CSC 217 Lab 11

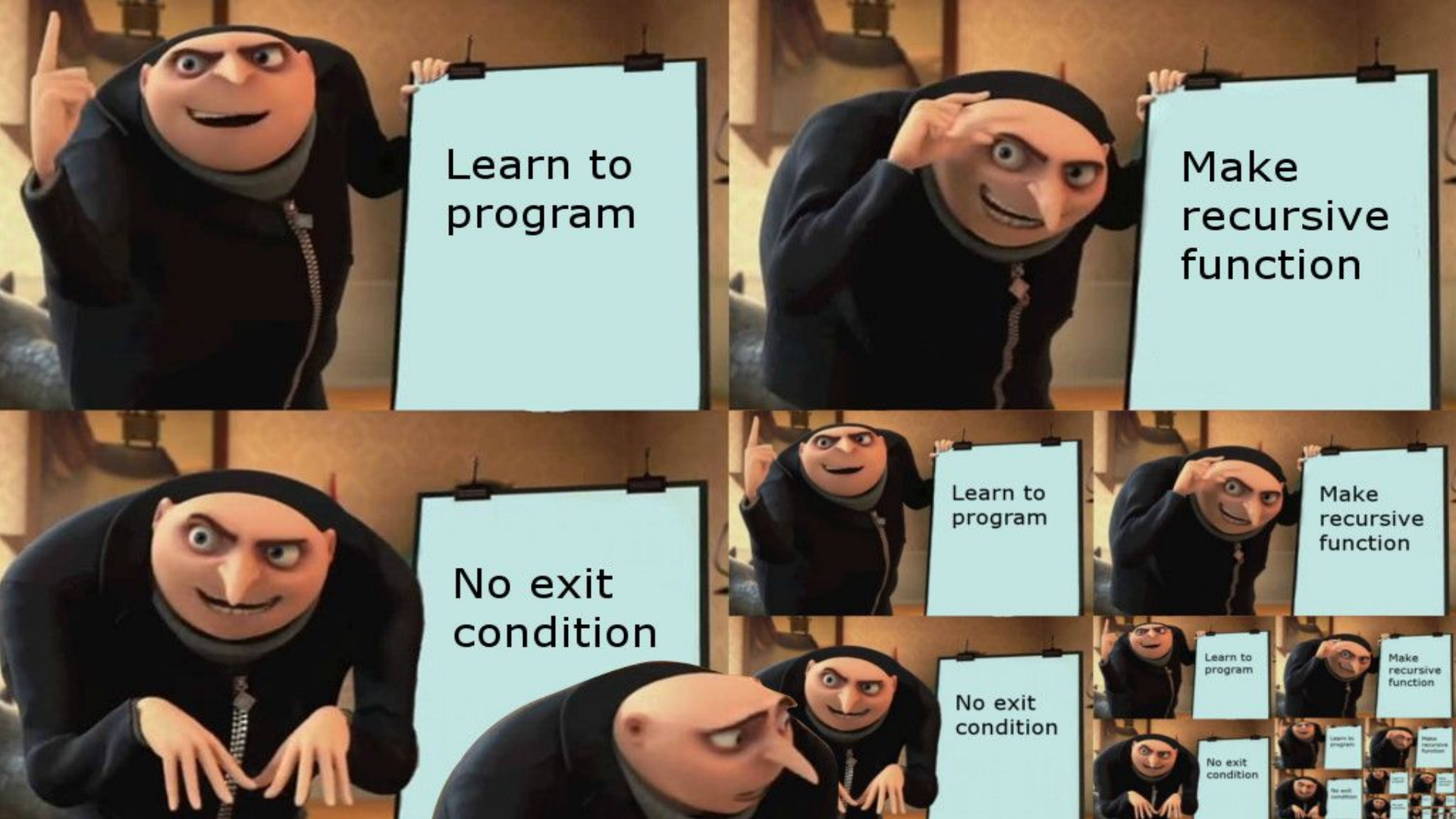
Recursive LinkedLists

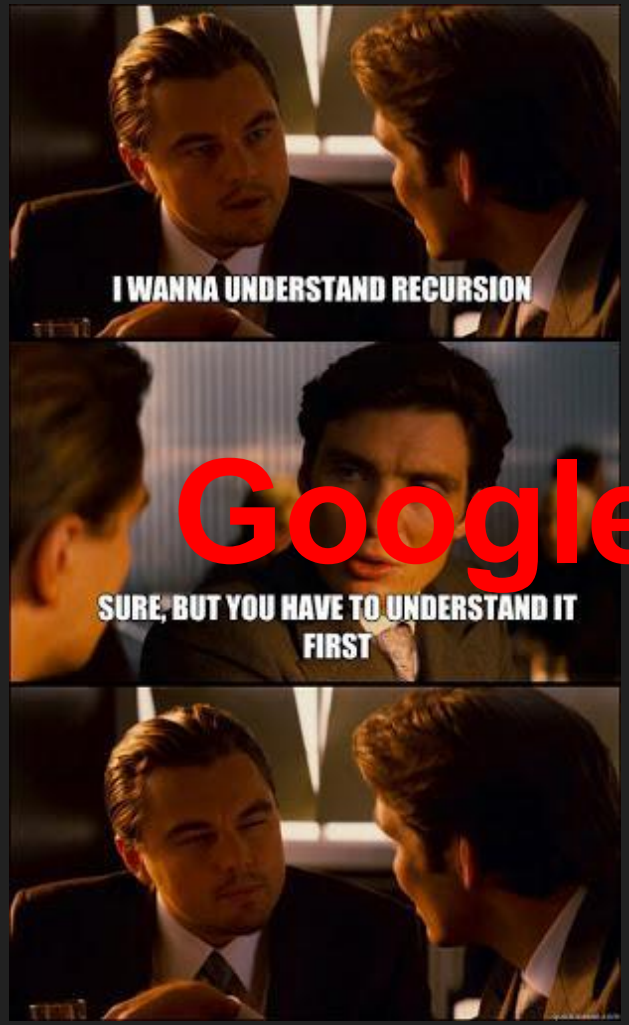
Lab Overview

- Deadlines & Reminders
- Recursion Tips
- Activities
  - Implement LinkedListRecursive recursively
    - You will be graded on using recursion through an inspection of your code!
  - FacultySchedule Functionality
    - FacultySchedule - a colleague will help you out here!
    - Faculty
    - Course and CourseRecordIO - to tie together the Faculty and the Courses they are assigned to teach through FacultySchedule
    - RegistrationManager
- Lab Wrap-Up

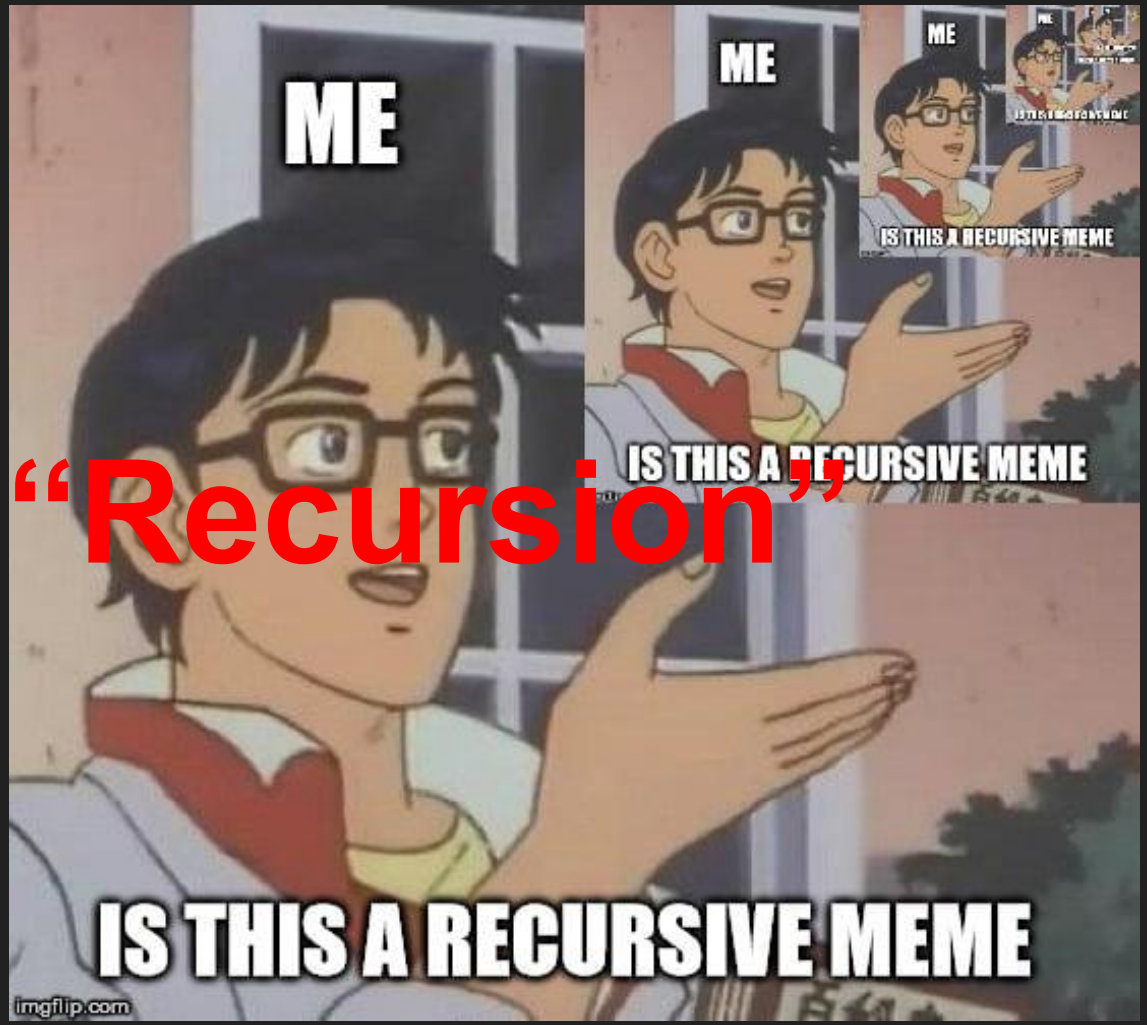
Deadlines & Reminders

- Deadlines
  - See lab write up for deadlines.
- Reminders
  - Dr. Heckman has decided to drop the lowest lab grade, *except for Lab 11!*
  - This is the last lab of the semester!
  - Lab grades should be finalized and posted to MyPack the last week of April.


Learn to program
Make recursive function
No exit condition
Learn to program
Make recursive function
No exit condition
Learn to program
Make recursive function
No exit condition



Google “Recursion”



Recursive LinkedLists

Implementation will have public/private pairs

- Public method, in LinkedList
  - handles special cases
  - performs error checks
  - delegates to `ListNode.method()` once done checking
- Private method, in ListNode, handles the recursion
  - next is the rest of the list
  - If there's an index parameter, that typically changes with each recursive call!

Recursion Tips

Each recursive method has 2 components:

- Base case
  - contains(E data):
    - if this node has the data, list *does* contain data
    - if this node has no next node, rest of list *does not* contain data
- Recursive case
  - contains(E data):
    - if this node doesn't have the data, and has a valid next:
      check the rest of the list
  - One step at a time

Patterns of Recursion

Method	Base Case			Parameter Values			Return Value		
	Empty	Head	Index	Same	Vary	None	Same	Vary	None
contains(E)	X			X			X		
size()	X					X		X	
get(idx)	X		X		X		X		
set(idx, E)	X		X		X		X		
add(E)	X	X		X			X		
add(idx, E)	X	X	X		X				X
remove(E)	X	X		X			X		
remove(idx)	X	X	X		X		X		

Activities

- Implement LinkedListRecursive recursively
  - You will be graded on using recursion through an inspection of your code!
- FacultySchedule Functionality
  - FacultySchedule - a colleague will help you out here!
  - Faculty
  - Course and CourseRecordIO - to tie together the Faculty and the Courses they are assigned to teach through FacultySchedule
  - RegistrationManager

Wrap-Up

General Wrap-Up

- Deadline Reminder - deadline in writeup is for ALL SECTIONS! Lab 11 grades are NOT dropped!
- Exchange contact information with your partner
- Make a plan for finishing up the lab

Participation Outside of Lab (Guess which the teaching staff prefer?)

- If you pair program/design, **note that in the commit comments so everyone gets credit!**
- If you split the work, at least one contribution by each partner

REMINDER: We are expecting a significant contribution from all team members outside of lab!

- If you pair program/design, you **MUST** note it in your commit messages or there will be deductions
- Students who don’t allow their partners to contribute will receive deductions
- Students who don’t contribute will receive deductions

Record Tasks & Owners

Tasks only get done when someone owns them!

Identify the tasks required to complete Lab 11

- Edit README.md to list the tasks required to complete Lab 11(at top of README - should come before Lab 10 tasks)
- Add an owner to each task
- Add a deadline to each task

Deadlines should be at least 48 hours before the lab deadline so team members can help out and finish the lab if a team member runs into issues.

Notify team early if you run into problems with your tasks!