CS 60A Midterm #1 — February 10, 1992

Your name _____

login c60a–_____

Discussion section number _____

TA's name _____

This exam is worth 20 points, or 12.5% of your total course grade. The exam contains four substantive questions, plus the following:

Question 0 (1 point): Fill out this front page correctly and put your name and login correctly at the top of each of the following pages.

This booklet contains five numbered pages including the cover page. Put all answers on these pages, please; don't hand in stray pieces of paper. This is an open book exam.

When writing procedures, don't put in error checks. Assume that you will be given arguments of the correct type.

Our expectation is that many of you will not complete one or two of these questions. If you find one question especially difficult, leave it for later; start with the ones you find easier.

0	/1
1	/4
2	/5
3	/5
4	/5
total	/20

Question 1 (4 points):

What will Scheme print in response to the following expressions? Assume that they are typed in sequence, so definitions affect later interactions. If an expression produces an error message, you may just say "error"; you don't have to provide the exact text of the message.

```
(* (+ 2 3) (- 7 (* 5 0)))
```

```
(if (+ 2 3) (+ 4 5) (+ 6 7))
```

```
(define six 6)
```

```
(let ((- +)
      (ringo six))
  (- six ringo))
```

```
(- 6 4)
```

```
(first (butlast (last (butlast '(the long and winding road)))))
```

```
(+ (bl 472) (first 38))
```

```
(lambda (a b) (word b (first a)))
```

```
((lambda (d c) (word (last c) d)) 'here 'not)
```

Your name _____ login c60a–_____

Question 2 (5 points):

Write a procedure `add-numbers` that takes a sentence as its argument. Some of the words in the argument sentence might be numbers. The procedure should return the sum of those numbers. For example:

```
> (add-numbers '(8 days 1 week))
9
> (add-numbers '(76 trombones, 4 calling birds, and 2 turtle doves))
82
> (add-numbers '(all you need is love))
0
```

You may use the primitive predicate `number?` in your procedure.

Question 3 (5 points):

Implement the following function as a Scheme procedure:

$$f(a,b) = \begin{cases} 1, & \text{if } b = 0; \\ a \times f(a, b-1), & \text{if } b > 0. \end{cases}$$

Also, explain in one English sentence the mathematical meaning of this function.

Your name _____ login c60a–_____

Question 4 (5 points):

Consider the function

```
(define (reciprocal x) (/ 1 x))
```

This function doesn't work too well if its argument is zero. Suppose we want the function described in words this way: "If the argument is zero, return zero; otherwise, use reciprocal." We'd like to be able to create that function this way:

```
(define reciprocal-unless-zero (choose-function zero?
                                                (lambda (x) x)
                                                reciprocal) )
```

Write the function `choose-function`. It takes three arguments, all of which are functions of one argument. The first argument is a predicate function. Let's call the arguments `pred`, `f1`, and `f2`. `Choose-function` returns a new function of one argument, such that if `pred` of that argument is true, the value is the result of applying `f1` to the argument; if not, the result of applying `f2`.

```
> (define funny-fn (choose-function even? first last))
> (funny-fn 376)
3
> (funny-fn 495)
5
```