

# CS 357 - Lab 002

Session 2

Let & Homework 2

Kage Weiss

# CS 357 - Lab 002

TA: Kage Weiss

Office Hours: Friday 3-4pm, FEC 2360, or by appointment.

Contact: [mmweiss@unm.edu](mailto:mmweiss@unm.edu)

Website: <http://cs.unm.edu/~kageweiss/TA/cs357.html> -- **SLIDES POSTED**

- Sign in by quiz today: see last slide for code
- Today we are working on Homework 2, if you have any questions, please let me know!
- Please read the CODING STANDARDS and use the provided .scm as a base

## How I grade your code

If you use the provided file, we all get grades done quicker.

Late penalty is 10%/24hrs late.

- Assuming you defined the required functions correctly, all I get is a #t to show you passed.
  - If not, I print the tests that failed and take a closer look
- If everything is good, I'll glance over your code for coding standards, assignment-specific code patterns, notable workarounds, etc.
  - Anything important enough to note and I'll write a grade comment

# let, let\*, letrec

## let

- Allows for scoped definitions
- Does not allow for individual definitions to reference each other
- Does not allow for individual definitions to reference themselves

```
(define fn
  (let
    ( ;; definitions
      (id1 def1)
      (id2 def2)
      ...
    )
    ( ;; usage
      (lambda (x y) (id2 (id1 x ...)))
    )
  )
)
```

# let, let\*, letrec

## let\*

- Allows for scoped definitions
- Allows for individual definitions to reference each other
- Does not allow for individual definitions to reference themselves

```
(define fn
  (let*
    ( ;; definitions
      (id1 def1)
      (id2 (... def2 ... id1 ...))
      ...
    )
    ( ;; usage
      (lambda (x y) (id2 (id1 x ...)))
    )
  )
)
```

# let, let\*, letrec

## let\*

- Allows for scoped definitions
- Allows for individual definitions to reference each other
- Allows for individual definitions to reference themselves

```
(define fn
  (letrec
    ( ;; definitions
      (id1 (... id1 ...))
      (id2 (... def2 ... def1 ...))
      ...
    )
    ( ;; usage
      (lambda (x y) (id2 (id1 x ...)))
    )
  )
)
```

# CS 357 - Lab 002

Go forth,  
write your software.

Remember, these slides are available:

[cs.unm.edu/~kageweiss/TA/cs357.html](http://cs.unm.edu/~kageweiss/TA/cs357.html)