Higher Order Procedure Notes

1. Using higher order procedures: *procedures that return procedures*

```
(* 2 5)          (* 3 2)
(* 2 8)          (* 3 4)
```

```scheme
(define (double n)
  (define (triple n)
    (* n 2)
    (* n 3))

(define (make-mult x)
  (lambda (n)
    (* n x))

(define double
  (define triple
    (make-mult 2)
    (make-mult 3))
```

2. Using higher order procedures: *procedures that take procedures as arguments*

very useful hops: map, filter, (fold-right and fold-left next recitation)

examples:

```
(map double (list 1 2 3 4)) => (2 4 6 8)

(filter even? (list 1 2 3 4)) => (2 4)
```

```scheme
(define (map op items)
  (if (null? items)
      '()
      (cons (proc (car items))
            (map proc (cdr items))))

(define (filter pred items)
  (cond ((null? items) '())
        ((pred (car items)) (cons (car items) (filter pred (cdr items))))
        (else (filter pred (cdr items)))))
```