**ICS 365: Organization of Programming Languages**

**Programming Assignment 4**

**Due**: See Syllabus

**Points**: 40

**Goal: Understand scheme and aspects of functional programming languages.**

**Part 1:**

1. **Install a scheme interpreter. I strongly recommend Dr. Racket, but you can use anyone you want.**

**https://racket-lang.org/ . Dr. Racket has support for Windows, MAC, and linux.**

1. **Watch Scheme how-to videos.**

<https://www.youtube.com/watch?v=6k78c8EctXI>

**<https://www.youtube.com/watch?v=AqBxU-Zmx00>**

1. (10 points)

Write a scheme function named up-to-first-char that takes a list as its input and returns a list containing all the elements up to the first character element in the input list. Hint: think of using some predicate function(s) in your logic.

Sample runs:

(up-to-first-char '(1 2 3 4 5 a b c)) returns (1 2 3 4 5)

(up-to-first-char '(8 7 5 f)) returns (8 7 5)

(up-to-first-char '(1 2 9)) returns (1 2 9)

(up-to-first-char '(z w x)) returns NIL

1. (15 points)

Write a Scheme function that takes a list and an atom as parameters and returns a list identical to its parameter list except with all top-level instances of the given atom deleted. Hint: recursive algorithm.

1. (15 points)

Write a Scheme function that takes two atoms and a list as parameters and returns a list identical to the parameter list except all occurrences of the first given atom in the list are replaced with the second given atom, no matter how deeply the first atom is nested.

**Hand in: code files, these can be text files that have your code in them. Also, screen shots of the functions running and doing what they are supposed to do (i.e. proving your logic/code works). Do more than 1 screen shot for each one, using different inputs for each run. No ZIP files please.**