Linked Lists

* In a linked list, data is packaged into nodes, and each node links to the next one
* Add data to the list
* Delete data from the list

Adding an iterator

* What is an iterator? It is a way to traverse through an entire list, one element at a time
* What can you do with an iterator? You can use a for-each loop
	+ You can use a for-each loop (enhanced for loop) with any data structure that implements Iterable interface

To add an iterator:

* Data collection must implement the Iterable interface
	+ Iterable interface has only one method: iterator()
	+ iterator() returns an object of type Iterator
	+ Type parameter of Iterable should be the underlying data type (e.g. String)
* Iterator is an interface with three methods:
	+ hasNext() – returns true if there is still data to traverse through (i.e. there are still elements in the iteration)
	+ next() – returns data and moves along to the next element in the list
	+ remove() – optional, we won’t implement it
	+ Type parameter should be the underlying data type (e.g. String)

Auxiliary classes to help a big class

* e.g. LinkedListNode, MyIterator for LinkedList
* These classes make no sense outside of LinkedList – there is no need for the client to see them or be concerned with them
* Make them private “inner” classes within LinkedList