

```
1  /*
2   Question 5
3   Creator: Wahid Bawa
4   Purpose: To create a method for removing duplicates to implement within a linked list class
5  */
6  public class LList {
7      private LNode head, tail;
8
9      public LList() {
10         head = null;
11         tail = null;
12     }
13
14     public void add(int val) {
15         LNode tmp = new LNode(val, tail, null);
16         if (tail != null) {
17             tail.setNext(tmp);
18         }
19         if (head == null) {
20             head = tmp;
21         }
22         tail = tmp;
23     }
24
25     public String toString() { // displays the parts of the linked list
26         String ans = "";
27         LNode tmp = head;
28         while (tmp != null) {
29             ans += tmp.getVal() + (tmp.getNext() == null ? "" : "-");
30             tmp = tmp.getNext();
31         }
32         return ans;
33     }
34
35     public void delete(LNode node) {
36         if (node == head) {
37             head = node.getNext();
38         }
39         if (node == tail) {
40             tail = node.getPrev();
41         }
42         if (node.getNext() != null) {
43             node.getNext().setPrev(node.getPrev());
44         }
45         if (node.getPrev() != null) {
46             node.getPrev().setNext(node.getNext());
47         }
48     }
49
50     public boolean contains(int val) { // this checks if a value is present in the list, returns true or false
51         boolean found = false;
52         LNode tmp = head;
53         while (tmp != null) {
54             if (tmp.getVal() == val) {
55                 found = true;
56                 break;
57             }
58             tmp = tmp.getNext();
59         }
60         return found;
61     }
62
63     public void removeDuplicates() {
64         LNode tmp = head;
65         LList singles = new LList(); // creating a linkedlist of unique values in the original list
66         while (tmp != null) {
67             if (singles.contains(tmp.getVal())) { // If the value is not unique, delete it
68                 delete(tmp);
69             }
70         }
71     }
72 }
```

```
70         else { // Count is as unique
71             singles.add(tmp.getVal());
72         }
73         tmp = tmp.getNext();
74     }
75 }
76 }
```