

# 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

```
(define f
  (lambda (n)
    (cond
      ((= n 1) 1)
      ((= n 2) 1)
      (else (+ (f (- n 1)) (f (- n 2))))))
  )
)
```

# Homework - Sunday Night at Midnight

---

- 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.

# Next Slide Quiz

---

- Closed Computers,
- Closed Neighbors,
- Closed Cell Phones
- Open Notes, and books

# Quiz

---

What is:

```
(define x '(a b c))  
(define y '(d e f))  
(cons (car y)  
      (cons (car (cdr x)) ()))  
)
```

# What does this do?

```
(define f
  (lambda (n)
    (cond
      ((= n 0) 0)
      ((= n 1) 1)
      (else (+ (f (- n 1)) (f (- n 2))))))
  )
)
```

Fibonacci numbers:

$$F(n) = \begin{cases} 0 & \text{if } n = 0; \\ 1 & \text{if } n = 1; \\ F(n-1) + F(n-2) & \text{if } n > 1. \end{cases}$$

What is:

(f 4)

(f 5)

(f 6)

# Fibonacci Numbers by If and Case

```
;(if <test> <consequent> <alternate>)  
(define g  
  (lambda (n)  
    (if (= n 0) 0  
        (if (= n 1) 1  
            (+ (f (- n 1)) (f (- n 2)))))  
    )  
  )  
)
```

```
;(case <key> <clause1> <clause2> ...)  
(define h  
  (lambda (n)  
    (case n  
      ((0 1) 1)  
      (else (+ (f (- n 1)) (f (- n 2)))))  
    )  
  )  
)
```

# What is (k 25)

---

```
(define g
  (lambda (a b) (zero? (remainder a b))))
)
```

```
(define h
  (lambda (a b)
    (if (= a b) 'yes
        (if (g a b) b (h a (+ b 1)))
        )
    )
  )
)
```

```
(define k (lambda (a) (h a 2)))
```