

# THIRD 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 (2 points)

Evaluate each of the following expressions. (That is, what does DrScheme display in the interactions window when you enter the expression(s) in the definitions window and click Run?) Recall that `even?` and `odd?` are predefined predicates that tell whether a number is even or odd, respectively.

```
(define LAB1 22)
(define LAB2 29)
```

(a) `(>= (+ LAB1 LAB2) 50)`

(b) `(and (even? LAB1)
 (even? LAB2))`

(c) `(string-append "Good "
 (cond
 ((even? (/ 50 2)) "Morning")
 (else "Day")))`

## Problem 2 (4 points)

Evaluate each of the following expressions. Use this definition independently for each of the five parts:

```
(define L (cons "David" (cons "Joel" (cons "Wiwat" (cons "Jordan" (cons "Grace" empty))))))
```

(a) `(first L)`

(b) `(first (rest L))`

(c) `(rest L)`

(d) `(first (rest (rest L)))`

(e) `(cons "Tutors:" (rest (rest (rest L))))`

## Problem 3 (12 points)

We run the Anteater Fruit Farm and sell our produce at local farmers' markets. Our sales management software defines a fruit structure as follows to hold each kind of fruit we grow:

```
(define-struct fruit (name howmany price organic?))
```

where `name` is a string representing a kind of fruit, `howmany` is the number of pieces we brought to the market, `price` is the number of dollars we charge per pound, and `organic?` is a boolean that's true if this fruit was organically grown and false otherwise.

So you don't have to keep flipping the page over: `(define-struct fruit (name howmany price organic?))`

(a) (2 points) We have harvested 450 organic apples for this week's market and we plan to sell them for \$2.50 a pound. Write a Scheme expression to create a fruit structure that represents this.

(b) (2 points) Define the function `sale-price` as described below.

```
;; sale-price: fruit number -> number
;; The input is the number of pounds of this fruit a customer has bought.
;; Return the amount the customer must pay for this purchase.

(define sale-price
  (lambda (F weight)
    (let ((price-per-pound (lookup F)))
      (* price-per-pound weight))))
```

(c) (3 points) Define the function `fruit-update-inventory` as described below.

```
;; fruit-update-inventory: fruit number -> fruit
;; Someone buys the specified number of pieces of this fruit. Return the fruit
;; structure updated to reflect the decreased number you now have to sell.

(define fruit-update-inventory
  (lambda (F numsold)
```

(d) (5 points) Our farmer's market adds a 10% surcharge on the price of every sale of non-organic produce; organic produce has no surcharge. Define the function `sale-surcharge` as described below. Where applicable, use functions you have already defined rather than duplicating code.

```
(define SURCHARGE-RATE 0.10)
;; sale-surcharge: fruit number -> number
;; The input is the number of pounds of this fruit a customer has bought.
;; Return the surcharge the customer must pay for this purchase.

(define sale-surcharge
  (lambda ( F weight )
    (cond
```

#### Problem 4 (2 points)

In class we discussed circuit-switched networks and packet-switched networks.

(a) A transmission typically gets split up into pieces that may take different routes to the destination. Does this describe circuit-switching or packet-switching?

(b) One describes message routing on the internet; the other describes conventional telephone service. Which is which?