

## CSC151.02 2013F, Class 24: Recursion Basics

---

### Overview

- Admin.
- Exercises in "pass the buck" problem solving.
- The idea of recursion.
- Lab.

### Admin

- Since only about half the class responded to the partner survey, I worry that those who want to choose partners will have a limited group. So ... if you're choosing partners (or working alone), please let me know by 8 p.m. tonight who your partners are. After that, I'll assign partners.
- We will continue today's topic on Friday (with another guest faculty member). You may find it useful to reread "Recursion Basics" for Friday's class.
  - And yes, you'll have time in class on Friday to continue to work on the lab.
- Upcoming EC opportunities
  - Learning from Alumni, Thursday, 2:15-4:05: Skype Visit with Tony Stubblebine '00 from Lift
  - Thursday extra at 4:15: Grad school in CS
  - Friday CS Table: Reports from Hopper
  - Friday film: Codebreaker, 7pm, Harris
  - Post-Codebreaker discussion
  - Wit, the play, opens Thursday.
  - Any Queer Culture Week Activity
    - E.g., Wednesday at 7:30, Consent, Negotiation, and BDSM,
  - Men's Soccer, 2:30 p.m. 10/12 vs. Monmouth
  - Men's Soccer, 1:30 p.m. 10/13 vs. UWisc/Platteville
  - ...
- Other upcoming stuff
  - Review session 1:15 pm Thursday
  - Mentor session 7:30 pm Thursday
  - Mentor session 7:00 pm Sunday
  - ...
- 10/10 is this weekend. Please behave responsibly.

### Welcome to "Pass the Buck" Problem Solving

```
Remove one element
Pass the remaining elements to our left to get a count
Add 1
```

```
(define toby-count (lambda (books) (+ 1 (alex-count (cdr books)))))
```

```
(define alex-count (lambda (books) (+ 1 (andrew-count (cdr books)))))
```

```
(define please-find-all-the-books-by-david-campbell-in-this-list (lambda (books) (if (null? books) null (let ([abook (car books)] ; The first book [rest (cdr books)]) ; All but the first book (if (is-by-david-cambell abook) (cons abook (please-find-all-the-books-by-david-campbell-in-this-list rest)) (please-find-all-the-books-by-david-campbell-in-this-list rest)))))
```

(map by-david-campbell? books) (for-each ? books) -> Returns nothing (repeat ? books) - Won't work with lists

We have learned a more general strategy for repetition.

```
(define proc (lambda (lst) (if (null? lst) simple-result (extend (proc (cdr lst)))))
```

```
(define length (lambda (lst) (if (null? lst) 0 (+ 1 (length (cdr lst)))))
```

```
(define bbdc (lambda (lst) (if (null? lst) null (magic (bbdc (cdr lst)))))
```

## The idea of recursion

### Lab

---

Samuel A. Rebelsky, rebelsky@grinnell.edu

Copyright (c) 2007-2013 Janet Davis, Samuel A. Rebelsky, and Jerod Weinman. (Selected materials are copyright by John David Stone or Henry Walker and are used with permission.)



This work is licensed under a Creative Commons Attribution 3.0 Unported License. To view a copy of this license, visit <http://creativecommons.org/licenses/by-nc/3.0/> or send a letter to Creative Commons, 543 Howard Street, 5th Floor, San Francisco, California, 94105, USA.