CSC151.02 2013F, Class 07: Writing Your Own Procedures

Overview

- Preliminaries
  - Admin.
  - Connecting to MathLAN from your own computer.
- Why define your own procedures?
- How to define your own procedures.
- Lab.
- Reflection: What was the most confusing thing you dealt with in lab today?
  - I'll probably ask this question tomorrow, but think about it today.
  - "Sam" is not an acceptable answer.

Admin

- Reminder: Mentor sessions are Thursdays and Sundays at 7:30 pm.
- Reminder tomorrow's class is next door in 3815.
- HW3 will be distributed in class tomorrow.
- Reading for Wednesday
  - Documenting Procedures
- EC Opportunities:
  - Humanities Center Speaker Sarah Hendron Wed., Sept. 11, 7:30 p.m., JRC101 Waking the Machines: Art, Design, and Adaptive Technology
  - CS Table Friday (Reflections on Trusting Trust; NYT on NSA)
  - Others

How do I do CSC 151 work from my own computer?

- General instructions at http://www.cs.grinnell.edu/vnc
- I'll demo on my Mac. It's a bit harder if you use Microsoft Windows (and I don't know all of the details because I rarely use Windows).
  - I won't type the steps as I go

Why define your own procedures?

What is a procedure?

- A procedure is like a variable except it’s an action (or series of actions) rather than just a simple value
  - That is, it automates a series of tasks
- On the scheme side, a procedure takes inputs and generates outputs
  - a square function might take a number as an input and produces a number as an output x -> x*x
  - make-a-circle: take as input x,y,radius, (number) etc.; output is a drawing taht represents a circle
  - um is a procedure that takes a question as input and gives as output the value "um" which is text (a string)

Why write them?

- Less redefinition
- More elegant
- More readable

- Easier to write

  (make-a-circle 10 10 100)

  is probably clear than (drawing-shift (drawing-scale drawing-unit-circle 100) 10 10)

How to define your own procedures

```
(define *procname*
  (lambda (*inputs*)
    *expression*))
```

For example

```
(define um
  (lambda (question)
    "um"))

(define square
  (lambda (num)
    (* num num)))

(define make-a-circle
  (lambda (x y radius)
    (drawing-shift (drawing-scale drawing-unit-circle (* 2 radius))
                   x y)))
```

Questions:

- Can procedures define other procedures?
  - Yes, but we won’t cover that for a few weeks
- What’s the delay?

- You can't really evaluate the body of a procedure until you use the procedure?

- So how do you know it's correct?
  - You try it and see.
  - You try it a lot of times and see.
  - You do a formal proof

- How do I "read" a definition

  (define *procname* (lambda (*inputs*) *expression*))

  I am defining a procedure named *procname* that takes as input *inputs* and computes its result by evaluating *expression*.

- When

Lab

- Remember to shift who is at the keyboard! (Unless you're one of those great groups that seems to function as one person.)
- Problems 5 and 6 will be your lab writeup (and yes, I'll email that out)

Samuel A. Rebelsky, rebelsky@grinnell.edu