

CSC151.02 2014S, Class 25: Recursion Basics, Continued

Overview

- Preliminaries.
 - Upcoming work.
 - Admin.
 - Questions on HW6.
- Reflection.
- Questions.
- Lab.

Preliminaries

Upcoming Work

- Reading for Tuesday: Recursion with Helper Procedures
- Today's lab writeup: Exercises 4 and 5
 - CSC 151.02 Writeup 16: Recursion Basics
- HW6

Admin

- Kim will be teaching class Tuesday. Thanks Kim!
- Mr. Stone will be teaching class Wednesday and Friday.
- Exam 2 will be distributed on Wednesday.
- No quiz on Friday!
- Choose your own (new) partners tomorrow. Strive to find someone you have not worked with before.
- Extra credit for volunteers to do eboards on Tuesday, Wednesday, and Friday.
 - KS, Tuesday
 - CS, Friday
- Extra credit:
 - Convocation Wednesday, March 5.
 - Orchestra
 - With additional Narration and Animation, Wednesday, March 5
 - 7:30 p.m. in Sebring-Lewis
 - Grinnell image presentations at noon on Thursday and Friday.
 - First Years: How to Use Your Summer, 8-9 pm on March 6 in JRC 101.
 - Balancing Acts
 - Friday, 7 March at 7:30 in Flanagan
 - Saturday, 8 March at 7:30 in Flanagan

- Sunday, 9 March at 2:00 in Flanagan
- Neverland Players this weekend
 - Friday, 7 March at 7:00 in the Wall
 - Saturday, 8 March at 2:00 in the Wall
 - Saturday, 8 March at 7:00 in the Wall
 - Sunday, 9 March at 2:00 in the Wall
- Men's Tennis, Saturday, 8 March (8 am or 4pm)

Questions on HW6

Should every procedure have documentation?

Yes. You can be casual on preconditions and postconditions

Will our notes suffice for 3 and 4?

Yes, provided I remember to tell the graders.

What should the inputs to problem 4 look like?

It's a program, so it doesn't really need inputs. Just show me the sequence of procedure calls to generate the image. E.g.,

```
(define canvas (image-new 400 400))
(render-polar-rose-on-image! canvas 50 23 "blue")
```

Reflection

What did you learn on Friday?

- Recursion
- Using a function within a function so as to achieve repetition
- Purpose: Achieve repetition
- Technique: Implement a procedure by making that procedure call itself (some of the time)
 - If the condition holds, call itself
 - If the condition does not hold, stop
 - Or vice versa
- If we're working with a list, we can throw away an element of the list each time and stop when there's nothing left in the list.
- Why not just use map?
 - Can't have the condition part; always deals with every element in the list
 - map only produces a list; sometimes we want to produce values other than a list.
- General form of a recursive procedure

```
(define fun (lambda (vals) (if (TEST) (fun ) BASE-CASE))))
```

Questions

Lab

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.