Chapter 17

Linked List

Brooks/Cole
Thomson Learning™

OBJECTIVES

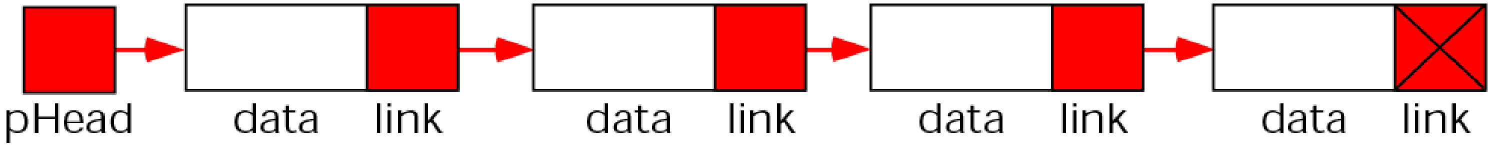
After studying this chapter you will be able to:

- ❑ **Understand and discuss the basic concepts of linked list processing.**
- ❑ **Write functions to insert, delete, search, and traverse a linked list.**
- ❑ **Create a class that maintains and processes data in a linked list.**
- ❑ **Discuss the basic factors involved in developing quality software.**

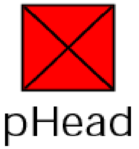
17.1

LINKED LIST STRUCTURE

Figure 17-1 A linked list

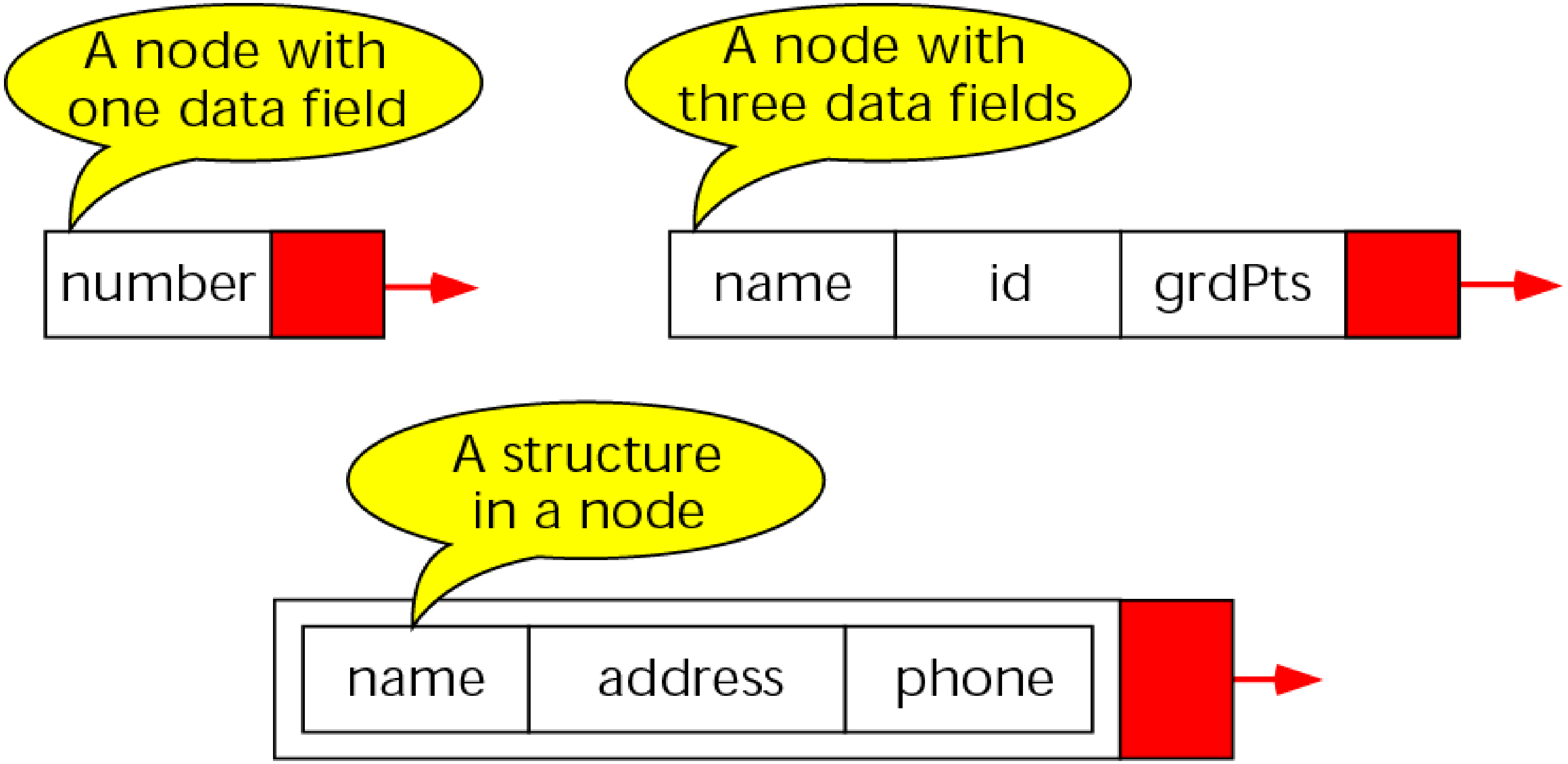


A LINKED LIST WITH A HEAD POINTER *pHead*



AN EMPTY LINKED LIST

Figure 17-2 Nodes

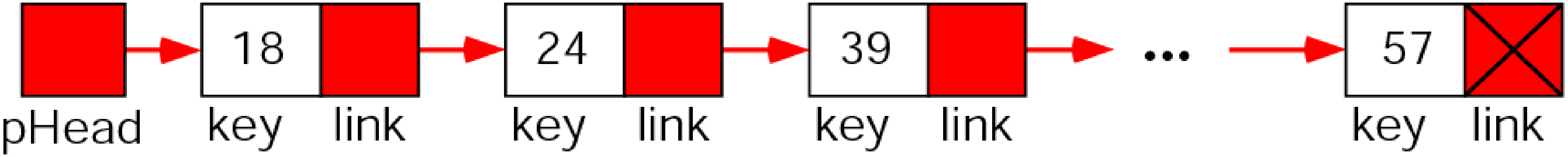


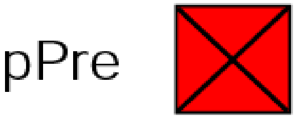
17.2

Basic Linked List Functions

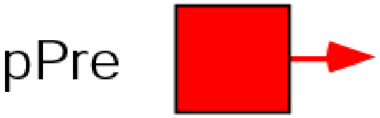
Brooks/Cole
Thomson Learning™

Figure 17-3 Pointer combinations for add





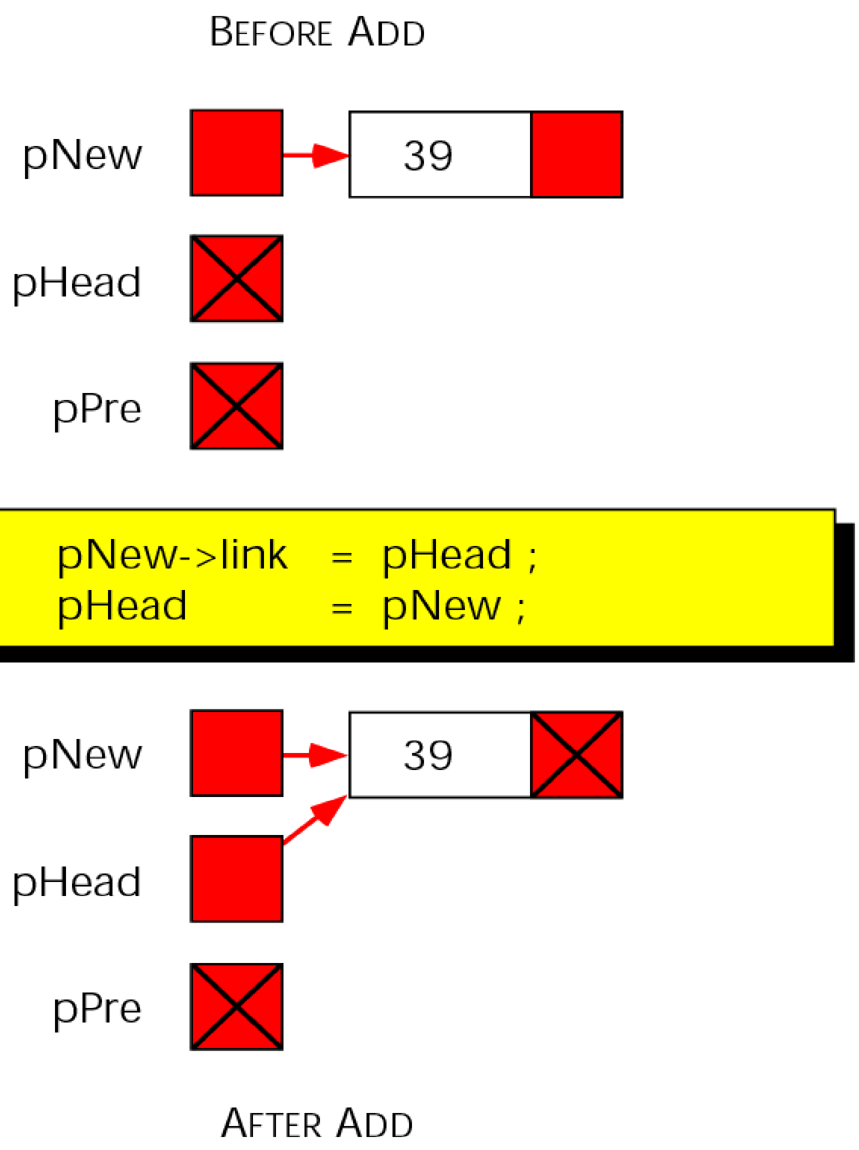
Null (0): Add to empty list or add at beginning of list



Not Null (0): Add in middle of list or add at end of list

Brooks/Cole
Thomson Learning™

Figure 17-4 Add node to empty list



BEFORE ADD

pNew

39

pHead

pPre

```
pNew->link   =  pHead ;
pHead        =  pNew ;
```

pNew

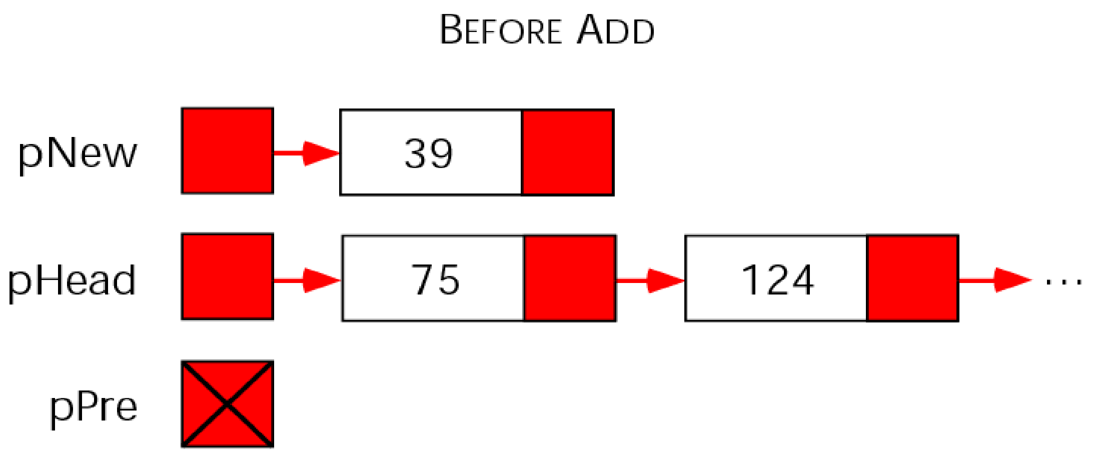
39

pHead

pPre

AFTER ADD

Figure 17-5 Add node at beginning



```
pNew->link  =  pHead ;
pHead       =  pNew ;
```

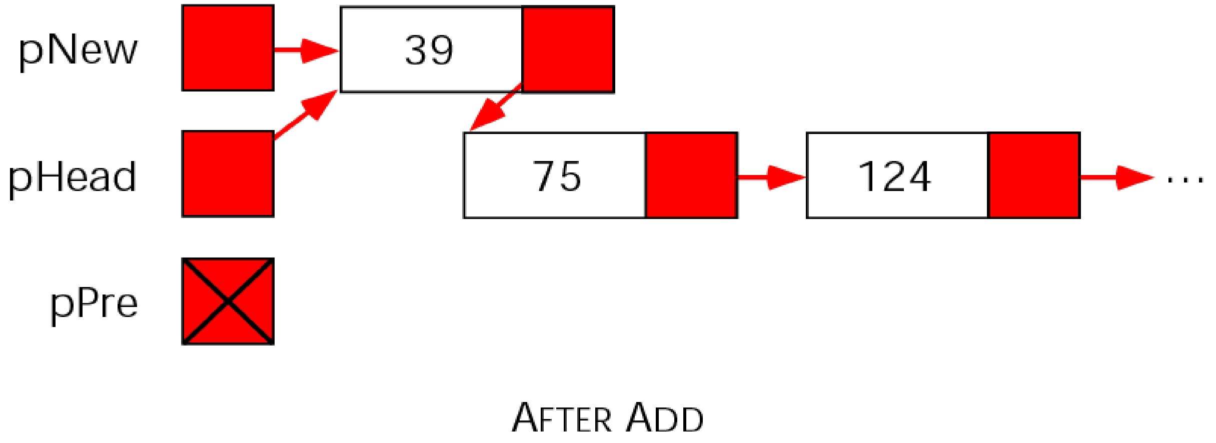
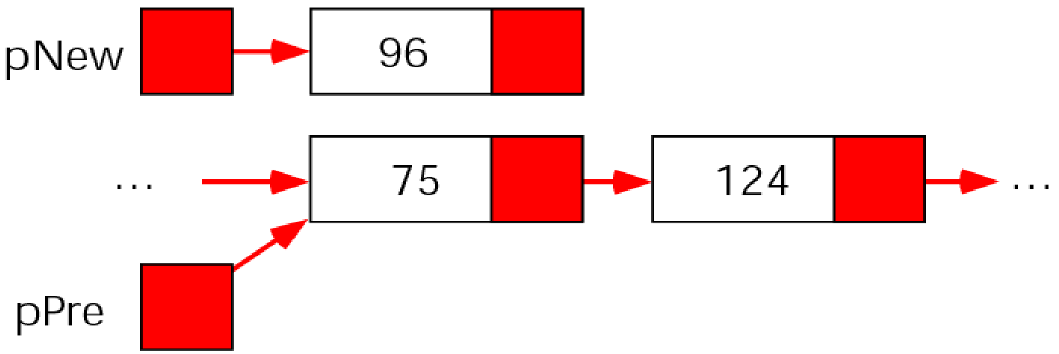
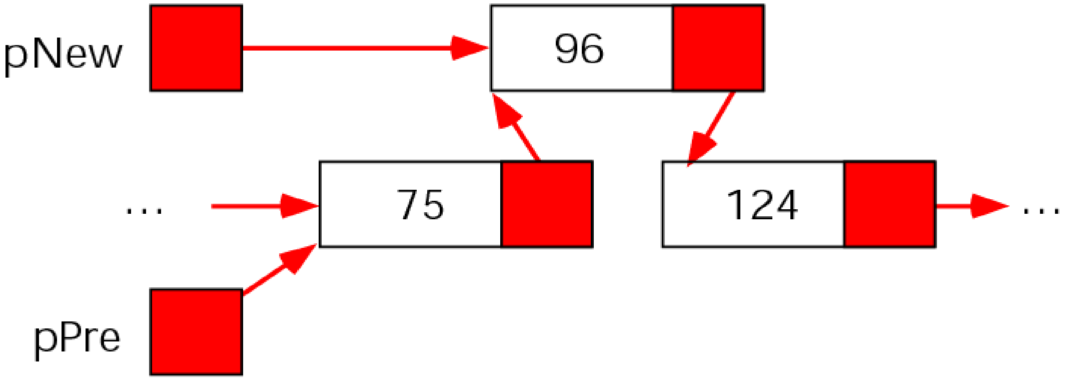


Figure 17-6 Add node in middle

BEFORE ADD



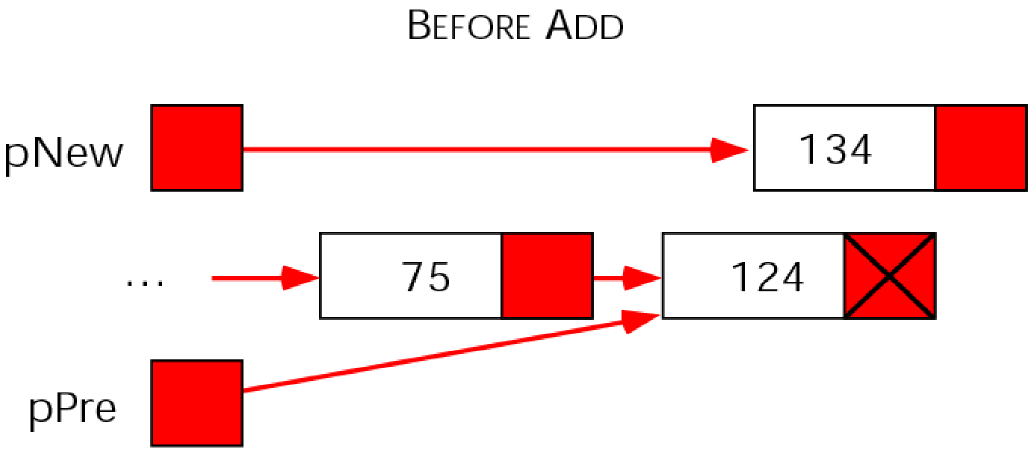
```
pNew->link   =  pPre->link;
pPre->link   =  pNew ;
```



AFTER ADD

Brooks/Cole
Thomson Learning

Figure 17-7 Add node at end



```
pNew->link   =  pPre->link  ;
pPre->link   =  pNew ;
```

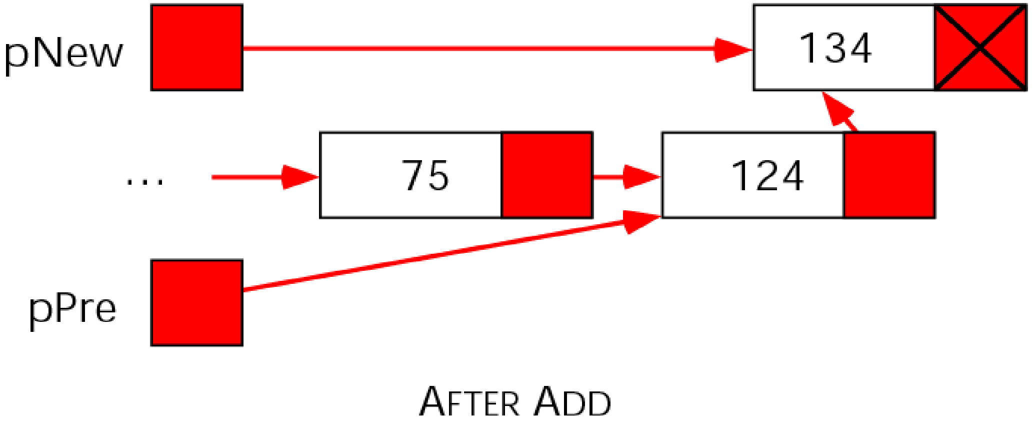
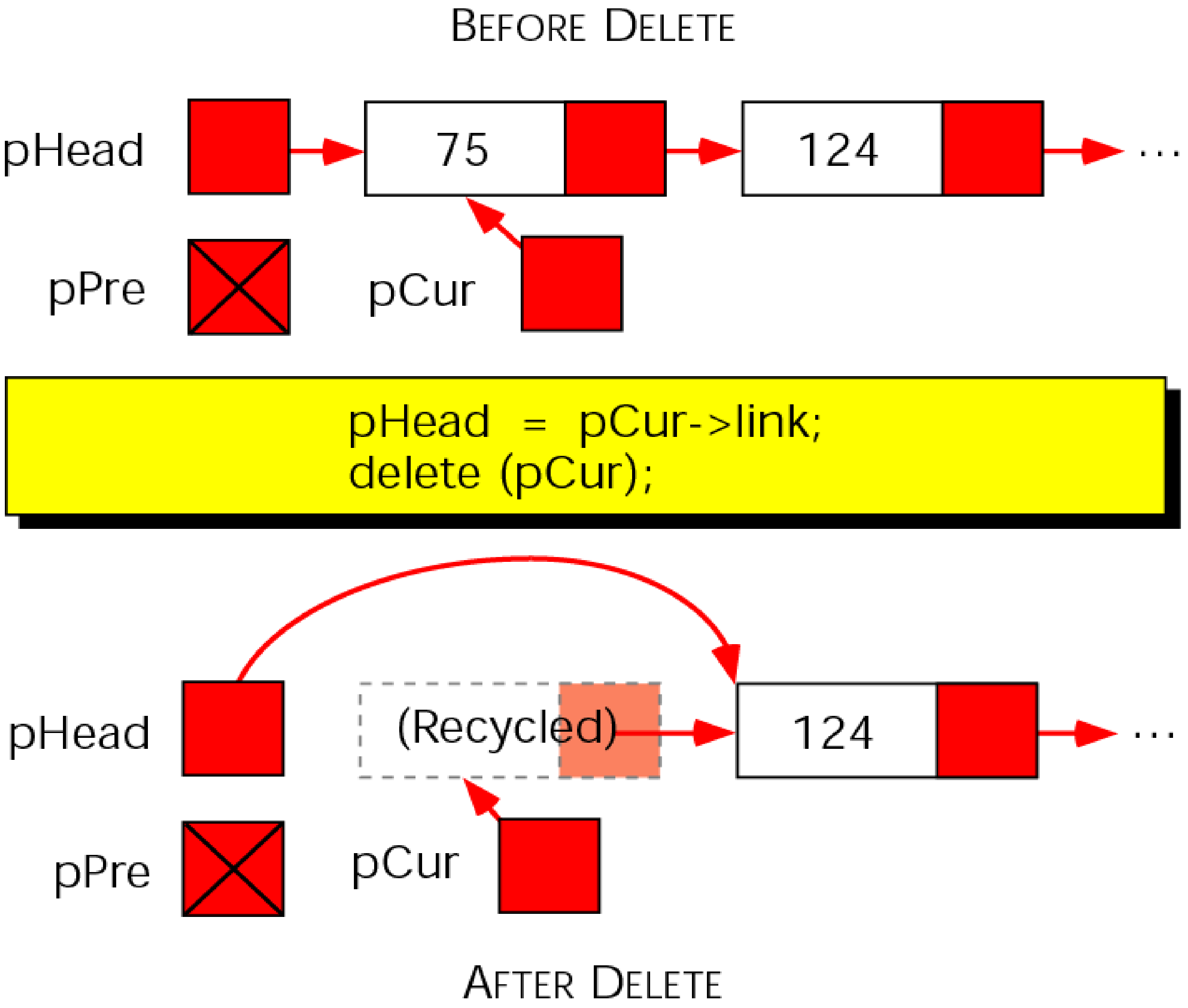
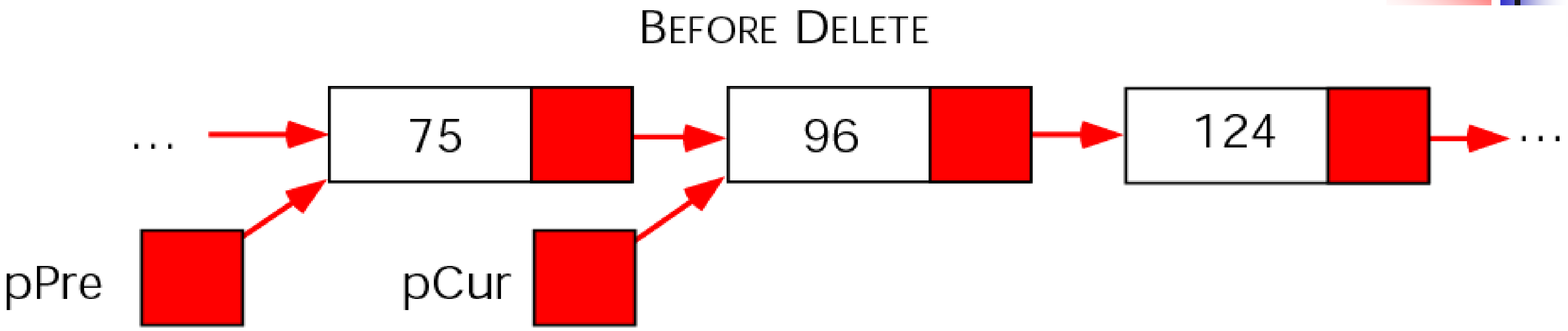


Figure 17-8 Delete first node



```
pHead  =  pCur->link;
delete (pCur);
```

Figure 17-9 Delete—general case



```
pPre->link =  pCur->link ;
delete (pCur);
```

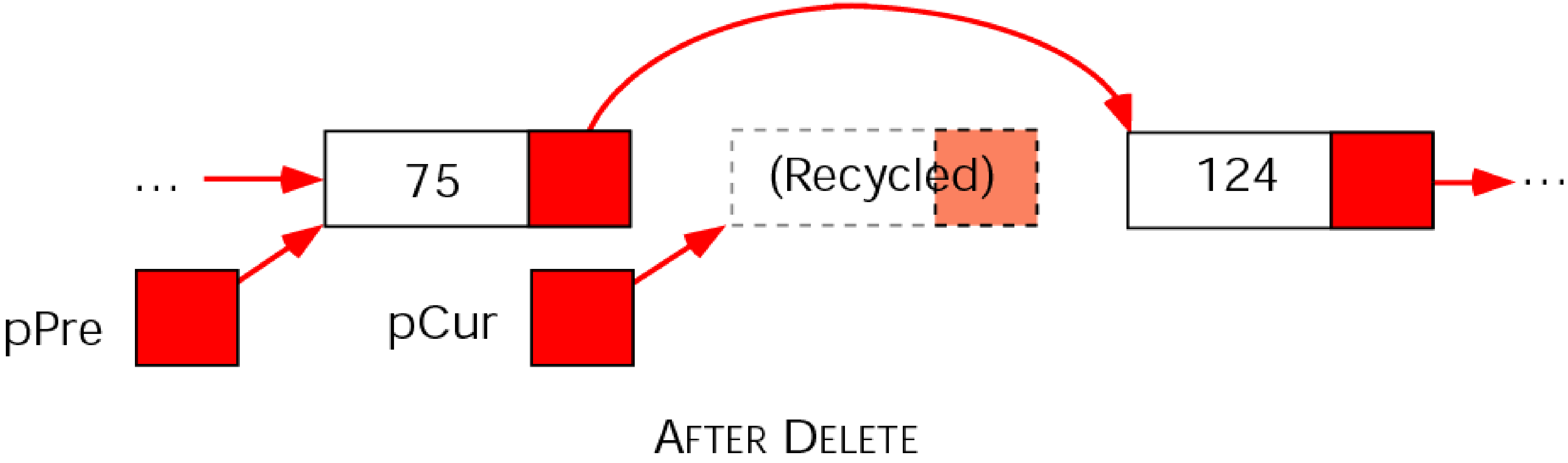
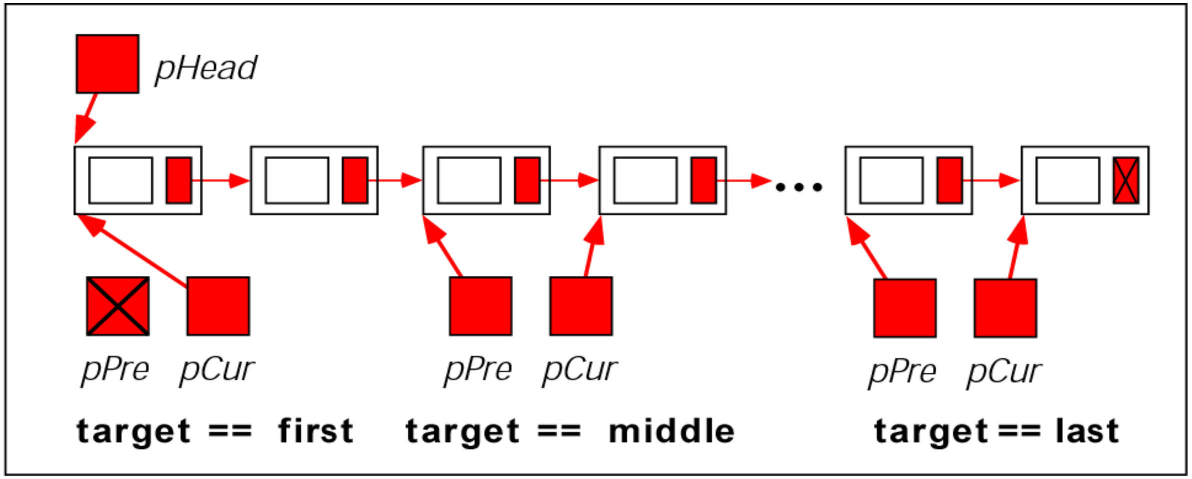
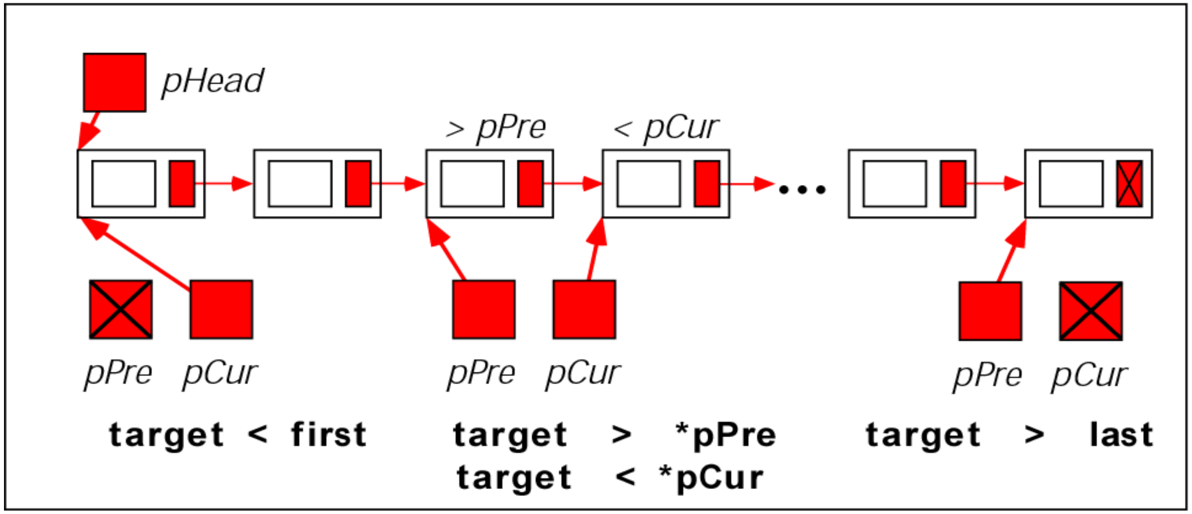


Figure 17-10 Search results

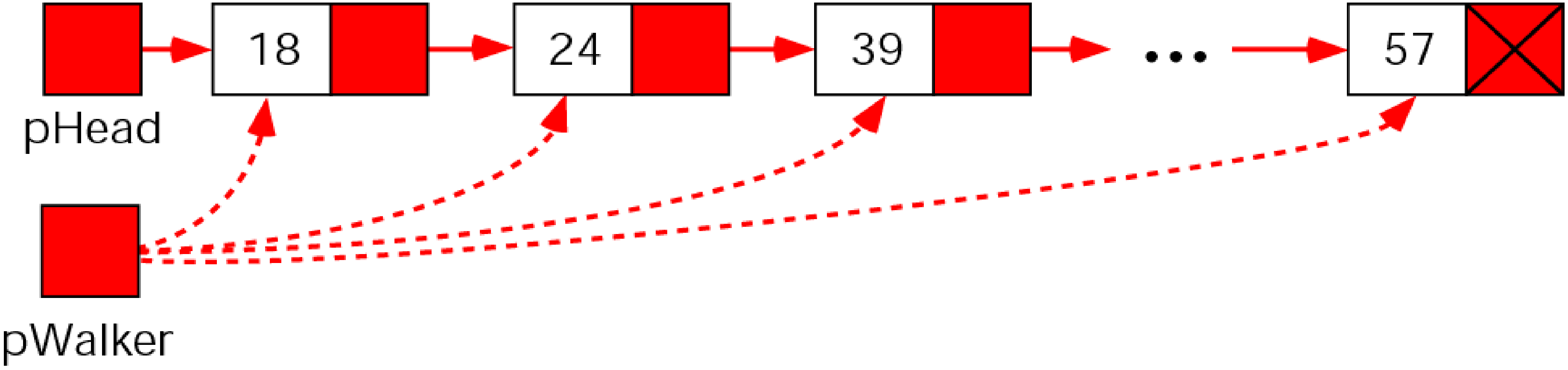


Successful Searches (Return *true*)



Unsuccessful Searches (Return *false*)

Figure 17-11 Linked list traversal



17.3

LINKED LIST DESIGN

Brooks/Cole
Thomson Learning™

Figure 17-12 List and node interrelationships

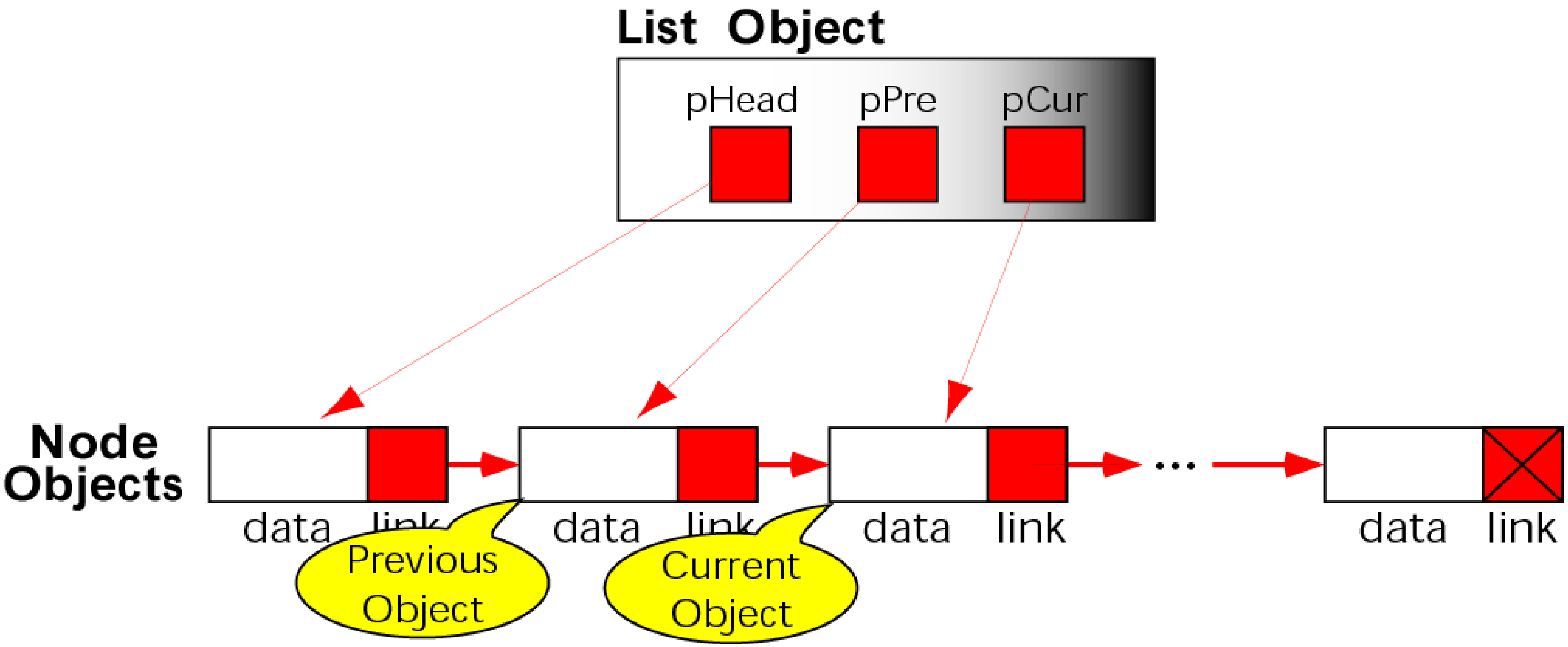


Figure 17-13 Design for *addToList*

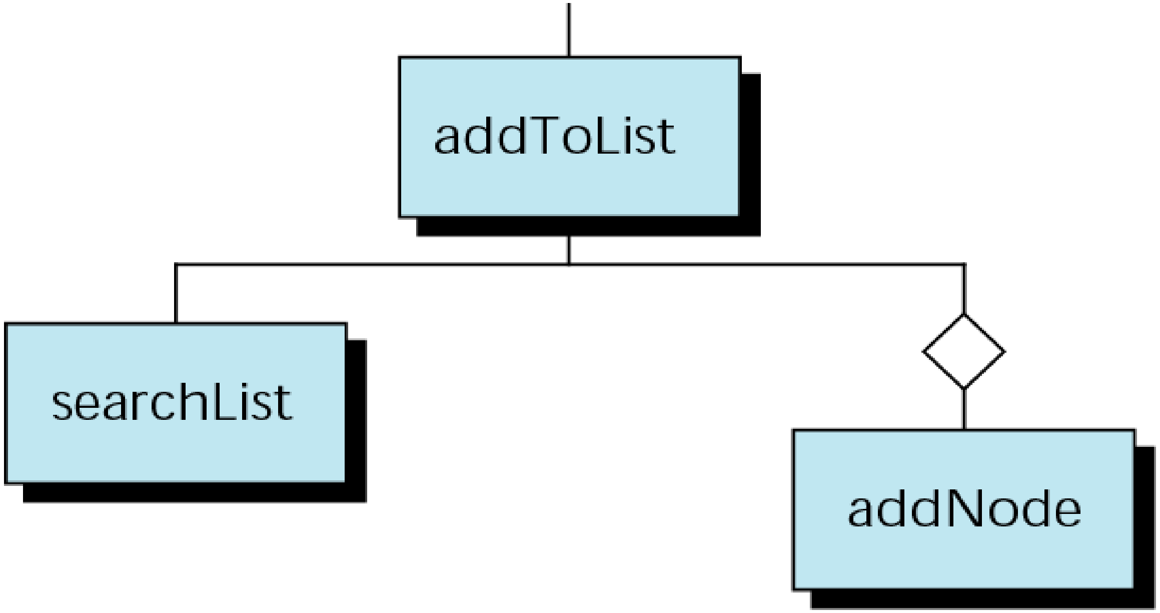


Figure 17-14 Design for *deleteFromList*

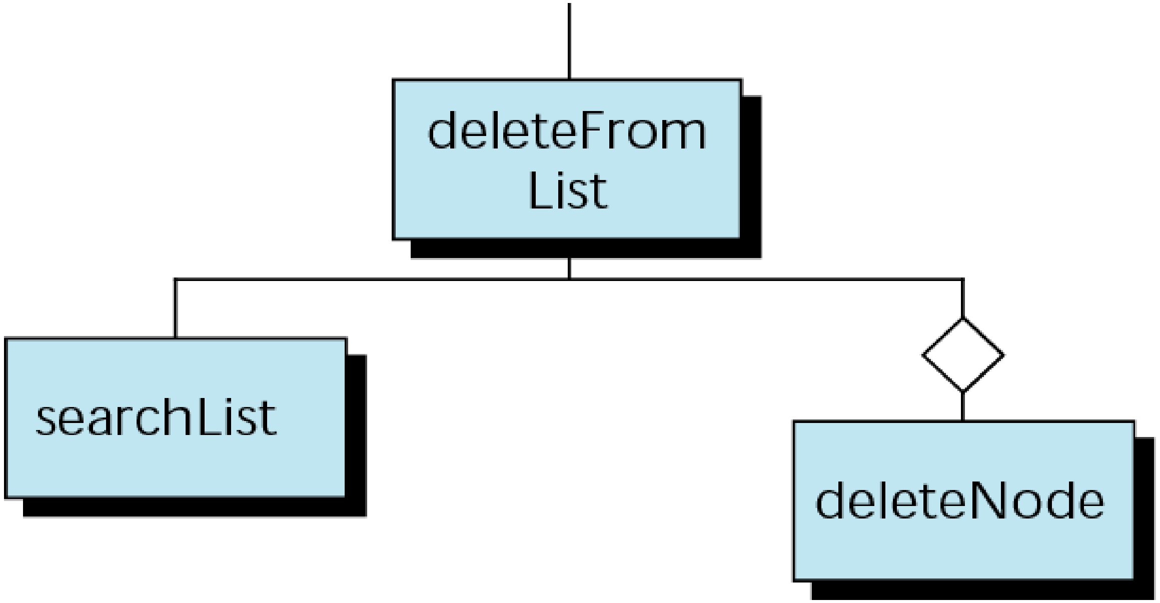
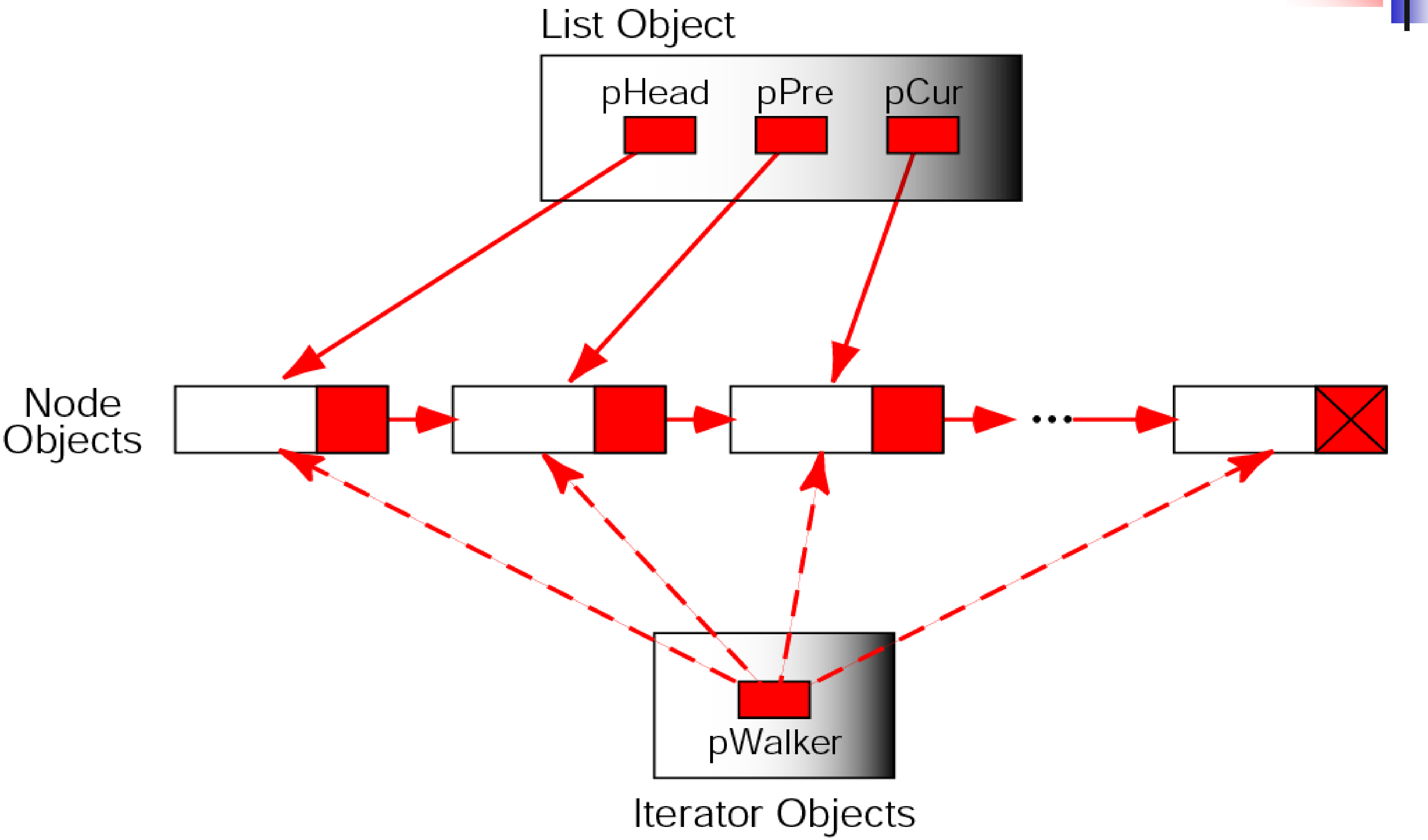


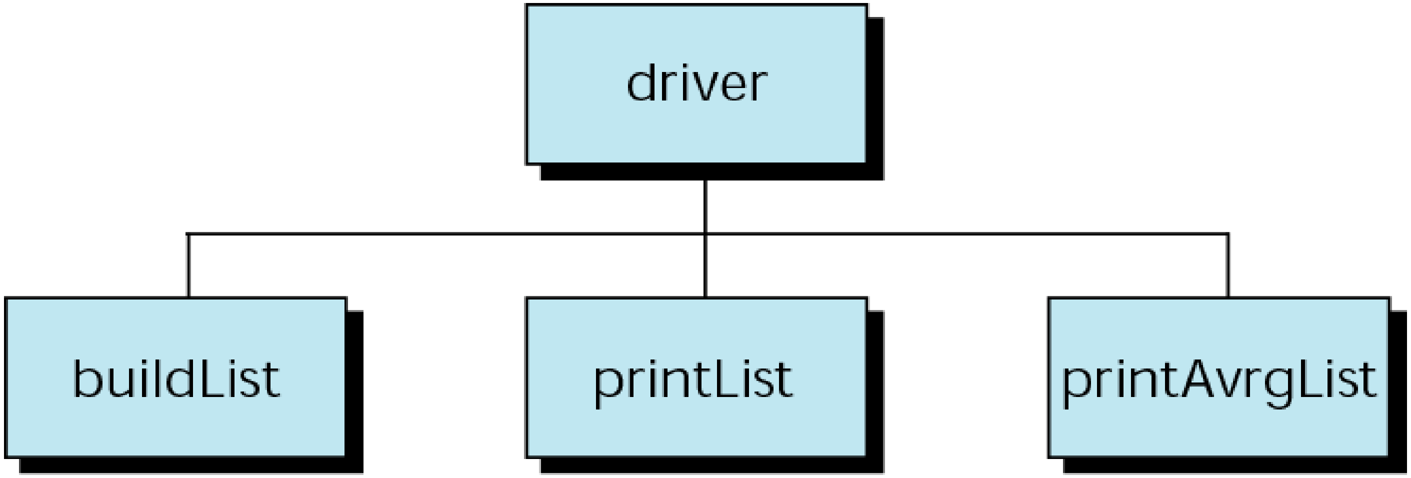
Figure 17-15 Iterator, list, and node objects



17.4

Programming Example— Linked List Average

Figure 17-16 Design for linked list average



17.5

Software Engineering and Programming Style

Brooks/Cole
Thomson Learning™

Note:

Software that satisfies the users' explicit and implicit requirements, is well documented, meets the operating standards of the organization, and runs efficiently on the hardware for which it was developed.

Figure 17-17 Software quality

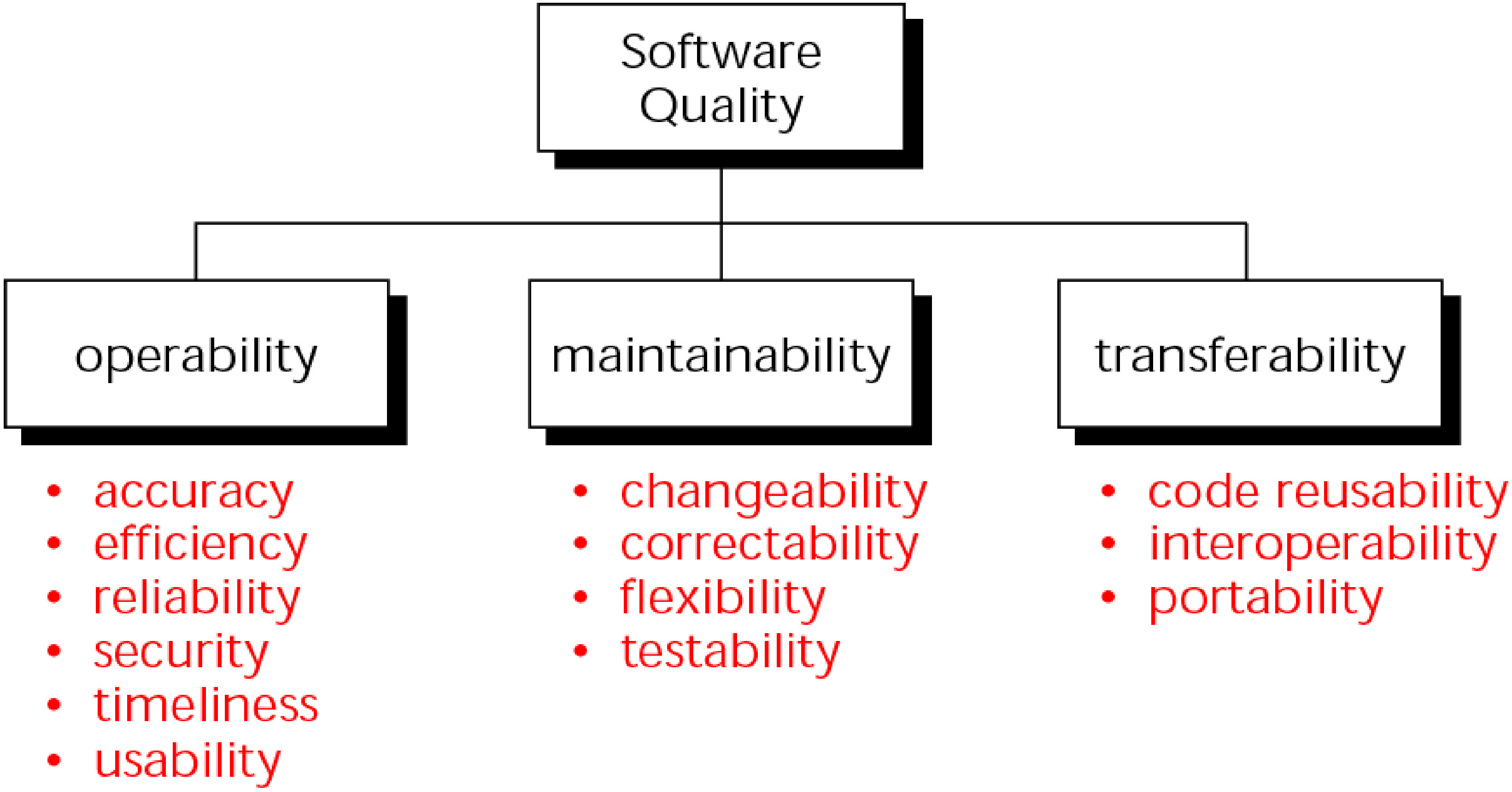
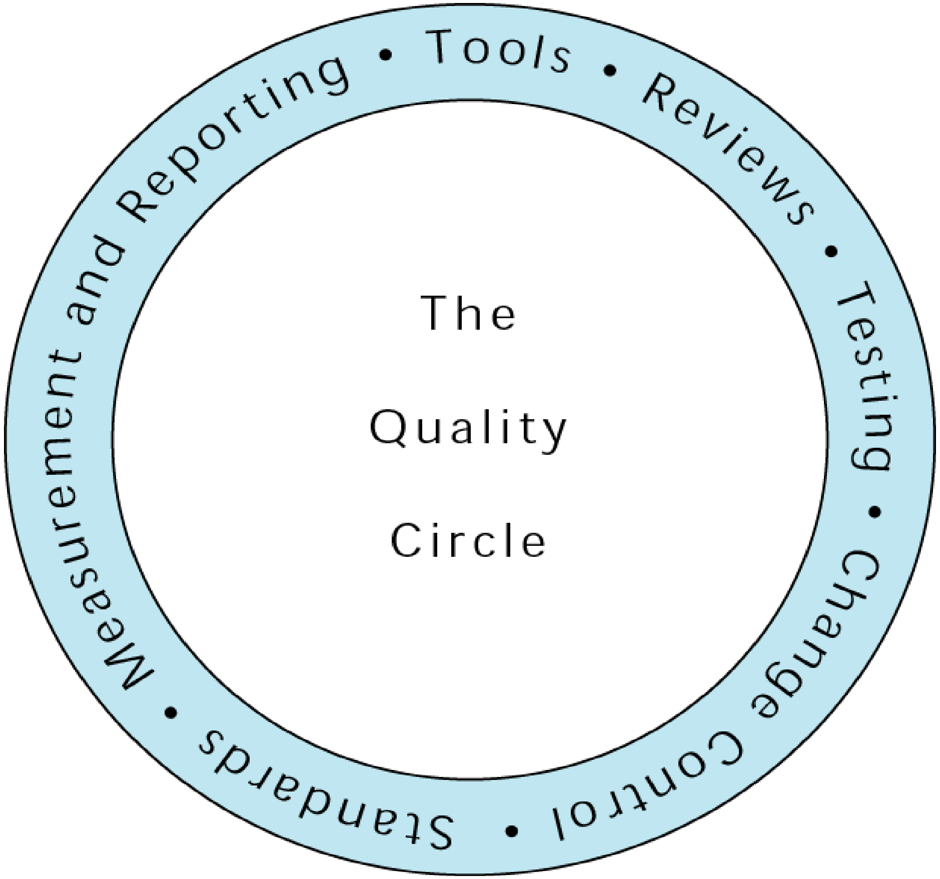


Figure 17-18 Software quality


Tools
Reviews
Testing
Change Control
Standards
Measurement and Reporting
The
Quality
Circle