Add

Figure 7.6a | The first step in inserting "calm" at the front of the SinglyLinkedList object of Figure 7.4: constructing a new Entry object (whose two fields are automatically preinitialized to **null**).

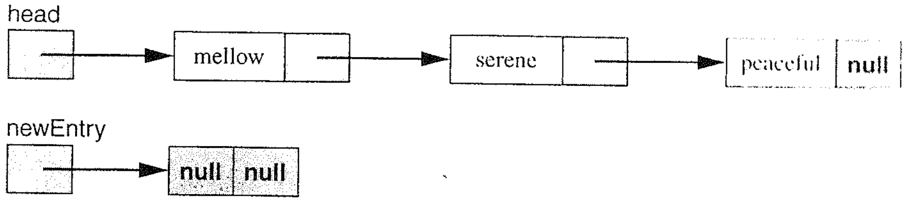


Figure 7.6b | The second step in inserting "calm" at the front of the SinglyLinkedList object of Figure 7.4: assigning the object-reference element to the element field of the newEntry object.

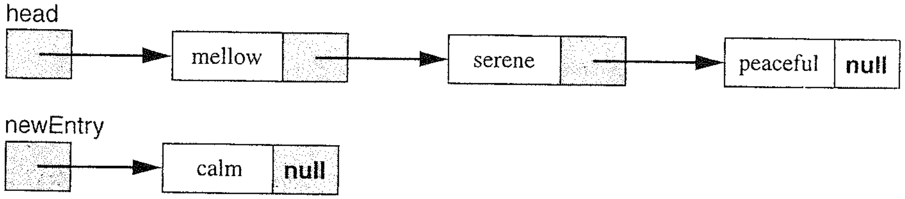


Figure 7.6c | The third step in inserting "calm" at the front of the SinglyLinkedList object of Figure 7.4: assigning head to the next field of the newEntry object.

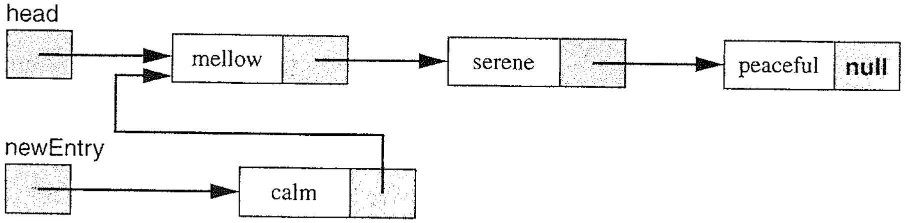
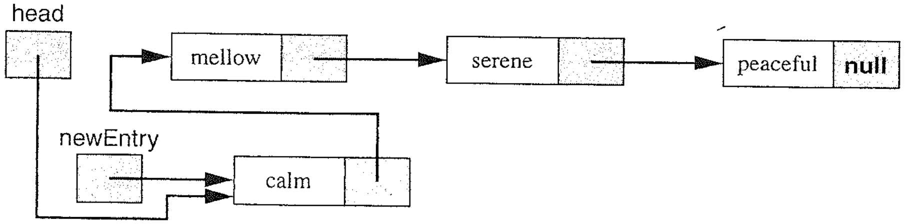


Figure 7.6d | The fourth step in inserting "calm" at the front of the SinglyLinkedList object of Figure 7.4. The SinglyLinkedList object is now as shown in Figure 7.5.



Here is the definition of the size method

```
public int size ( )
{
    int count = 0;

    for (Entry<E> current = head; current !=
        count ++;
    return count;
} // method size
```

The loop goes through the entire SinglyLinke linear in n (as is averageTime(n)). Note that if LinkedList class, the definition of the size( ) meth

```
return size;
```

But then the definition of the add method would value of the size field.

Finally, for now, we develop the contains me the one in the definition of the size method, exce to current.element. And because **null** elements object, we need a separate loop for the case wher

```
public boolean contains (Object obj)
{
    if (obj == null)
    {
        for (Entry<E> current = head; cu
            if (current.element == null
                return true;
    } // if obj == null
    else if (obj instanceof E)
        for (Entry<E> current = head; cu
            if (obj.equals (current.eleme
                return true;
    return false;
} // method contains
```

As we discussed in Section 7.2, in the note for contains (Object obj), make sure that the def element's class compares elements for equality == **null** because the message obj.equals (curre Exception if obj is **null**. And of course, if obj is r be returned. We had to put the **instanceof** test af **instanceof** E returns **false**.

One important point of the SinglyLinkedLis storing a collection of elements is different from key respects: