You have 15 minutes from the start of class to complete this quiz. Read the questions 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)

Evaluate each of the following expressions. The functions even? and odd? are predefined in DrScheme; even? returns true if its argument is an even number and odd? returns true if its argument is odd.

(a) (define take-temperature
    (lambda (temp)
      (cond
       ((not (number? temp)) "Bad input")
       (>= temp 100) "High fever")
       (> temp 98.6) "Fever")
       (= temp 98.6) "Normal")
       (else "Below normal")))

(b) (and (number? 'in4matx) (even? 4))

(c) (not (or (odd? 18) (even? 19)))

(d) (* 4
    (cond
     ((string=? "Hello" "Hola") 6)
     ((symbol=? 'Aloha 'Aloha) 5)
     (else 10)))

Problem 2 (5 points)

Complete the following function definition according to the contract and purpose given.

;; weight-per-inch: symbol -> number
;; If input is 'maple, return 1; if it’s ‘aluminum, return 0.7;
;; if it’s ‘plastic, return 0.2; otherwise return zero.
(define weight-per-inch
    (lambda (material)
Problem 3 (16 points)

(a) (2 points) Define a structure to represent a picture frame with a width, a height, a material, and a weight. The width and height will be numbers of inches, the material will be a symbol, and the weight will be a number of ounces.

(define-struct picture-frame (width height material weight))

1 point for the right idea, 1 more point for everything correct (name can be “frame” instead of “p-f”)

(b) (2 points) Write an expression that constructs and returns a picture frame made of maple that’s 24 inches tall, 36 inches wide, and weighs 96 ounces. (Pay close attention to what we’re asking for.)

(make-picture-frame 36 24 'maple 96) — 1/2 for width and height reversed

(define my-frame (make-picture-frame 36 24 'maple 96)) okay if followed by my-frame; otherwise —1/2.

(c) (4 points) Complete the following function definition according to the contract and purpose given.

;;; picture-frame-perimeter: picture-frame -> number
;;; Compute the total length of framing material necessary for the frame (top, bottom, and both sides)
;;; Example: (picture-frame-perimeter (make-picture-frame 5 7 ‘plastic 19)) returns 24
(define picture-frame-perimeter
  (lambda (PF)
    (+ (* 2 (picture-frame-width PF)) (* 2 (picture-frame-height PF))))

Scoring: 2 points total for correct computation; 2 points total for Scheme syntax

(d) (8 points) Complete the following function definition according to the contract and purpose given. Where appropriate, use the functions you defined above.

;;; frame-with-weight: picture-frame -> picture-frame
;;; Return a (newly created) picture-frame based on the input frame, but with the weight calculated depending on the frame’s perimeter and material.
;;; Example: (frame-with-weight (make-picture-frame 5 7 ‘plastic 19)) returns (make-picture-frame 5 7 ‘plastic 4.8)
(define frame-with-weight
  (lambda (PF)
    ...))