

Scott Waldron  
 11/29/16  
 CSCI 301  
 Lab 9

### 1. What are total functions?

Total functions are all the functions we have seen so far up until chapter 9. They are not partial and won't loop infinitely if there is a unnatural loop within. They will always return a result or complete a task.

### 2. What are partial functions?

Partial functions are functions the recur unnaturally. Like eternity for example, it is the most partial function. In short, there can be a sequence in which a partial never stops looping.

### 3. What is the general purpose of looking functions and what kind of function (total or partial) are they?

The general purpose of looking functions are to print #t or #f based on whether the expression exists within the lat. They are partial functions because the recur unnaturally and get caught in an infinite loop.

### 4. Use three lambda statements to define a Scheme function that determines the length of lists with two or fewer arguments.

```
(lambda (l)
  (cond
    ((null? l) 0)
    (else
     (add1
      ((lambda (l)
         (cond
           ((null? l) 0)
           (else
            (add1
             ((lambda (l)
                (cond
                  ((null? l) 0)
                  (else
                   (add1
                    (eternity
                     (cdr l))))))
              (cdr l))))))
        (cdr l))))))
```

### 5. Use two lambda statements to define a Scheme function that determines the length of lists with two or fewer arguments.

```
((lambda (mk-length)
  (mk-length
   (mk-length
    (mk-length eternity))))
 (lambda (length)
  (lambda(l)
   (cond
    ((null? l) 0)
    (else (add1 (length (cdr l))))))))
```