

# 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 Monday, March 10

---

- Nothing to hand-in.
- Scheme and The Art of Programming
- Read Chapter 5, and do
  - Exercise 5.3
  - Exercise 5.9
  - Exercise 5.10
- Be prepared for a quiz on this material.
  - Open notes
  - Open book
  - Closed Computer

# Scheme and the Art of Programming: Exercise 4.14

---

```
(define harmonic-sum-it
  (lambda (n)
    (cond
      ((or (not (integer? n)) (< n 1))
        ("Error: argument must be a positive whole
number."))
      ((= n 1) 1)
      (else (+ (/ 1 n) (harmonic-sum-it (- n 1)))))
    )
  )
)
```

# Scheme: Lexically Scoped

---

```
((lambda (x)
  ((lambda (y)
    (- x y)
    )
    15
  ))
  20
)
```

# Scope of Bindings for Variables

---

User global environment bindings:

`(define variable expression)`

Local bindings:

`(let ((variable expression)) body)`

`(let ((var1 exp1) (var2 exp2) ...  
 (varn expn)) body)`

## Quiz 4 – What is the output?

```
(define count-parens
  (lambda (ls)
    (display ls) (newline)
    (cond
      ((null? ls) 2)
      (else
       (cond
          ((atom? (car ls))
           (count-parens (cdr ls)))
          (else
           (+ (count-parens (car ls))
              (count-parens (cdr ls))))
         )
      )
    )
  )
)

(load "atom.scm")
(count-parens '((a b) () c))
```