Programming Paradigms (COMP 3007)
Midterm Exam I

Carleton University
School of Computer Science
Fall 2012

Name: _____ **Student #:** _____

1. What will Scheme print in response to the following expressions (15 marks):

a.
```
(if (and (> 2 1) #f) (* 3 2) 8)
```

8

b.
```
(define x 1)
(let ((x 5) (y (+ x 2)))
     (* x y))
```

15

c.
```
(map sqr (filter odd? (list 2 4 5)))
```

'(25)

d.
```
(cadr (cons 1 (cons 2 null)))
```

2

e.
```
((lambda (f) (f (list 5 2))) (lambda (x) (car x)))
```

5

2. **If f is a numerical function and n is a positive integer, then we can form the nth repeated application of f, which is defined to be the function whose value at x is f(f(...(f(x))...)). For example, if f is the function f(x) = x + 1, then the nth repeated application of f is the function f(x) = x + n. If f is the operation of squaring a number, then the nth repeated application of f is the function that raises its argument to the 2^nth power.**
   Write a procedure that takes as inputs a procedure that computes f and a positive integer n and returns the procedure that computes the nth repeated application of f. (20 marks)

   Your procedure should be able to be used as follows:

```
((repeated sqr 2) 5)
625
```

```
(define (repeated f n)
  (lambda (x)
    (if (= n 1)
        (f x)
        (f ((repeated f (- n 1)) x)))))
```

3. **Consider the following mathematical function:**

$$f(n) = \begin{cases} n & if\ 1 \le n \le 3 \\ f(n-3) + 2f(n-1) & if\ n > 3 \end{cases}$$

a. **Write a procedure that computes f by means of a recursive process (10 marks)**

```
(define (f n)
  (if (and (> n 0) (< n 4))
      n
      (+ (f (- n 3)) (* 2 (f (- n 1))))))
```

b. **Write a procedure that computes f by means of an iterative process (10 marks)**

```
(define (f n)
  (define (f-iter n a b c)
    (if (and (> n 0) (< n 4))
        c
        (f-iter (- n 1) b c (+ a (* 2 c)))))
  (f-iter n 1 2 3))
```

4. **We want to create a data structure for rectangles in a plane:**

   a. **Choose a representation for this data structure and write constructor and accessors for your representation (15 marks)**

```
(define (rect w h)
    (cons w h))

(define (width r)
    (car r))

(define (height r)
    (cdr r))
```

   b. **Create procedures that compute the perimeter and the area of a given rectangle (10 marks)**

```
(define (area r)
    (* (width r) (height r)))

(define (perimeter r)
    (* 2 (+ (width r) (height r))))
```

c. **Create a "message passing" implementation of the rectangle data structure (20 marks)**

```
(define (make-rect w h)
  (define (dispatch op)
    (cond ((eq? op 'width) w)
          ((eq? op 'height) h)
          (else error "wrong parameter")))
  dispatch)
```