

FOURTH QUIZ

You have 15 minutes from the start of class to complete this quiz. Read the problems with care; work with deliberate speed. Don't give us more than we ask for. The usual instructions apply. Good luck!

Problem 1 (4 points)

The Anteater Bookstore wants to computerize its inventory. Write a Scheme structure definition for a structure that represents a book with four fields: its title (a string), its author (a string), its year of publication (a number), and its price (a number).

(a) (2 points) Just write the structure definition.

(b) (2 points) Write a Scheme expression that creates a book structure for the book *War and Peace* by Leo Tolstoy, published in 1869 and selling for \$13.60.

Problem 2 (4 points)

In this problem you may represent lists in any of the three ways we showed in class, as illustrated below:

```
(define L (cons "Wilson" (cons "Davis" (cons "Schwarzenegger" (cons "Brown" empty)))))  
(define L (list "Wilson" "Davis" "Schwarzenegger" "Brown"))  
(define L '("Wilson" "Davis" "Schwarzenegger" "Brown"))
```

What is the value of each of these expressions? Use the above definition independently for each part.

(a) `(first L)`

(b) `(rest L)`

(c) `(rest (rest L))`

(d) `(cons "Current:" (rest (rest (rest L))))`

Problem 3 (12 points)

Anteater Classic Cars sells vintage automobiles, representing each car using a structure defined as follows:

```
(define-struct auto (maker model year price))
```

The maker (manufacturer) is a string, as is the model name; the year the car was produced and its price are numbers.

(a) (3 points) Fill in the blanks in the following definition, consistent with the contract and purpose statement. Each blank contains one function or parameter.

```
;; between-years?: auto number number -> boolean
;; Return true if the auto's year is at least the first number and at most the second
(check-expect (between-years? (make-auto "Ford" "Falcon" 1960 15000) 1960 1970) true)
(define between-years?
  (lambda (A year1 year2)
    (and
      (_____ (_____ A) _____)
      (_____ (_____ A) _____))))
```

(b) (4 points) Complete the definition of the function `replace-maker` as described below.

```
;; replace-maker: auto string -> auto
;; Return an auto like the input but with its maker replaced with the input string
(check-expect (replace-maker (make-auto "Chrysler" "Imperial" 1955 20000) "Daimler")
  (make-auto "Daimler" "Imperial" 1955 20000))
(define replace-maker
  (lambda (A new-maker)
```

(c) (5 points) Cars made by Nissan were called Datsun in the U.S. between 1933 and 1986. In our list of cars, we want to change the maker from "Nissan" to "Nissan/Datsun" for any Nissan cars made in that period. Fill in each blank with a function name, parameter name, or constant value, using functions defined elsewhere in this quiz where appropriate. (Note that you don't need to supply any of the list-processing code.)

```
;; add-Datsun: list-of-auto -> list-of-auto
;; Return the input list changed as follows: Cars made by Nissan between 1933 and
;; 1986 should have their maker relabeled "Nissan/Datsun".
(define add-Datsun
  (lambda (L)
    (cond
      ((empty? L) empty)
      ((and (string=? (auto-maker (first L)) "Nissan")
            (_____ (first L) 1933 1986))
        (cons (_____ (first L) _____) (add-Datsun (rest L))))
      (else (cons (first L) (add-Datsun (rest L)))))))
```