

## Reversing a Singly-Linked List

In this exercise:

- you will implement a “supplemental” linked list method
- you will practice working with nodes and pointers

### The Task

Design and implement an `SLList` method `reverse()` that reverses the order of elements in an `SLList`. This method should run in  $O(n)$  time, should not use recursion, should not use any additional data structures, and should not create any new nodes. (You may of course need a few extra variables to store temporary values.)

### The Details

For simplicity, just write this method in the book’s `SLList` files.

Test your code. Write a main program that creates a list, fills it with some values, calls `reverse()`, and checks that the resulting list is indeed reversed. Be sure you test your code on a list that contains 0 values, 1 value, 2 values, and many values.

### General Note

The exercises in this chapter are *very* good. I’d recommend working all of them (with the possible exception of the last four, though give those a shot too) as a way to study for the midterm.