Lab 05: Structures

Create a separate file for each question. Keep them in your “Labs” folder, with the name `liiqj` for Lab ii, Question j.

Download the headers for each function from the file `labinterface05.rkt` linked off the “Labs” page on the course Web site.

After you have completed a question (except class exercises), including creating tests for it, you can obtain feedback by submitting it and requesting a public test. Follow the instructions given in the Style Guide.

This lab makes use of the following structure and data definitions:

(*define-struct game* (*winner loser high low*))
;; A game is a structure (make-game w l hi lo), where
;; * w is a string,
;; * l is a string,
;; * hi is a nat, and
;; * lo is a nat such that hi is greater than lo.

(*define-struct timer* (*hours mins secs*))
;; An timer is a structure (make-timer h m s), where
;; * h is a natural number,
;; * m is an integer in the range from 0 to 59, and
;; * s is an integer in the range from 0 to 59.

(*define-struct clock* (*hours mins*))
;; A clock is a structure (make-clock h m), where
;; * h is an integer in the range from 0 to 23 and
;; * m is an integer in the range from 0 to 59.

(*define-struct card* (*value suit*))
;; A card is a structure (make-card v s), where
;; * v is an integer in the range from 1 to 10 and
;; * s is a symbols from the set ’hearts, ’diamonds,
;; ’spades, and ’clubs.

Language level: Beginning Student.

1. *[Class exercise with lab instructor assistance]* Create a function *fixed-game* that consumes a game *g* and produces the game formed by giving all of the loser’s points to the winner.

2. Create a function *convert-time* that consumes a *timer* and produces the equivalent time in seconds.

3. Create a function *bigger-card* that consumes two *card*s and produces the *card* with the higher value (or the second *card* if they have the same value). Hint: the suit of the *card* has no impact on its value.

4. Create a function *big-card-small-suit* that consumes two *card*s and produces a *card* as follows: the *card* produced will have the value of the *card* with higher value and the suit of the *card* with lower value. If the two *card*s consumed have the same value, then the *card* produced should have the suit of the first *card*.

5. Create a function *dur* that consumes two *clock* structures and produces an integer indicating the number of minutes elapsed between two times. If the second time is later than the first, you can assume that both times are on the same day. If the second time is earlier than the first, you can assume that the first time is on one day and the second time is on the next day. One way to approach this problem is to write a helper function that determines if two times are on the same day.

6. *Optional open-ended question* Choose a simple game or puzzle and devise a structure to represent it. Write one or more functions that consume a structure and produce the structure representing how it would change after a single move.