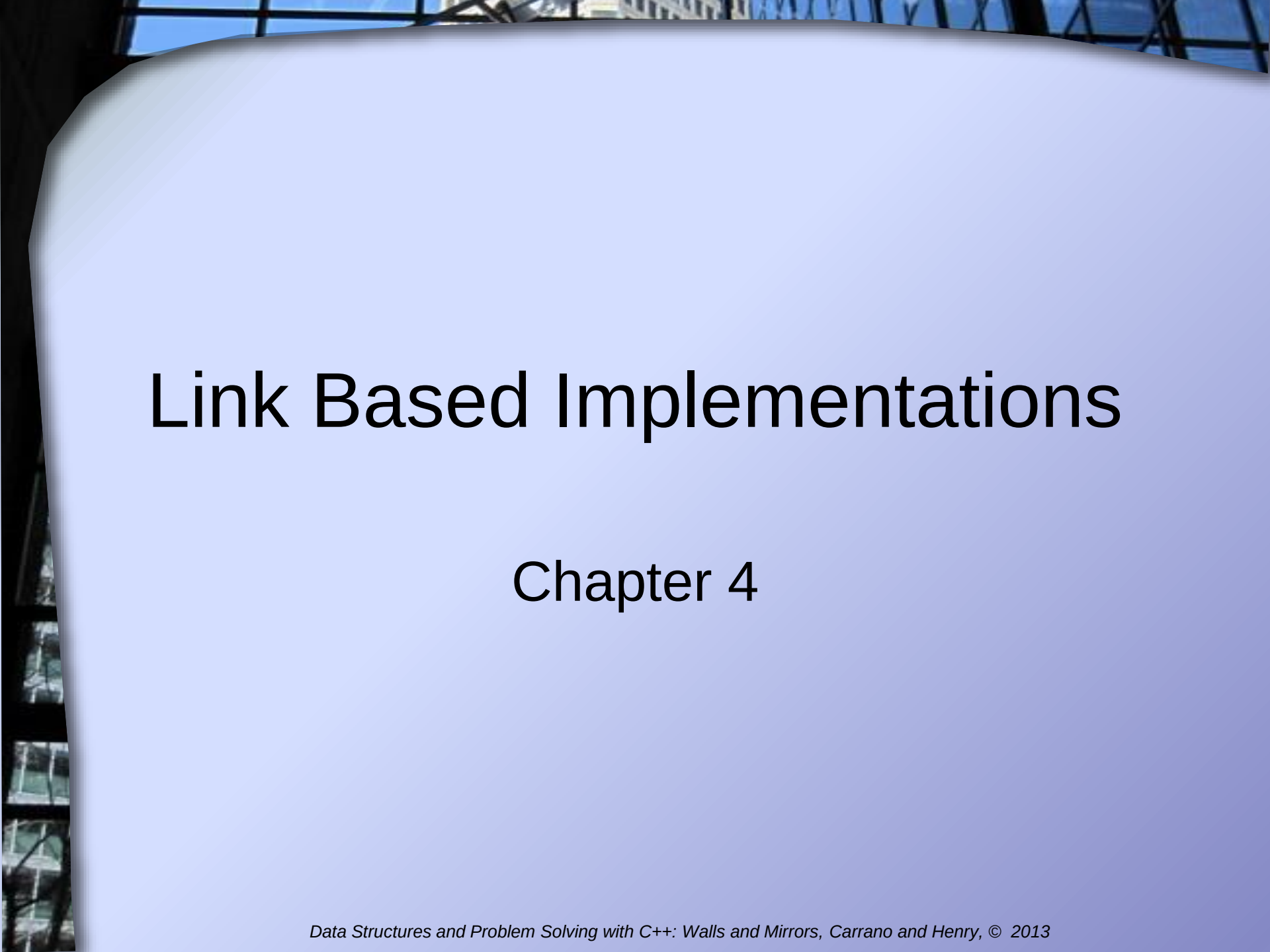


Link Based Implementations

Chapter 4

Contents

- Preliminaries
- A Link-Based Implementation of the ADT Bag
- Using Recursion in Link-Based Implementations
- Testing Multiple ADT Implementations
- Comparing Array-Based and Link-Based Implementations

Preliminaries

- Components that can be linked

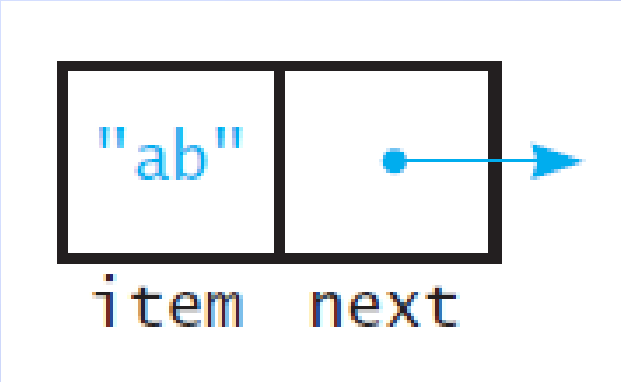


FIGURE 4-1 A node

Preliminaries

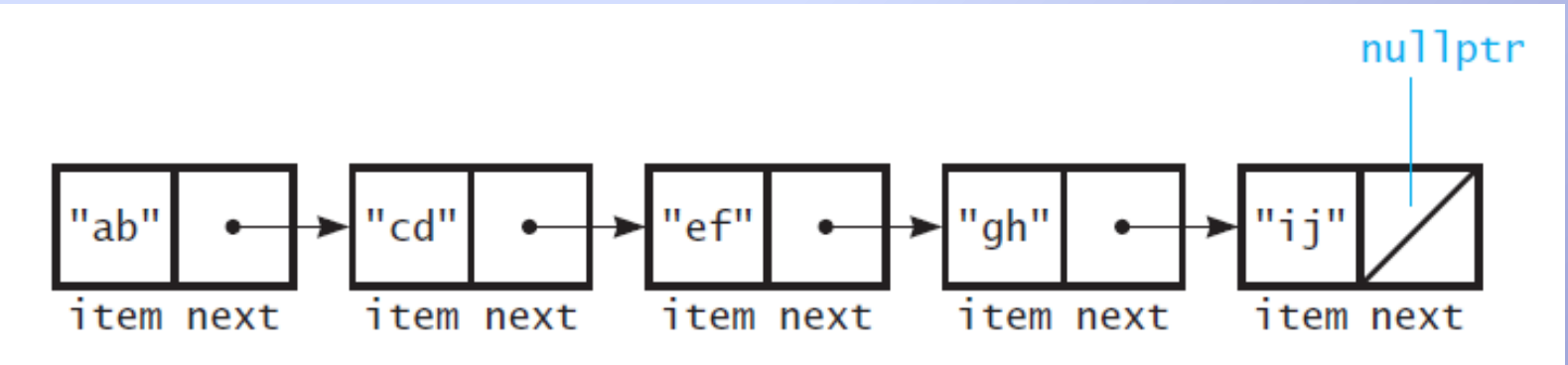


FIGURE 4-2 Several nodes linked together

Preliminaries

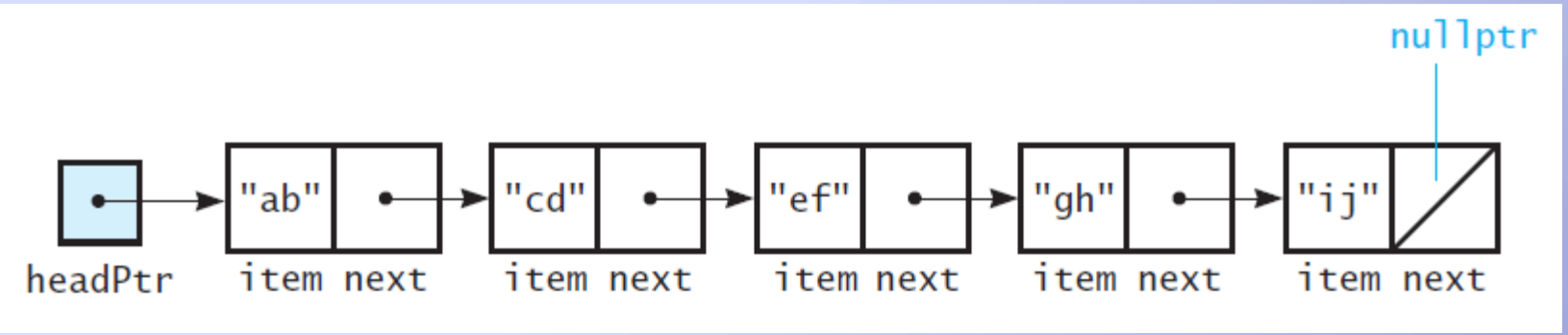


FIGURE 4-3 A head pointer to the first of several linked nodes

Preliminaries

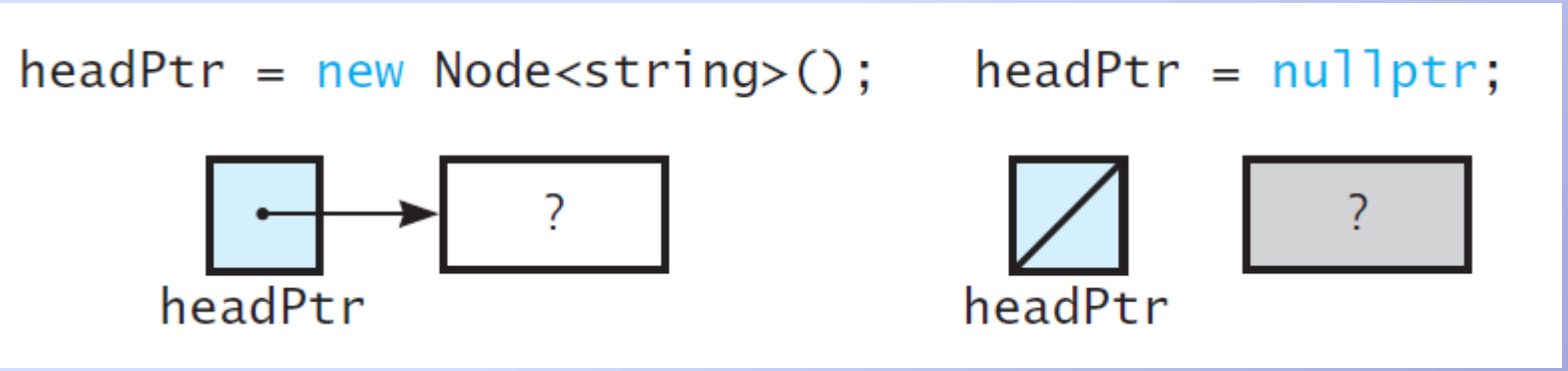


FIGURE 4-4 A lost node

The Class **Node**

- View **Node** header file, Listing 4-1
- Note implementation of class **Node**, Listing 4-2

.htm code listing files must be in the same folder as the .ppt files for these links to work

Link-Based Implementation of ADT Bag

- View header file, Listing 4-3
  - Note methods to be implemented

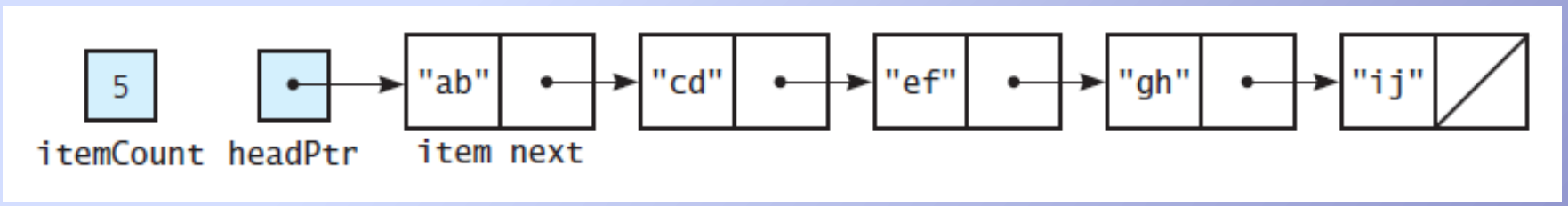


FIGURE 4-5 A link-based implementation of the ADT bag

Link-Based Implementation of ADT Bag

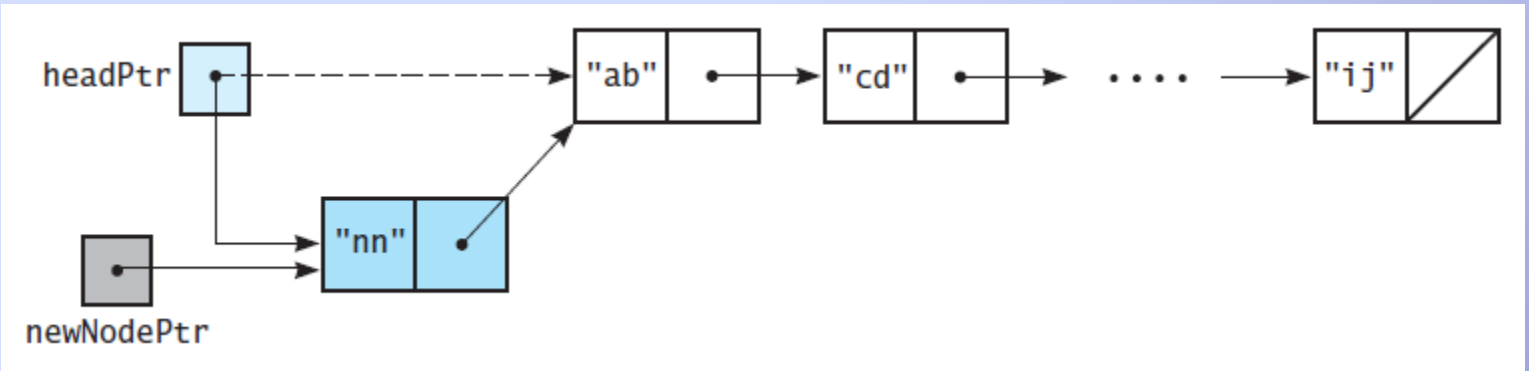


FIGURE 4-6 Inserting at the beginning of a linked chain

Link-Based Implementation of ADT Bag

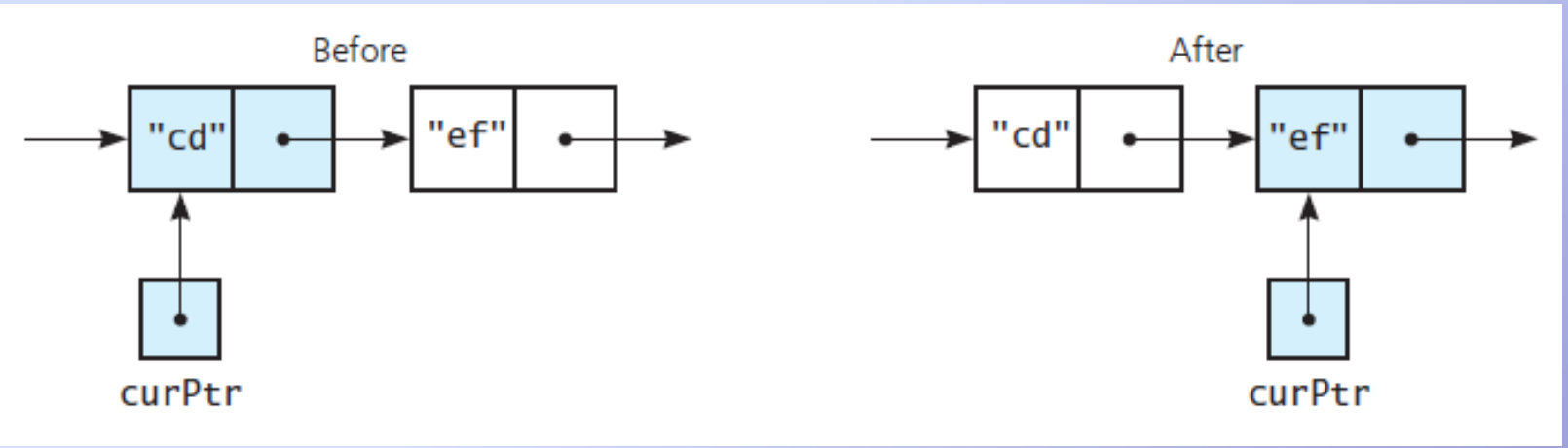


FIGURE 4-7 The effect of the assignment **`curPtr = curPtr->getNext()`**

Link-Based Implementation of ADT Bag

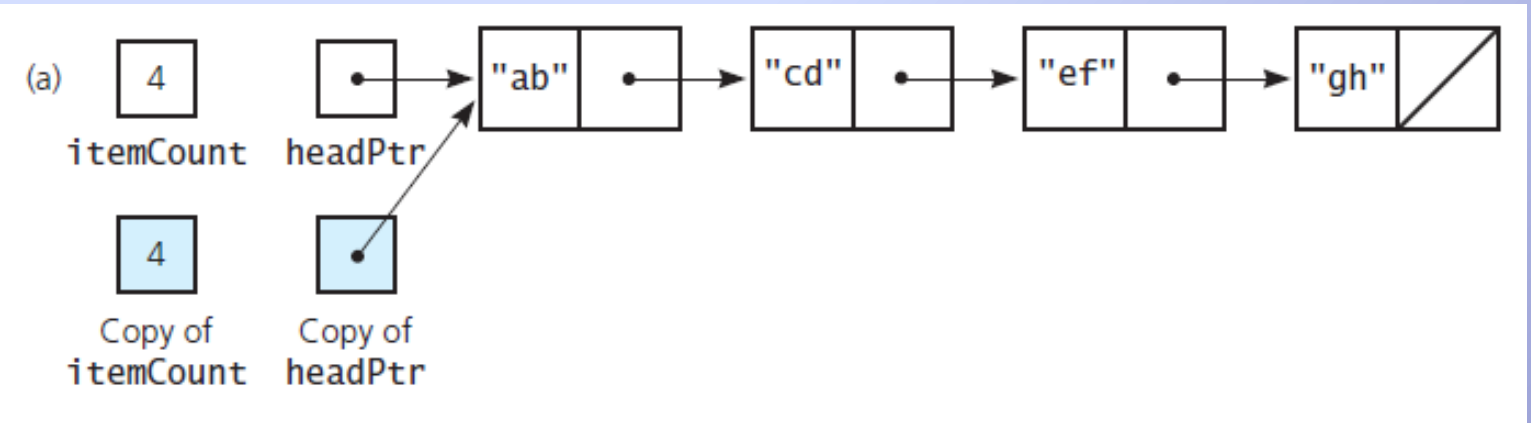


FIGURE 4-8 (a) A linked chain and its shallow copy;

Testing Multiple ADT Implementations

- Note program which tests core methods, Listing 4-4

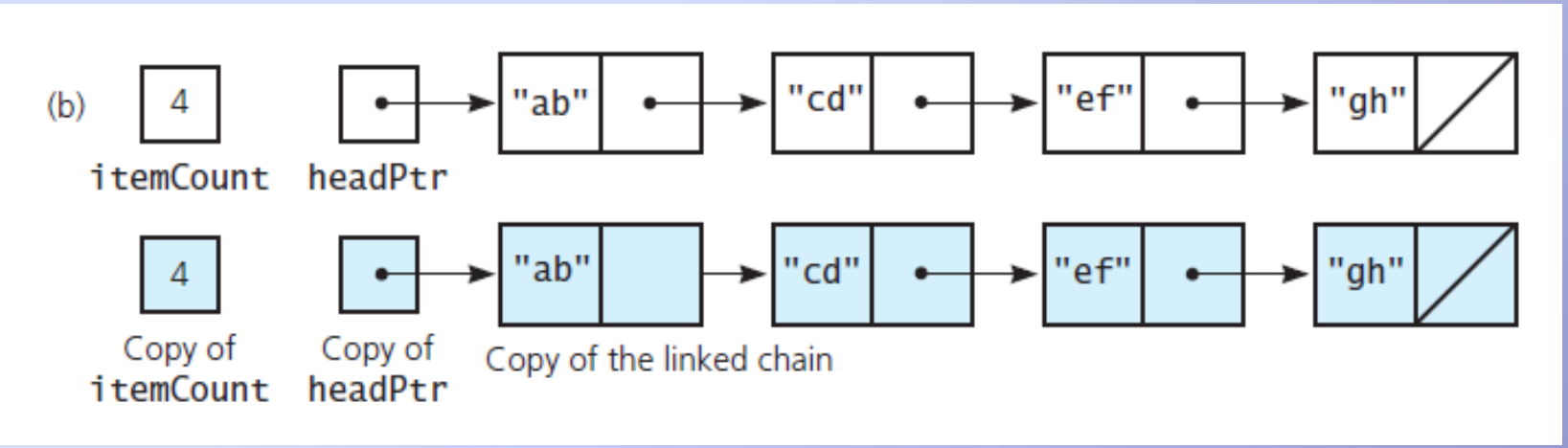


FIGURE 4-8 (b) a linked chain and its deep copy

Testing Multiple ADT Implementations

Sample output 1 of test program

```
Enter 'A' to test the array-based implementation
or 'L' to test the link-based implementation: A
Testing the Array-Based Bag:
The initial bag is empty.
isEmpty: returns 1; should be 1 (true)
Add 6 items to the bag:
The bag contains 6 items:
one two three four five one

isEmpty: returns 0; should be 0 (false)
getCurrentSize returns : 6; should be 6
Try to add another entry: add("extra") returns 0
All done!
```

Testing Multiple ADT Implementations

Sample output 2 of test program

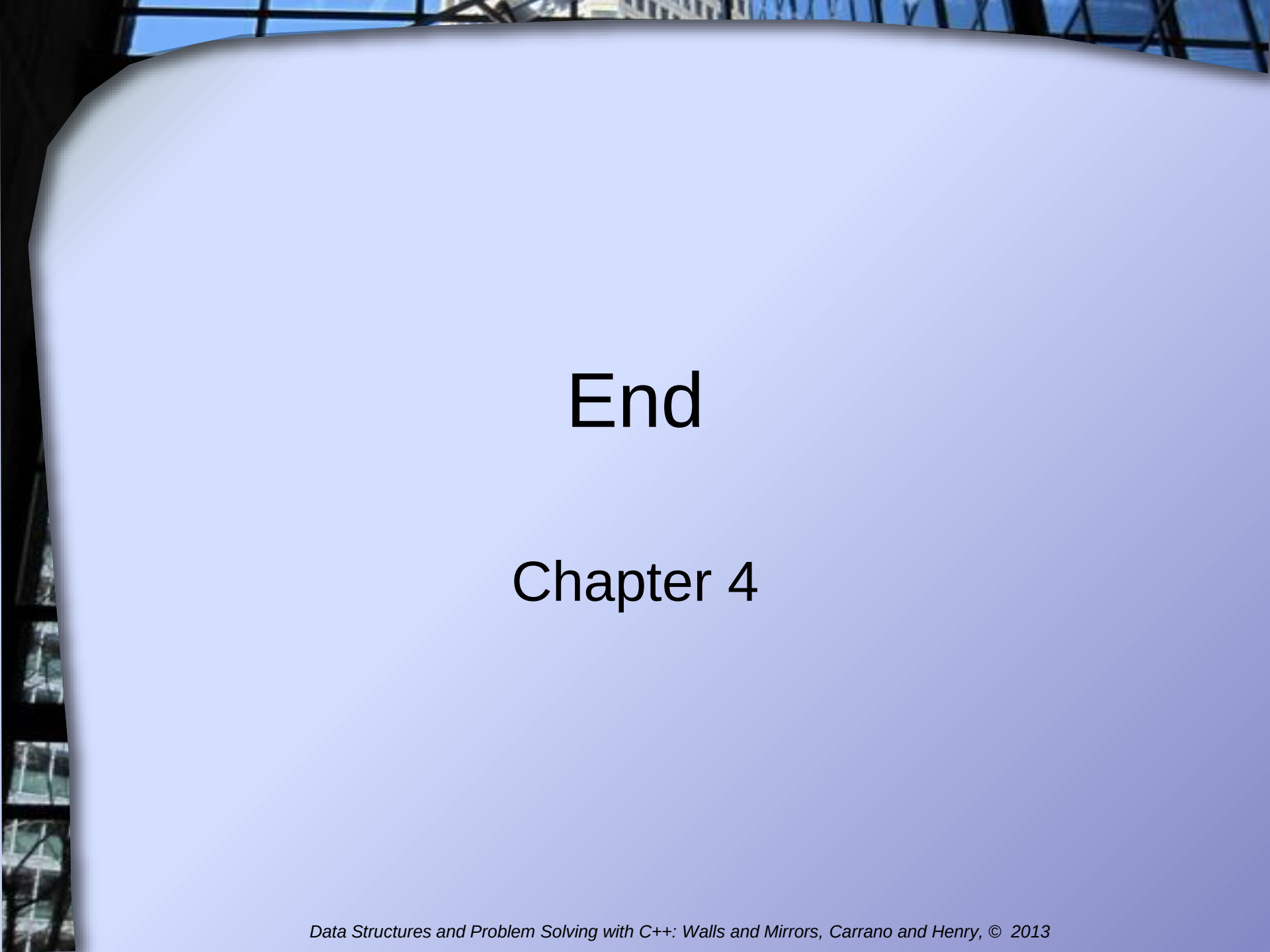
```
Enter 'A' to test the array-based implementation
or 'L' to test the link-based implementation: L
Testing the Link-Based Bag:
The initial bag is empty.
isEmpty: returns 1; should be 1 (true)
Add 6 items to the bag:
The bag contains 6 items:
one five four three two one

isEmpty: returns 0; should be 0 (false)
getCurrentSize returns : 6; should be 6
Try to add another entry: add("extra") returns 1
All done!
```

Comparing Array-Based and Link-Based Implementations

- Arrays easy to use, but have fixed size
- Increasing size of dynamically allocated array can waste storage, time
- Array based implementation good for small bag
- Linked chains do not have fixed size

Comparing Array-Based and Link-Based Implementations

- Item after an array item is *implied*
  - Item in a chain of linked nodes points *explicitly* to next item
- Array based implementation requires less memory
- Array items accessed directly, equal access time
  - Must traverse linked chain for i^{th} item – access time varies



End

Chapter 4