Programming Paradigms
CISC-280

Example Midterm I

NAME:

There are 100 points on 7 pages.

10 points

Write the value of each of the following lisp expressions. If the expression is an ill formed lisp expression, write "error".

_____ `(- (/ (* 5 4) 2) 4)`

_____ `(/ (* 4 2) (+ 2 2))`

_____ `(/ (* (+ (2 3) 3 1)))`

```
(define a 4)
```

```
(define b (+ a 1))
```

_____ `(= a b)`

_____ `(if (and (> a b) (< b (* a b))) b a)`

9 points

Here is a simplified version of the second order procedure for summation that we studied.

```
(define (sum term a b)
  (if (> a b)
      0
      (+ (term a) (sum term (+ a 1) b)) ))
```

_____ What is the value of `(sum square 3 5)`, if `square` is defined by `(define (square x) (* x x))`?

_____ What is the value of `(sum (lambda (i) 2) 1 50)`?

_____ What is the value of `(sum 2 1 50)`?

9 points

For each of the following procedures, indicate whether it is (i) tree recursive, (ii) linear recursive, or (iii) tail recursive (iterative process).

A:
```
(define (gcd a b)
  (if (zero? b)
      a
      (gcd b (remainder a b)) ))
```
_____

B:
```
(define (pt a b)
  (if (or (= a 1) (= a b))
      1
      (+ (pt (- a 1) (- b 1)) (pt (- a 1) b)) ))
```
_____

C:
```
(define (fibonacci-modified a b k n)
  (if (= k n)
      b
      (fibonacci-modified b (+ a b) (+ k 1) n) ))
```
_____

12 points

Evaluate the following, which use the procedures defined in the previous question.

A: `(gcd 60 40)` _____

B: `(pt 3 1)` _____

C: `(fib-modified 0 1 1 3)` _____

D: The n-th fibonacci number can be computed by `midtermfib` if we define it as:

```
(define (midtermfib n) (fib-modified 1 1 1 n))
```

In the textbook, we computed the n-th fibonacci number by using the function `fib`, which is tree-recursive:

```
(define (fib n)
  (cond ((= n 0) 0)
        ((= n 1) 1)
        (else (+ (fib (- n 1))
                 (fib (- n 2))))))
```

How does the running time of our new definition `midtermfib` compare with the running time of the tree-recursive definition `fib` from the text? (Don't worry—you do not have to give a precise answer!)

9 points

Test your Scoping knowledge! first, assume we make these definitions

```
(define x 3)
(define y (* x 2))
(define z (+ x y))
(define (foo x y)
  (+ x y z))
(define (foobar a b)
  (let ((x (+ a b))
        (y (+ x b)))
    (+ x y z)))
```

Now, Evaluate the following Scheme expressions:

_____ `(foo 5 3)`

_____ `(foo x y)`

_____ `(foobar 4 3)`

18 points (8+8+2)

PART A. Write a procedure called `dots` that takes one argument, `n`, and displays n dots on the screen. The procedure `dots` can return any value you want. Hint: remember that you can do more than one thing in a COND clause.

PART B. Write a procedure called `plot` that takes three arguments: a function f to plot, a start point a , and an end point b. You may assume that the start and end points are integers. The `plot` procedure should plot the number of dots equal to the value of f at the integers starting at a and going through b, inclusive. Of course, you should use your `dots` procedure from Part A. Print each line of dots on a line by itself. For example, if the function f is `square`, and a is 5, then the first line of dots will have 25 dots on it.

PART C. Write how you would call your `plot` procedure in order to plot the function $f(x) = x^2 + 7x$ from 1 to 10. Do this in one line, using `lambda` to define the function.

9 points (7 + 2)

Procedures that return Procedures. We can "smooth" a continuous function by changing the value of the function at each point to be the average of nearby points. Assume we have a function $f(x)$, and a "nearby" distance d Then

$$\text{smooth}(x) = \frac{f(x-d) + f(x) + f(x+d)}{3}$$

Part A. Write `smooth` in Scheme. The inputs are a function of one argument, and a distance d. The output should be *a new function of one argument*, as defined above. Hint, use `lambda`. I'll start you off:

```
(define (smooth f d)
```

Part B. Show what you would type in the Scheme interpreter to find the smoothed value of `sin(x)` at x=2, for distance = 0.01. You don't have to show what DrScheme's result would be, of course!!

24 points

Part A:

6 pts. Give the combinations of cars and cdrs that will pick out the number "7" from each of the following (a different answer for each one, of course!):

- `(1 (5 7) 8)`
- `((7))`
- `(1 (2 (7)))`

Part B:

8 pts. I'm writing a parts database. Each part has a unique ID number, a price, and a location bin number. All three values are numbers: the ID and bin number are integers, and the price is a real number. Design an abstract data type for a "part" by defining the appropriate constructor and selectors. Use a list as your implementation.

Part C:

10 pts. Draw the box and pointer diagram resulting from the following definitions. Please indicate what x and y are pointing to as well.

```
(define x (list (cons 20 null) 4 10))
(define y (cons 12 (append (quote (dog cat)) x)))
```