CS 370 Introduction to Teaching Computer Science
Fall 2018 Christopher Hunn, Sequoia Eyzaguirre, & Bryan Tong

Station 1

Name:

Section:

At this station, you will design questions that target mechanics and edge cases. Below, you'll find some common edge cases in CS 61A that often go unnoticed until it's too late.

- `(define a 3)` versus `(define (a) 3)`
- Order of evaluation: operator, operands, apply. (Consider using `global` or `nonlocal`.)
- Class / instance attributes.
- Negative list indexing!
- Things that copy a list, versus things that mutatie a list. (For example `A = A + B`.)
- `__str__` and `__repr__`.
- The `lambda`'s parent in `f(lambda...)`.
- When does a Scheme promise get computed / re-computed?
- Why / when we need a base case.
- Order of recursive calls. (For example *do stuff, recurse, recurse* versus *recurse, do stuff, recurse*.)

1. **Task 1**
   For as many of the edge cases above as you can, design a new practice problem. If a student completed your practice problem, they should come away with a clear understanding of how to handle that edge case should they come across it in the future. Remember, quality over quantity! Feel free to move on to other tasks once you have come up with 2 or 3 or problems. Don't feel compelled to work in the same order they're listed above.

2. **Task 2**
   Now write a clear explanation for each problem you came up with. It should explain why the solution is correct, and briefly what the thought process is to get there from scratch.

3. **Task 3**
   Brainstorm 3 different edge cases, not listed above. They can be edge cases that you encountered as a student, or while teaching this semester. Repeat **Task 1** for the edge cases that you come up with.