# Tutorial Quiz #4 — Solutions

Consider the following `Node` class that can be used to store singly-linked lists of `Objects`.

```
class Node {
    public Object data;
    public Node next;
    public Node(Object obj) {
        data = obj;
        next = null;
    } // Node(Object)
} // class Node
```

Complete method `append()` in the following class. (**Hint:** Don't forget to deal with degenerate cases.)

```
public class SimpleLinkedList {
    private Node head; // first node in this list (null if the list is empty)
    /**
     * Adds the specified Object at the *end* of this linked list.
     * @param  <code>obj</code>  the Object to insert
     */
    public void append(Object obj) {

        ////  ANSWER  ////
        if ( head == null ) {
            head = new Node(obj);
        } else {
            Node current = head;
            while ( current.next != null )  current = current.next;
            current.next = new Node(obj);
        } // if

    } // void append(Object)
} // class SimpleLinkedList
```

**Marking Scheme:**

A. 1 mark for dealing correctly with the case when `head == null`

B. 1 mark for correctly adding the object at the end of the list

C. 1 mark for having the right general idea (looping through the list)

D. 1 mark for correctly creating a new `Node` to hold the object

E. 1 mark for general Java syntax