H211 Fall 2015 09/03 Attendance and Worksheet for (Name/Username): _________________________

Last time we met the design recipe for the first time. In Lab Two one of your goals will be to write (or should I say, design) a function that calculates the amount of time between two time representations. Here’s the design recipe without the template and the code:

; time-between :  String String -> String
; calculates the difference between t1 and t2 (in hours and minutes)
; given: "09:12" for t1 and "09:41" for t2
; expected: "0 hours and 29 minutes."
; (define (time-between t1 t2)
;   ... ) ; you need to provide this part

(check-expect (time-between "09:12" "09:41") "0 hours and 29 minutes.")
(check-expect (time-between "09:12" "12:05") "2 hours and 53 minutes.")

It is quite unlikely that you will be able to knock this problem off in only one stage with what you know so far so in this lecture we will learn to break the problem in smaller problems, solve those, then compose the solutions to the smaller problems into one big solution for the initial, larger problem. The reading assignment is 2.3 Composing Functions\(^1\) in your (online) text.

1. Start by defining a function that takes a time representation and extracts the number of hours in it.

; extract-hours :  String -> Number
; extracts the number of hours in the time representation time
; given: "12:03" expected: 12
; (define (extract-hours time)
;   ... ) ; you need to provide this part (template and actual code)

(check-expect (extract-hours "12:03") 12)
(check-expect (extract-hours "09:12") 9)}

Time representations in this problem are in military time format, that is, 5:12pm is written as "17:12". It follows that accepted time representations are in between "00:00" representing midnight to "23:59" representing one minute before midnight. Even this simple problem will have to be solved using a composition of other, preexisting functions. You might need: string->number, substring.

2. Next define a similar function that extracts the minutes from a time representation:

; extract-minutes : String -> Number
; extracts the number of minutes in the time representation time
; given: "09:12" expected: 12
; (define (extract-minutes time)
;   ... ) ; you need to provide this part

(check-expect (extract-minutes "12:03") 3)
(check-expect (extract-minutes "09:12") 12)

\(^1\)http://www.ccs.neu.edu/home/matthias/HtDP2e/part_one.html#%28part._sec-3acomposing%29
3. If we have the hours and the minutes (as numbers) in a time representation we can convert that pair of values into a number of minutes. As an example if the two numbers are 9 (for hours) and 51 (the minutes) we would convert that into a value representing 9 hours and 51 minutes which comes to 591 minutes. Define/design a function that implements this conversion:

```scheme
;; hours-and-minutes->minutes : Number Number -> Number
;; converts a pair of numbers (hours, minutes) into a total
;; duration of time (expressed in minutes) see examples below
;; given: 2 12 expected: (+ 12 (* 2 60))
;; given: 1 59 expected: 119
(define (hours-and-minutes->minutes hours minutes)
  ...) ; you need to provide this

(check-expect (hours-and-minutes->minutes 2 12) 132)
(check-expect (hours-and-minutes->minutes 1 59) 119)
```

You might need: +, * and 60.

How far are we now from the initial target? How can we bridge the gap?

Below please use the space to write a short summary entitled *Previously on H211...*

Describe succinctly what you have learned thus far. Don’t use more than the space provided.