Goal: This lab will give you practice with linked lists.

Copy the Lab 3 directory by starting from your home directory and typing:

cp -r $master/lab/lab3 .
cd lab3

Getting Started

Please make sure you have a partner for this lab.

The files in the lab3 directory contain classes to implement a singly-linked list. The classes include an encapsulated SLList and the underlying SLListNode class (which is akin to a Scheme cons cell). Compile SLList.java (using "javac -g SLList.java"). Run the resulting test code with

java SLList

The main() method of SLList includes test code, which can be used to help debug the list code before SLLists are used in other programs.

Items in our SLLists are indexed starting from 1, unlike Java arrays.

Part I: Using SLLists (1 point)

In the main() method, construct a list that looks like:

[ 6 9 12 ]

and print the resulting list.

Add more lines to change this list to:

[ 3 6 9 12 15 ]

and print the resulting list.

Part II: Adding to the End of a SLList (3 points)

A method called insertEnd() exists, but it runs in linear time, because every time it is called, it walks down the list to find the end. Without changing the meaning of this method or any other, modify the representation of a SLList and whatever methods are necessary to make insertEnd() run in constant time.

Your SLList class will need to continually maintain a record of the last (tail) SLListNode in an SLList, and all SLList’s methods will have to ensure that this record stays current.

Check-off

-----
Show your TA or Lab Assistant your main() and insertEnd() methods and run the program.

1 point: Show your main() method, and show that it is printing the proper output for Part I.

3 points: Show your insertEnd() method, and explain how you got it to work in constant time. Show that your program still prints the right output. Which other methods had to be modified?