


[In Class] Lesson 6 (10/18)

Looping over Lists

 **Note:** we will be referring to a connected sequence of `ListNode`s as a "list"

Write the following methods:

`printList`: takes a `ListNode` as a parameter and prints out the values in all nodes in the list in a bracketed, comma-separated form

`createList`: takes an array of integers as a parameter and returns a `ListNode` that is the front of a list containing the values in the array in the same order they appear in the array

`countOccurrences`: takes a `ListNode` and an integer as parameters and returns the number of times the integer appears as a value in the list

`insertAt`: takes a `ListNode` and *two* integers as parameters, and inserts the first integer into the list at the index specified by the second integer (assume there are at least as many nodes in the list as the index)

`removeAll`: takes a `ListNode` and an integer as parameters and removes all nodes that have the integer as a value from the list