

SEG 2105 - LECTURE XX

How To ANSWER As ASSIGNMENT

**YOU CAN APPROACH ALMOST ALL YOUR
HOMEWORK IN A SIMILAR MANNER, WITH FEW
EXCEPTIONS**

OBJECTIVES OF AN ASSIGNMENT

To answer the questions?

- ▶ No exactly

More so, to

- ▶ To learn how to follow instructions
- ▶ To learn to organize your thoughts into a coherent manner
- ▶ To explore the subject matter in a hands on manner

**YOU CAN APPROACH ALMOST ALL YOUR
HOMEWORK IN A SIMILAR MANNER, WITH FEW
EXCEPTIONS**

3 PARTS TO ANSWERING YOUR ASSIGNMENT

LOGISTICS ANSWER TEMPLATE SOLUTION

LOGISTICS

- ▶ Put it in the desired format, with all the necessary information
- ▶ Naming your files for good sorting and to be easily identified
- ▶ Include an exact copy of the questions, always!
- ▶ Always make sure to provide all the necessary information
 - ▶ Name
 - ▶ Student Number
 - ▶ Course Code
 - ▶ Professors Name
 - ▶ TA names
 - ▶ Homework "Title"
 - ▶ Due Date

SEG2105 - HOW TO ANSWER AN ASSIGNMENT

Andrew Forward (1484511)

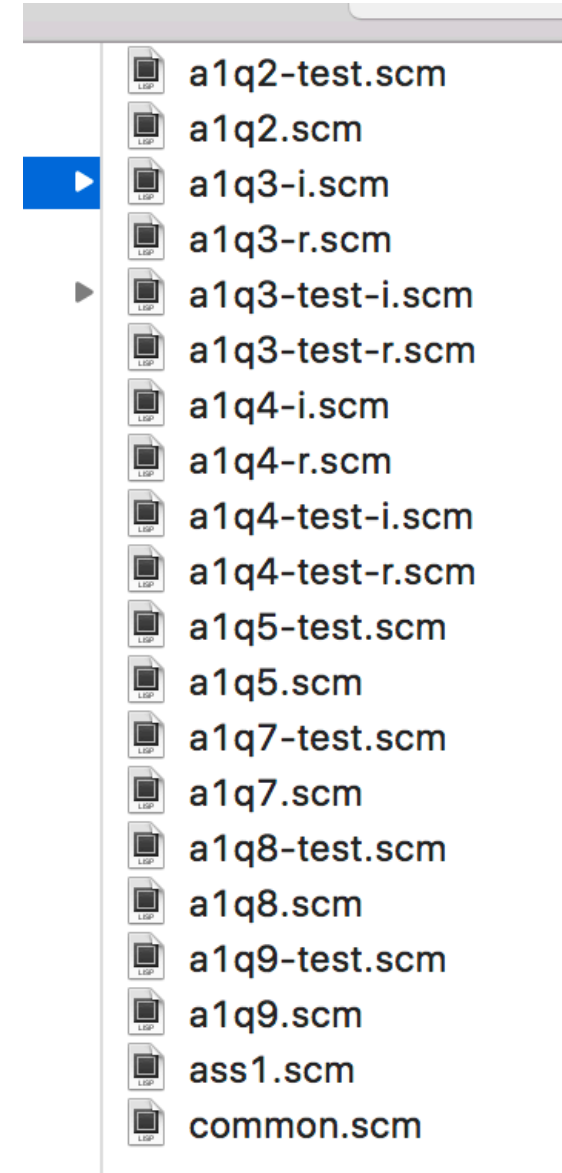
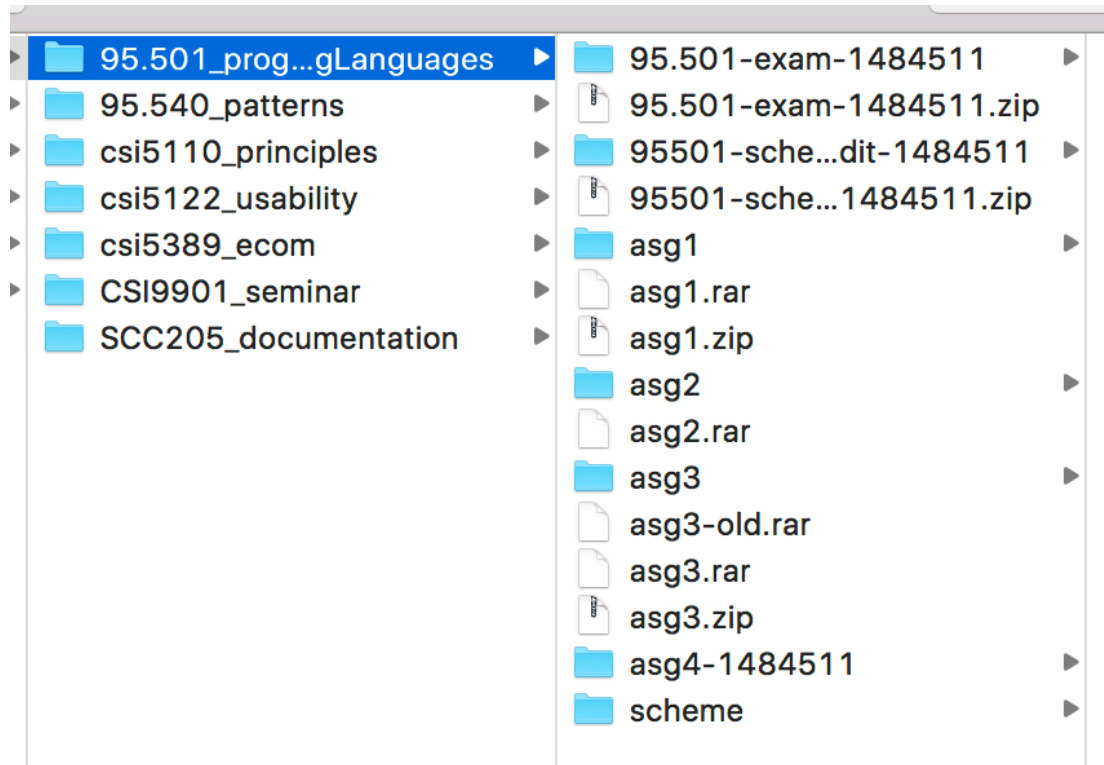
Asg 1

Due: Friday September 28, 2001

Scheme Assignment 1

Author: Andrew Forward
Number: 1484511
Prof: Dr. Dwight Deugo
TA: Darrell Ferguson
Asg: 1

SEG2105 - HOW TO ANSWER AN ASSIGNMENT



RE-ITERATE THE QUESTION FOR 3 REASONS

- 2) *Define a procedure that takes three numbers as arguments and returns the sum of the squares of the numbers that are duplicated.*

For example,

(sum-square-duplicates 1 2 4) => 0

(sum-square-duplicates 1 1 4) => 2

(sum-square-duplicates 1 1 1) => 3

(sum-square-duplicates 1 3 3) => 18

(sum-square-duplicates 5 2 2) => 8

Easier to read

Avoids answering in the wrong order

Allows homework to stand on
it's own, ever 15+ years later

ANSWER TEMPLATE

- ▶ Always get the ceremony of each question done at the start
- ▶ When you finally “get it working” at 3am, you don’t want to think about transcribing it into a coherent answer
- ▶ Also ensure that you answer every question

- 5) Define a procedure *double* that takes a procedure of one argument as an argument and returns a procedure that applies the original procedure twice. For example, if *increment* is a procedure that adds 1 to its argument, then *(double increment)* should return a procedure that adds 2. What is returned by the following and why?

((double (double double)) increment) 4)

The Double Procedure Procedure

```
(define (double p2)
  -1)
```

Using the above definition, the following results

```
> (((double (double double)) increment) 4)
999
```

Why?

Just Because

**TO DATE, THIS IS ALL CEREMONY,
NO SUBSTANCE**

- ▶ So, get this done as soon as possible
- ▶ You should be ready to “submit” your homework at any time

THE SOLUTION

The Double Procedure Procedure

```
(define (double p2)
  (lambda (x) (p2 (p2 x))))
```

Using the above definition, the following results

```
> (((double (double double)) increment) 4)
20
```

Why?

The double procedure is doubled 3 times... so in essence this would mean to do the original procedure is doubled 4 times, or in essence, do the procedure 16 times (2^4) to itself.

To double a double, means to do it 4 times (2^2).

To double an already doubled means to do it 16 times (4^2)

And since the increment only adds 1, we will add 1 sixteen times whereby $4 + 16 = 20$

THE SOLUTION

- ▶ This obviously depends on the actual homework
- ▶ But again, just to date, we have something that is almost ready to submit
- ▶ What follows will help you to maximize your result

ABC

ALWAYS BE COMPILING

- ▶ A program that doesn't compile (or interpret) has almost no value
- ▶ If it doesn't work, it has no "value" to the user
- ▶ Writing software is quite different than other engineering disciplines
- ▶ You can (almost) always start small
- ▶ You can (almost) always increment in small chunks
- ▶ Never write book to completion, and *then* turn on spell/grammar check

TESTS FIRST TO HELP GUIDE YOUR ANSWER

```
(define (heapsort list)
  "This Course Rocks!")

(print (heapsort ' ()))
(print (heapsort ' (1)))
(print (heapsort ' (1 2 3)))
(print (heapsort ' (4 3 2 1)))
(print (heapsort ' (5 3 4 1 2)))
(print (heapsort ' (9 -2 5 22 3 4 1 48 2)))
```

IDENTIFY A “NEED” AND IMPLEMENT IT

If only I had a “length” function

```
(define (length l)
  -1)

(if (not (equal? 0 (length '()))) (print "err1"))
(if (not (equal? 1 (length '(a)))) (print "err2"))
(if (not (equal? 2 (length '(a b)))) (print "err3"))
(if (not (equal? 3 (length '(a b c)))) (print "err4"))
```

```
(define (length list)
  (define (length-reduce list num)
    (if (equal? list '())
        num
        (length-reduce (cdr list) (+ num 1))))
  (length-reduce list 0))
```

RE-ITERATE

If only, I had an “nth” for a list

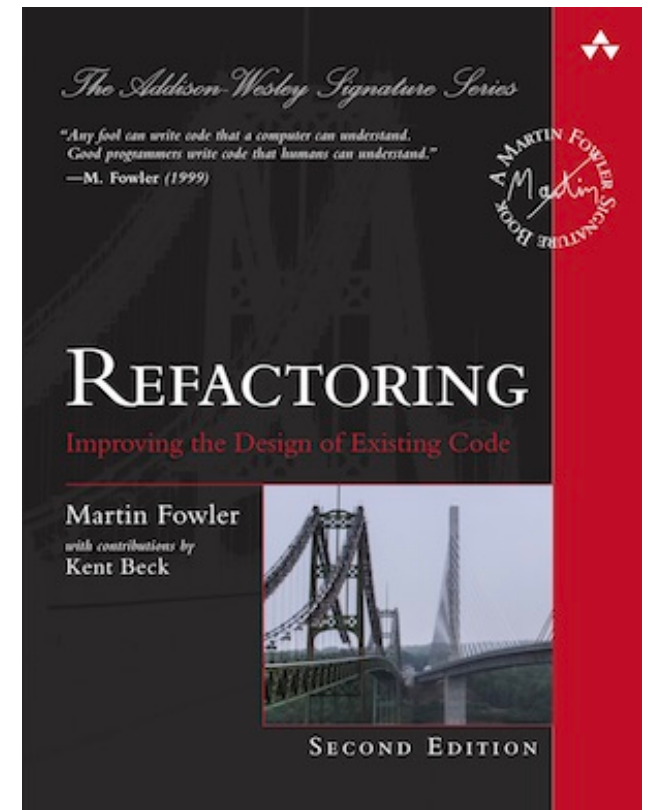

```
(define (nth l i)
  -1)

(if (not (equal? 'a (nth '(a b c) 0))) (print "err1"))
(if (not (equal? 'b (nth '(a b c) 1))) (print "err2"))
(if (not (equal? 'c (nth '(a b c) 2))) (print "err3"))
(if (not (equal? '() (nth '(a b c) 3))) (print "err4"))
(if (not (equal? '() (nth '(a b c) -1))) (print "err5"))
```

```
(define (nth list i)
  (define (nth-reduce list current-offset)
    (cond
      ((equal? list '()) '())
      ((= i current-offset) (car list))
      (else (nth-reduce (cdr list) (+ current-offset 1)))))
  (nth-reduce list 0))
```

REFACTORING

- ▶ Not just buzz word, but a real process
- ▶ Principle approach; move separate from change
 - ▶ Clean up code separately trying to make it better
 - ▶ Do not "move code" and "change the sense of the code" at the same time



SIDE EFFECTS OF EFFICIENT ASSIGNMENT

Professors would/could/should be as efficient with the assignments as possible

- ▶ Easier to mark
 - ▶ Harder to make mistakes marking
- ▶ Faster to mark
 - ▶ More time spent developing awe-inspiring lectures and labs
- ▶ Consistency is the key
 - ▶ Student numbers in file submissions (for easy identification)
 - ▶ Submissions as one file (in a ZIP if multiple files required)
 - ▶ Named well for sorting (e.g. **lab02** instead of **lab2** just in case we have a lab10)
 - ▶ Sorted for groups (e.g. **lab01_1111111_2222222.zip** instead of **lab01_2222222_1111111.zip** for sorting purposes)

MAKE IT EASY FOR THE MARKER TO
GRADE YOUR WORK

**IMAGINE IF YOU HAD TO MARK 130 ASSIGNMENTS
WITHOUT CONSISTENCY IN THE CEREMONY.**

CONSLUSION

- ▶ Start with the template that requires little to no “real thought” for your homework from the very start
- ▶ Always make it “ready” to submit
- ▶ Stop writing un-compile-able code, just stop