

CS115 – Lab 10: Other Data Structures

Spring 2020

1 Structures and Data Definitions

Structures and Data Definitions

! Do not copy these down! Download the interface file for this lab.

```
(define-struct name (first last))  
;; a Name is a (make-name Str Str)  
  
(define-struct info (name session clicker-grade))  
;; a Info is a (make-info Name Nat Num)  
;; requires:  
;; session is one of 1 and 2  
;; clicker-grade is a number between 0 - 100  
  
;; A Clicker-id is a Str of length 8  
;; A Clicker is a (list Clicker-id Info)  
;; A Clicker-Dict is a (listof Clicker)
```

Structures and Data Definitions

! Do not copy these down! Download the interface file for this lab.

```
(define name1 (make-name "Amy" "Clark"))  
(define name2 (make-name "Ziyu" "Jiang"))  
(define name3 (make-name "Mark" "Clark"))  
  
(define info1 (make-info name1 1 93))  
(define info2 (make-info name2 1 0))  
(define info3 (make-info name3 2 79.5))  
  
(define clicker1 (list "A24N5T01" info1))  
(define clicker2 (list "A24N5T02" info2))  
(define clicker3 (list "A24N5T03" info3))  
  
(define cdic-sample (list clicker1 clicker2 clicker3))
```

Question 1: Dictionary Lookup

Exercise

Write a function (`look-up-info clicker-dic id`) that consumes a `Clicker-Dict` and a `Clicker-id`, and returns the information about the student in `clicker-dic` whose ID is `id`.

You may require that there is such a student.

For example: (`look-up-info cdic-sample "A24N5T01"`) => (`make-info (make-name "Amy" "Clark") 1 93`)

```
(define-struct name (first last))
;; a Name is a (make-name Str Str)

(define-struct info (name session clicker-grade))
;; a Info is a (make-info Name Nat Num)
;; A Clicker-id is a Str of length 8
;; A Clicker is a (list Clicker-id Info)
;; A Clicker-Dict is a (listof Clicker)
```

Question 2: Data Analysis

Exercise Write a function (`clicker-avg clicker-dic`) that consumes a `Clicker-Dict` and returns the average of the grades of the students in it. For example: `(clicker-avg cdic-sample) => (/ (+ 93 0 79.5) 3) => 57.5`

```
(define-struct name (first last))
;; a Name is a (make-name Str Str)

(define-struct info (name session clicker-grade))
;; a Info is a (make-info Name Nat Num)
;; A Clicker-id is a Str of length 8
;; A Clicker is a (list Clicker-id Info)
;; A Clicker-Dict is a (listof Clicker)
```

Question 3: Find by Name

Exercise Write a function (`find-id clicker-dic last-name`) that returns a `(listof Clicker-id)` containing the ID of all the students in `clicker-dic` whose last name is `last-name`. For example:
`(find-id cdic-sample "Clark") => (list "A24N5T01" "A24N5T03")`

Question 4: Who's here?

Exercise Write a function (`session-name-list clicker-dic session`) that consumes a `Clicker-Dict` and a `Nat` and returns a `(listof Name)` containing the names of all the students in `clicker-dic` with `session`. For example: `(session-name-list cdic-sample 2) => (list (make-name "Mark" "Clark"))`