Lab 03: The design recipe and helper functions

The purpose of this lab is for you to get practice in using the design recipe, including testing. Make sure to use *check-expect* or *check-within*, as appropriate.

Download the headers for each function from the file labinterface03.rkt and create a separate file for each question saved as liiqj.rkt for Lab ii, Question j.

After you have completed a question (except class exercises), you can obtain feedback by submitting it on Markus.

Language level: Beginning Student.

1. *[Class exercise with lab instructor assistance]* Write a function *initials*, that consumes three non-empty strings (first, middle, last), and produces a string containing the first letter of each of the three strings.

2. Create a function *ring-volume* that consumes three positive numbers representing the inner radius, outer radius, and thickness of a ring, and produces its volume.
   Example:
   (ring-volume 3 4 2) => 43.9822...

3. Create a function *last-char* that consumes a nonempty string and produces a string consisting of the last character in the original string. Do not use *string-ref*.
   Example:
   (last-char "abc") => "c"

4. Create a function *shipping-bill* that determines the cost for shipping merchandise. It consumes the handling charge (a fixed cost for a shipment of any size), the charge per kilogram, the weight of a box in kilograms, and the number of identical boxes.
   Example:
   (shipping-bill 100 2 5 6) => 160

5. The airport parking lot has rates by the week and by the day, where you pay by the weekly rate of $74.95 for each complete week (any consecutive seven days) and by the daily rate of $14.95 for any remaining days. Create a function *airport-parking* that consumes an integer number of days and produces the bill. Be sure to use constants where appropriate. Note: don't worry if you produce a number that looks like 5 instead of 5.00 or a number that looks like 5.1 instead of 5.10.
   Example:
   (airport-parking 15) => 164.85

6. The cost of editing a film is calculated by the length of the film in minutes and the number of clips that the film contains. Each editing company offers the same cost per minute ($150 per minute) and base cost per clip ($100 per clip). The companies also each offer a discount after a certain number of clips (the *breakpoint*) have been edited; the discount and breakpoint

differ from company to company. After *breakpoint* number of clips, each clip costs the discounted price. Create functions described in parts a, b, and c and submit them in a single file.

(a) Create a function *length-cost* which consumes a natural number of minutes, m, and produces the cost of editing a film with m minutes.

Example:

(length-cost 20) => 3000

(b) Create a function *clips-cost* which consumes:

- A natural number of clips, c
- A natural number, bp, representing the breakpoint,
- And a number between 0 and 1 inclusive, disc, representing the discount price after bp number of clips

The function produces the cost of editing a film at that company.

Examples:

(clips-cost 5 3 0.25) => 450

because we have 3 clips at full price and 2 clips at 25% off

(clips-cost 5 10 0.5) => 500

because we have 5 clips at full price and 0 clips at 50% off

(c) Create a function *film-choice* which consumes:

- A natural number of minutes, m, representing the length of the film
- A natural number of clips, c, representing the number of clips in a film
- Natural numbers, bp1 and bp2 representing breakpoints at companies 1 and 2 respectively
- Numbers between 0 and 1 inclusive, disc1 and disc2, representing the discounts at companies 1 and 2 respectively.

The function produces a string in the following format:

"Cost for option 1 is X and cost for option 2 is Y"

Where X is the total cost of editing a film with company 1 and Y is the total cost of editing a film with company 2.

Example:

(film-choice 30 10 5 8 0.25 0.5) =>

"Cost for option 1 is 5375 and cost for option 2 is 5400"

because for option 1 we have 5 clips at full price ($500) and 5 clips at 25% off ($375), and for option 2 we have 8 clips at full price ($800) and 2 clips at 50% off ($100), and both options have 30 minutes at $150 per minute ($4500).

You may find it convenient to use the function *number->string*.

7. *Optional open-ended questions*

   (a) Create a function that consumes a string (an adjective) and produces a comparative by adding "er" (or adding "more" to the front) or into a superlative by adding "est" (or adding "most" to the front).

   (b) Pig Latin is a (not very difficult to decode) scrambling of English words created by taking the initial consonant sound off the front of the word and moving it to the end and then appending the string "ay". For example, "apple", "banana", and "grape" are translated to "appleay" (no consonant sound at the front), "ananabay" (single letter consonant sound), and "apegray" (multiple-letter consonant sound). Create a function that converts a string into a simple version of Pig Latin by just adding "ay" to the end of the string. Now create a function that moves the first letter in the string (whether or not it is a consonant) to the end and then appends "ay". We will refine this function as we learn more.

   (c) If time permits, consider creating a function that consumes a string and a position number, and returns the string formed by removing the letter in that position. Or a function that consumes a string and two position numbers, and returns the string formed by removing all the letters in between the two positions.

Helpful tips

Commenting and uncommenting Select a block of text (part of a line, one line, or more) in the Definitions window. Under "Racket" on the menu bar, select "Comment Out with Semi-colons". You can undo the change using "Uncomment". *Do not use comment boxes, or your assignments will be unmarkable.*

Indenting Select a block of text in the Definitions window. Under "Racket" on the menu bar, select "Reindent". You can also use "Reindent All" to reindent the entire program. To indent a single line, put the cursor on the line and press tab.

Jumping to a definition In the top left corner of your window is an arrow labelled "(define ...)". When you click here, you get a menu of all your definitions. You can choose whether they should be sorted by their order in the file or alphabetically.