

CS 2500, Spring 2013
Problem Set 6

Due date: Tuesday, February 19 @ 11:59pm

Programming Language: Beginning Student Language with List Abbreviations

What to submit:

You must work on this problem set in pairs. Homework partners have been chosen randomly and posted on Blackboard. You must work on and submit the homework with your partner. Using Blackboard, submit a single Racket file containing all of the code and documentation for this assignment. Place your names and husky email addresses in a comment at the beginning of your file.

Name your file: hw6-lastname1-lastname2.rkt

This problem set continues the study of self-referential unions. Some of the problems cover functions that process two arguments from classes with complex (self-referential) data definitions. You must follow the design recipe in your solutions: graders will look for data definitions, contracts, purpose statements, examples/tests, and properly organized function definitions. For the latter, you must follow templates. You do not need to include the templates with your homework, however, unless the question asks for it.

Problem 1.

In this problem, you will implement functions for a simple version of a social networking system such as Facebook.

- a) A `profile` consists of the user's name, location and relationship status and a `lof` (list of friends). A `friend` consists of a name, location and relationship status. Write data definitions and provide examples of data for `profile`, `friend`, and `lof`.
- b) Write the templates for `profile`, `friend` and `lof`.
- c) Write a function, `total-friends`, that consumes a `profile` and produces the total number of friends that the user has.
- d) Write a function `add-friend` that consumes a `profile` and the `friend` to add, and returns a `profile`. If the `friend` is not in the `lof` of the `profile`, then the `friend` is added to the `lof`. Otherwise, the `profile` is returned unchanged.
- e) Write a function `un-friend` that consumes a `profile` and the `friend` to delete, and returns a `profile`. If the `friend` is in the `lof` of the `profile`, then the `friend` is deleted from the `lof`. Otherwise, the `profile` is returned unchanged.
- f) Write a function `friends?` that consumes a `profile` and a `profile` and produces a Boolean. The function returns `true` if the user of the first `profile` is a friend of the user of the second `profile` and vice versa, and `false` otherwise.

