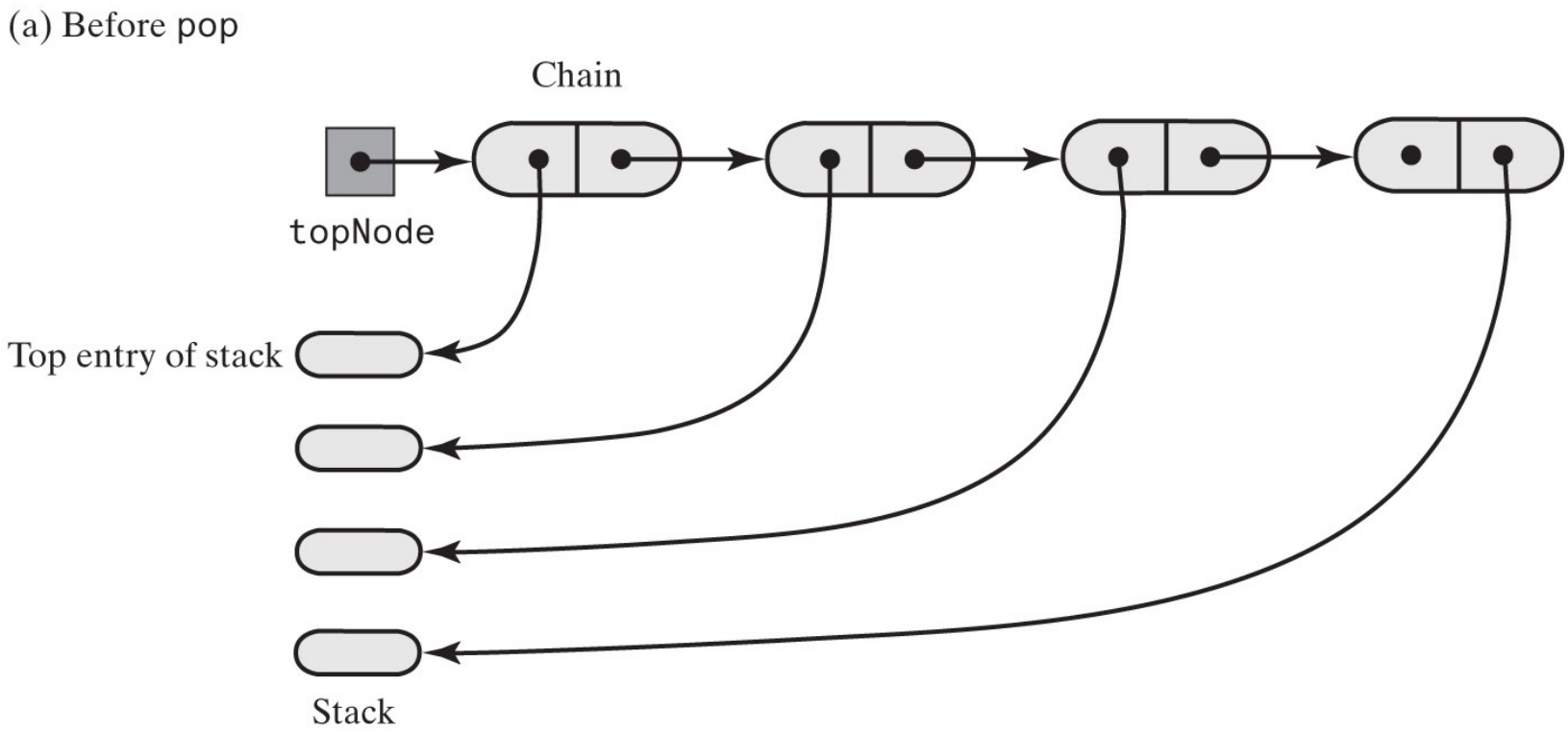


Figure 6-3: The stack (a) before the first node in the chain is deleted

Linked Implementation - Pop Operation (cont.)

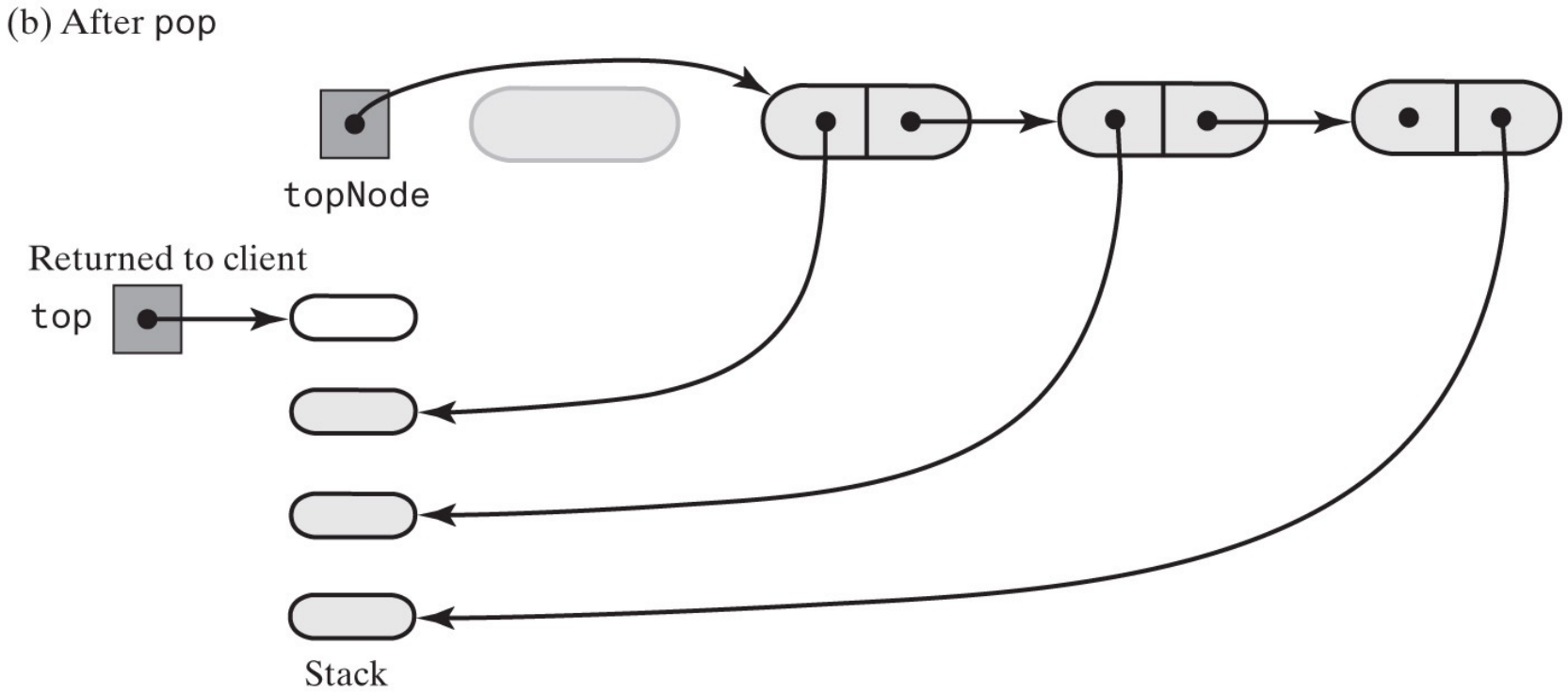


Figure 6-3: The stack (b) after the first node in the chain is deleted

Linked Implementation - Peek & Pop Operations

```
public T peek()
{
   if (isEmpty())
      throw new EmptyStackException();
   else
      return topNode.getData();
} // end peek

public T pop()
{
   T top = peek();  // Might throw EmptyStackException

   // Assertion: topNode != null
   topNode = topNode.getNextNode();

   return top;
} // end pop
```

Definition of `peek` and `pop`

Linked Implementation - isEmpty() & clear()

```
public boolean isEmpty()
{
   return topNode == null;
} // end isEmpty


public void clear()
{
   topNode = null;
} // end clear
```

Definition of **`isEmpty`** and **`clear`**

Exercise

- Download "L18_E1.zip" from the Canvas
- Revise the implementation of `pop()` so that it does not call peek()

```
public T pop()
{
   T top = peek(); // Might throw EmptyStackException
   assert topNode != null;
   topNode = topNode.getNextNode();
   return top;
} // end pop
```

Note: Assertions in Eclipse

- How to enable assertions in Eclipse?
  - » Open the Run Dialog (Run > Run Configurations)
  - » Click on the tab, "(x)= Arguments."
  - » Under the field for "VM arguments," type -ea to enable assertions.
  - » Click on the "Apply" and "Run" button

Answer

```
public T pop(){

  if(topNode != null){
     T top = topNode.getData();
     topNode = topNode.getNextNode();
     numberOfEntries--;
     return top;
  }else{
     throw new EmptyStackException();
  }

}
```

Exercise

- Is an implementation of the ADT stack reasonable if the top of the stack is at the end of a chain of linked nodes instead of its beginning? Explain.

Answer

- No
- Although adding a reference to the chain's last node enable you to access the stack's top entry or push a new entry onto the stack efficiently, it is not enough for pop the stack.
- You also need a reference to next-to-last node to remove the chain's last node.
- Therefore, placing the stack's top entry at the end of the chain is not as efficient or easy to implement as placing it at the beginning.