

In-class Exercises 07

*University of Wisconsin - La Crosse**Date: March 29*

1. Given the `SinglyLinkedList` class that appears after Question 2, write the instance method `splitEvenNodes` described below (this method will be located in the `SinglyLinkedList` class). You may choose to use other methods that are typically part of a linked list class (e.g., various `remove` methods), but you might find it easier to implement without them.

```
/**
 * Splits and returns a new list containing only the even nodes from the
 * original list. The original list thus ends up containing only the odd
 * nodes.
 *
 * @return A new list containing only the even nodes removed
 */
```

2. Implement a new inner class iterator `OddIterator`. This is an unusual iterator, in that it will only return the values at the odd indices in the list (usually an iterator returns every value). Your class should contain a constructor, implementations of the `next()` and `hasNext()` methods, and required global attributes.

Reference Classes

```
1 public class SinglyLinkedList<E> {
2     private int size;
3     private SingleListNode firstNode;
4
5     public SinglyLinkedList() {
6         // this list uses sentinel nodes
7         firstNode = new SingleListNode(null);
8         size = 0;
9     }
10
11     private class SingleListNode {
12         private E data;
13         private SingleListNode nextNode;
14
15         public SingleListNode(E i) { ... }
16     }
17 }
```

3. Conceptually, what advantage does using an iterator give us over using the `get` method for linked lists in order to retrieve all the elements of a linked list?

4. What advantage does using a sentinel node give us over not using a sentinel node?

5. Implement the `public void clear` method for the `SinglyLinkedList` class on the previous page given the use of sentinel nodes.

```
/**
 * Clears the data from the list.
 * @return void
 */
```