

# CS-257L

## Nonimperative Programming: Scheme!

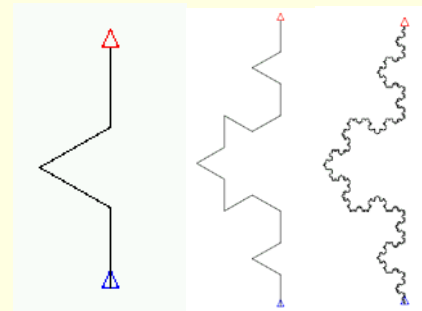
Instructor:

Joel Castellanos

e-mail: [joel@unm.edu](mailto:joel@unm.edu)

Web: <http://cs.unm.edu/~joel/>

Office: Farris Engineering  
Center (FEC) room 321



# Homework

---

- Due Sunday Night at Midnight (2/3/2008)
  - Read “The Little Schemer” Chapter 2.
  - Verify the Examples in MzScheme or DrScheme.
  - Create 3 original questions/answers in the style of the text.
    - Use only syntax introduced in chapter 1 & 2.
    - Submit into WebCT.
    - Use plain text.
  - Grading:
    - C: Follow the rules.
    - B: One is Creative, Insightful, Thought Provoking.
    - A: Two or Three are C/I/TP.

# Grading Rubric for HW-2: good

---

10 pts:

- Runs correctly in DrScheme on the hard-coded list: '(a b c d e f)
- Runs correctly if the hard-coded list is changed to: '(x y z d e f) and '(x y z d e)
- Makes a good attempt at error checking,
- B can die if given: '(a b) or ().
- Any style or approach is fine.
- Comments are not necessary.

9 pts:

- Runs as above, but without or with a poor attempt at error checking.

# Grading Rubric for HW-2: not so good

---

6 pts:

- Does not run.
- Is mostly correct, but has one or two small syntax error(s) such as a missing space.

4 pts:

- Has some ideas correct, but has many syntax errors or a significant logic error.

2 pts:

- Turned something in that appears to have had some effort applied.

0 pts:

- Missing, little effort, or demonstrates troll-like thought.

# A Solution

---

```
(define p '(a b c d e f))
(cons (car p)                ;cons a (b c)
      (cons (car (cdr p))    ;cons b (c)
            (cons (car (cdr (cdr p))) () ;cons c ()
                  )
            )
      )
)
)

(a b c)
```

# Aaron Holck

---

- What is  $(eq? (null? '(a)) (null? '(b)))$

- #t

because  $eq?$  can also compare true and false.

# Valerie Steinhaus

---

- What is  $(eq? a b)$   
where  $a$  is river, and  
 $b$  is (river).

```
(define a 'river)
(define b '(river))
(eq? a b)
```

- By the book: no answer, because  $b$  is not an atom.
- By DrScheme: **#f**

# Valerie Steinhaus – follow up

---

- What is  $(eq? a b)$

where  $a$  is (river), and  $b$  is (river).

```
(define a '(river))
```

```
(define b '(river))
```

```
(eq? a b)
```

- By the book: no answer, because  $a$  and  $b$  is not an atoms.
- By DrScheme: **#f**, because  $a$  and  $b$  are not atoms.

# Alexander Roessner

---

- Which atom is excluded from the inner list with the other atoms?

*((obama clinton edwards) kucinich)*

- *kucinich*

# Alexander Roessner's Sandwich

---

- What is this?

```
(define pb '(peanut butter))  
(define & '(and))  
(define j '(jelly))  
(define s '(sandwich))  
(cons (cons (cons pb &) j) s)
```

The tricky part is getting the parenthesis correct.

```
((((peanut butter) and) jelly) sandwich)
```

# Daniel McGraw

---

- Is this true?

*(null? (cons () ()))*

No, *cons* of a pair of *null* lists results in a *null* list contained in a list: *(( ))*.

**Law of Nothing:** Nothing is Nothing. Nothing in Nothing is Nothing, but Nothing around Nothing is a list.

# Inspired by Christian Romano

---

- Which is (are) false?

1. `(eq? '5 '5)`
2. `(eq? '(+ 1 4) '5)`
3. `(eq? 5 5)`
4. `(eq? (+ 1 4) 5)`

#2, because the quote keeps the + function from being evaluated.