```java
package Project4;

import java.io.Serializable;

/**********************************************************************
 * This program builds a second linked list that is used to undo the
 * items added and removed to the primary single linked list
 *
 * @author Justin Von Kulajta Winn and Nick Layman
 * @version 1.9
 *********************************************************************/

public class MySecondLinkedList implements Serializable {

    /** this is the NodeUndo that represents the top of the linked list */
    private NodeUndo top;

    /**********************************************************************
     * This is a basic constructor that creates an empty list
     *********************************************************************/
    public MySecondLinkedList() {
        top = null;
    }

    /**********************************************************************
     * This function returns the NodeUndo at the top of the list but does
     * not remove the NodeUndo at the top of the list
     * @return top is the NodeUndo at the first position in the linked list
     *********************************************************************/
    public NodeUndo getTop() {
        return top;
    }

    /**********************************************************************
     * This function adds the passed auto to the top of the list. It does
     * not need to be searched or sorted because it is the most recent
     * addition. Therefore, whenever an item is added to this linked list,
     * it needs to be set to the top of the list.
     * @param s is the auto being added to the list
     * @param ROA represents if the auto was added, removed, or loaded from
     *            the primary linked list. This then impacts what action is
     *            performed when the undo button is pressed based on the
     *            value of ROA.
     *********************************************************************/
    public void addU(Auto s, int ROA) {
        //ROA: Remove or Add
        //Add is represented by a 1
        //Remove is represented by a 0
        top = new NodeUndo(s, top, ROA);
    }

    /**********************************************************************
     * This function removes the auto at the top of the list and returns
     * the removed auto to the user.
     * @return the auto that was removed
     *********************************************************************/
```

```
57      public Auto undo() {
58          Auto s;
59          s = top.getDataU();
60          top = top.getNextU();
61          return s;
62      }
63
64      /**************************************************************************
65       * This function gets the value of the ROA of the first NodeUndo in the
66       * linked list.
67       * @return the value of the ROA in the first NodeUndo
68       *************************************************************************/
69      public int getROA() {
70          return top.getRemoveOrAdd();
71      }
72
73      /**************************************************************************
74       * This was used to the test the linked list. It is not used.
75       *************************************************************************/
76      public String toString() {
77          return null;
78      }
79  }
80
81
```