CSC151.02 2014S, Class 56: Discussion of Exam 3 and Review for Final

Overview

- Preliminaries.
  - Admin.
  - Questions.
  - Pre-quiz questions.
  - Quiz!
- Hacking 2048.
- About the final.
- Short discussion of exam 3.
- Some questions about the course.

Preliminaries

Admin

- Cookies! (And some gifts from TN.)
- You should have received a prospective grade sheet.
  - Your grade cannot go down from that grade.
  - Let me know questions and concerns.

Extra Credit

- Improv at a time to be improvised. CS will email you.
- Neverland study break Tuesday the 13th at 9 or something.

Questions

Quiz!

1. Write a definition for the map procedure [5 points]

   (define map
     (lambda (fun lst)
       (if (null? lst)
           null
           (cons (fun (car lst)) (map fun (cdr lst))))))
2. Write a function, squares, that, given n, computes the list (1 4 9 16 ... n^2) [5 points]

```scheme
(define squares
  (lambda (n)
    (map square (cdr (iota (+ n 1))))))
```

Hacking 2048

About the final

- In class.
- Written - Kind of like a really big quiz.
  - No computer.
- Four problems. Four mostly correct: A. Three mostly correct: B. Two mostly correct: C. One mostly correct: D. Here: F.
- One sheet of notes. 8.5x11 inches double sided. Can be typed.
  - Important learning comes from doing a review sheet.
- 9 am or 2pm Tuesday.
  - May be in 3815.
- Difficulty somewhere around the normal exam problems (i.e., a hard quiz)
  - Some code writing
  - Some code reading
- Sam should send out a sample exam (and will do so tonight).
- Cumulative, but emphasis on recent things
  - Object problem
  - Sorting or searching problem
  - Higher-order procedures
  - Lists or trees or ...
  - NO FILES
- Strategies for reviewing
  - Look at recent labs, particularly the extras
  - Design a few problems of your own
  - Read the code from readings
  - Make a list of the "most important" learning outcomes from this course (at least in terms of Scheme/programming)
Short discussion of exam 3

Some questions about the course

Something I should do differently/better in the fall  * Respond more to labs. * Generally not enthusiastic about switching to "what were the big ideas of this lab" writeups, which were recommended by 207. * Do more physical demonstrations of algorithms * Optional "flipped" videos? Good for some, worry about work with others. Watch someone write the code. Talk about some of the small parts. * Don’t force partners. Many people will learn more doing it on their own. * Distribute virtual machines earlier. * Focus on lab time. * Exams were challenging; maybe twice as many half as long?

Samuel A. Rebelsky, rebelsky@grinnell.edu

Copyright (c) 2007-2014 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.