CISC280 partial Midterm 1, Fall 2004

NAME _____

General instructions:

There are like, some problems worth points, and an extra credit problem worth 5 points. Read the problems very carefully. Identify what kind of answer the problem asks for. If writing a procedure, carefully look at requirements for input and output, and any restrictions on how it must be written or which other procedures may be used.

You may assume a list argument will be flat unless it is otherwise specified, and that the elements will not produce errors for the procedures described.

Do not do unnecessary testing. For example, testing for both list? and null? instead of using one test and then else would be considered unnecessary testing.

Do not make code unnecessarily inefficient. An extra procedure call here or there is ok, but do not make an $O(n)$ problem into $O(n^2)$.

Do problems you are confident about first. If you finish the problems you know, write what you do know about other problems to gain partial credit; but erroneous information may detract from that credit, so don't make stuff up.

You may use any of the Scheme primitives we use in class.

1. 16 pts. Show what is returned by the interpreter in response to each of these expressions. Each is to be evaluated by itself. If the result is an error, or other interpreter message, you do not have to be exact. For example, if the expression were the input "expt", you could show the interpreter's response as "primitive" instead of writing #<primitive:expt>.

   (a) (cons a b)

   (b) (cons () (quote b))

   (c) (cons () ())

   (d) (cond (#t + 3 5 7))

   (e) (cdr ())

   (f) (quote +)

   (g) (cons (list 1 2 3)())

   (h) (cadr (cons 7 2))

2. 20 pts. Write the function `interleave` that merges two lists, alternating the members. If one list is longer, its extra elements simply get added to the end. The first argument's element goes first.

```
> (interleave ()'(3))
(3)
> (interleave '(3 5 7) '(4 6 8))
(3 4 5 6 7 8)
```

3. 21 pts. As discussed in class, what three elements are shared by all programming languages? List them here. Give **examples** of each in both Scheme and either C, C++, or Java. Did you write nine things?

   (a)

   (b)

   (c)

4. 14 pts. Write a tail-recursive procedure to compute `factorial` of an integer argument `n`, where $factorial(n) = n * n - 1 * ... * 1$

5. 5 pts. Draw the box and pointer structure for (((1 2) 3) 4 5).

6. 5 pts. Write the Scheme representation of the box and pointer structure shown.

7. 10 pts. Write `accumulate`.

8. 7 pts. Using only calls to sequence operators, define a procedure `somof` that takes an integer `n` as argument and returns the sum of the first `n` positive multiples of 5, starting with zero. You do not need to define the sequence ops you use. You may define small lambdas to feed to your sequence ops.

```
> (somof 5)
75
```

9. 3 pts. Write a single call to `accumulate` that sorts a list. Write any supporting procedures you need.

10. 5 pts Extra Credit: Paul Graham accomplished amazing things in two different industries. What were the industries? What did he do?