

Recitation 7, March 2

Higher Order Procedure Notes

Dr. Kimberle Koile

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

```
(* 2 5)  
(* 2 8)
```

```
(* 3 2)  
(* 3 4)
```

```
(define (double n)  
  (* n 2)  
)
```

```
(define (triple n)  
  (* n 3)  
)
```

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

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

```
(define triple  
  (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)
```

```
(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)))))
```