Assignment 2
Programming functions using the design recipe

Programming Fundamentals 1

Submit by: Thursday, 8 Oct 2020 at 23:00

1 The language

Develop this assignment in DrRacket, using the **Beginning Student Language** (BSL). The official documentation for BSL is available here (as well as through DrRacket's *Help* menu).

2 The assignment

You must follow the design recipe to solve the following exercises. That is, you should include for every function definition:

- data type definitions, written as comments, for all the function's input or output types that are not built-in types (the built-in types are `Number`, `String`, `Boolean`, and `Image`)
- a signature, purpose statement, and header
- a template, commented out
- several input/output examples, in the form of executable tests

Your functions need not be "robust". That is, you can assume that the actual arguments passed to your functions agree with the function signature.

2.1 Functions and data types

1. Write a data type definition for a length in centimeters.

2. Write a data type definition for a length in inches.

3. Design a function `inch->cm` that converts a length in inches to centimeters. (One inch is 2.54 centimeters: define a constant for the conversion factor.)

4. Design a function `mean` that computes the average of four numbers.

5. Design a function `area-triangle` that computes the area of a triangle given its base and height.

6. A *perfect square* is a an integer that is also the square of another integer. For example, 4 is a perfect square, whereas 8 is not. Design a function `perfect-square?` that returns `#true` if its input is a perfect square, and `#false` otherwise. (*Hint*: the square root of a perfect square is an integer.)

7. Write a data type definition for a price in CHF.

8. A store has toilet paper on sale: if you buy up to 6 rolls, you pay the regular price of 2 francs per roll; if you buy up to 20 rolls, you get a 10% discount on the regular price; if you buy more than 20 rolls, you get a 15% discount on the regular price. Design a function `final-price` that computes how much you pay for any given number of toilet paper rolls.

9. Design a function that inputs the name of a note in the C-major scale given using neo-Latin naming and outputs the same note's name using English naming. The neo-Latin names (used, for example, in Italy and France) are *do*, *re*, *mi*, *fa*, *sol*, *la*, and *si*; the corresponding English names are *C*, *D*, *E*, *F*, *G*, *A*, and *B*.

The following two exercises are based on similar ones by Nate Nystrom.

10. Design a function `quarter->angle` that takes a time in quarter hours (an *integer* in the interval between 0 included and 4 excluded), and returns the angle (in degrees) a clock's minute hand should make. Angles are computed counter-clockwise from the 0 (horizontal to the right) to 360. At 0 quarter hours, the minute hand should point straight up (90 degrees); at 1 quarter hours, it should point to the right (0 degrees); at 2 quarter hours, it should point down (270 degrees); and at 3 quarter hours, it should point to the left (180 degrees).

11. Design a function `clock-minutes` that takes a time in quarter hours (now any nonnegative integer, not necessarily less than 4) and returns an image of a clock with the minute hand placed at the appropriate position. The function's implementation should use function `quarter->angle` you defined above. *Hints*:

    - The clock face can be just a circle.

- The minute hand can be just a narrow rectangle.
- To handle arbitrary numbers of quarter hours, use the `modulo` function to wrap the value into the interval from 0 to 4. For instance `(modulo 10 4)` evaluates to 2, since $2 = 10 - (2 \times 4)$.
- Remember to include the library for drawing images with `(require 2htdp/image)`.
- Use function `rotate` from `2htdp/image` to rotate the minute hand at the correct angle.

3 How and what to turn in

Using *iCorsi*'s website for Programming Fundamentals 1, upload under **Assignment 2** a single Racket source file named `YourLastName_YourFirstName_PF1_Assignment2.rkt` including *all function definitions* described in this assignment, each with the *design recipe*'s artifacts.

The submission **deadline** is hard, and late submissions will not be accepted. If you have a justified reason that prevents you from submitting this assignment on time, ask the instructors for an extension **well before** the deadline.

4 Points

This assignment will be graded on a *four-step scale*:

not submitted: you didn't submit anything (or barely anything)

fail: the submission shows some work has been done, but it is clearly insufficient

pass: the submission is incomplete or has flaws, but it is overall acceptable

good: the submission is generally complete and satisfactory

According to your grade, you will receive the following points for this assignment (out of the 100 points awarded in this course):

- *not submitted*: 0 points
- *fail*: 1 point
- *pass*: 3 points
- *good*: 4 points

5 Plagiarism policy

Assignments must be done alone. Students are allowed to generally discuss assignments and solutions among them, but each student must work on and write down their assignment independent of other students. In particular, sharing solutions of assignments is not allowed and constitutes cheating.

Remember that cheating and plagiarism are unacceptable. The penalty for cheating or copying – including allowing others to copy your work – is up to 100% of your grade for the course.