Sample Questions for Exam 1

This document contains some sample questions for the first exam, which will be given in class on Thursday 10/3. Note that the exam will have more questions than appear on this sample. The exam covers material from classes up to and including 9/19 and problem sets 1 and 2.

You may bring one handwritten sheet of notes to the exam with you.

Problem 1

What is the result of the evaluation of the final statement in each of the following groups of Scheme expressions? Write “error” if an error would result or “procedure” if a procedure would be returned. Assume that each group is evaluated separately.

```
(define a 1)
(define b 3)
(define c 5)

(let ((a 2)
      (b (+ a 5))
      (c b))
  (+ a b c))
```

```
(define (my-func f)
  (lambda (x y) (f (f x y) (f x y))))

((my-func *) 2 3)
```
Problem 2

a. Define applicative order evaluation

b. Define normal order evaluation

c. Given the following code,

```
(define (add a b)
  (display " plus ")
  (+ a b))

(add (begin (display " one ")
  1)
  (begin (display " two ")
    2))
```

Which of the following could be printed in Scheme? In normal-order Scheme?

<table>
<thead>
<tr>
<th>Scheme</th>
<th>Normal-order Scheme</th>
</tr>
</thead>
<tbody>
<tr>
<td>plus one two</td>
<td></td>
</tr>
<tr>
<td>plus two one</td>
<td></td>
</tr>
<tr>
<td>one plus two</td>
<td></td>
</tr>
<tr>
<td>two plus one</td>
<td></td>
</tr>
<tr>
<td>one two plus</td>
<td></td>
</tr>
<tr>
<td>two one plus</td>
<td></td>
</tr>
</tbody>
</table>
Problem 3

For the two expressions below, write the sequence of cars and cdrs needed to get the number 3 out of data structure. You may find it helpful to draw the box-and-pointer diagrams, but you will not be graded on them.

(define first (cons 1 (cons 2 (cons 3 4))))

(define second (cons (cons 1 2) (cons 3 4)))
Problem 4

Write a function `apply-twice` that takes a function \( f \) as its argument and returns a function that takes one argument as input and returns the value that one would obtain if \( f \) were applied twice to that argument.

For example,

\[
((\text{apply-twice square}) 2)
\]

would return

16

and

\[
((\text{apply-twice (lambda (x) (+ x 2))) 5})
\]

would return

9
Problem 5

Write a procedure \texttt{prod} that takes two parameters representing a range from \(a\) to \(b\). The procedure should return the value of \(a \times (a+1) \times (a+2) \times \ldots \times b\). For example,

\[(\text{prod } 1 \ 5)\]

will return

\[120\]

What is the order of growth of your \texttt{prod} procedure in terms of time and space?

Time: _________________________________

Space: _________________________________

What is \(n\) dependent upon?

Does your definition of \texttt{prod} generate a recursive or an iterative process?
Problem 6

Given the following definition of cons, write the definitions of car and cdr.

\[
\text{(define (cons x y)} \\text{)} \\
\text{(lambda (m)} \\
\text{(cond ((= m 0) x)} \\
\text{((= m 1) y)} \\
\text{(else (error “Unknown message – CONS” m)))))}
\]