CS 115 – Spring 2017
Assignment 03
Due: Wednesday, May 31 at 10:00 a.m.

- Unless otherwise indicated in the question, you may use only the built-in functions and special forms introduced in the lecture slides from CS115 up to and including the modules covered by this assignment. A list of functions described in each module of the lecture slides can be found at https://www.student.cs.uwaterloo.ca/~cs115/built_in
- For this and all subsequent assignments, you are expected to use the design recipe when writing functions from scratch, including helper functions.
- Download the interface file from the course Web page to ensure that all function names are spelled correctly, and each function has the correct number and order of parameters.
- Read each question carefully for restrictions.
- Test data for all questions will always meet the stated assumptions for consumed values.
- Do not copy the purpose directly from the assignment description. The purpose should be written in your own words and include references to the parameter names of your functions.
- The solutions you submit must be entirely your own work. Do not look up either full or partial solutions on the Internet or in printed sources.
- Do not send any code files by email to your instructors or any other course staff. Course staff will not accept it as an assignment submission. Course staff will not debug code emailed to them.
- You may post general assignment questions using the discussion groups on Waterloo LEARN. Choose Connect → Discussions. Read the guidelines for posting questions. Do NOT post any code as part of your questions.
- Check Markus and your basic test results to ensure that your files were properly submitted. In most cases, solutions that do not pass the basic tests will not receive any correctness marks.
- Any string or symbol constant values must **exactly** match the descriptions in the questions. Any discrepancies in your solutions may lead to a severe loss of correctness marks. Basic tests results will catch many, but not necessarily all of these types of errors.
- Read the course Web page for more information on assignment policies and how to organize and submit your work. Follow the instructions in the Style Guide. Your solutions should be placed in files a03qY.rkt, where Y is a value from 1 to 3.

Language level: Beginning Student
Coverage: Module 3

1. Write a Racket function called `valid-name?` that consumes four nonempty and case-sensitive strings fname, lname, s1, and s2, and a natural number n, which is greater than 0. The function produces `true` if fname starts with s1, lname ends with s2, and the length of fname and lname when concatenated is greater than n. Note that s1 is not longer than fname and s2 is not longer than lname.

   Examples:
   - `(valid-name? "Emily" "Clarke" "Em" "arke" 10) => true`
   - `(valid-name? "Emily" "Clarke" "mi" "ark" 8) => false`

2. Write a Racket function `data-type` that consumes an input, `value` (of one of the following types: number, string, or symbol), and produces one of the following strings:
   - `"String. Length >= 10"`,
   - `"String. Length < 10"`,
   - `"Symbol"`,
   - `"Integer. Positive value"`,
   - `"Integer. Negative value"`,
   - `"Zero"`,
   - `"Number. Non-integer value"`.

   Some examples:
   - `(data-type "To be or not to be") => "String. Length >= 10"`
   - `(data-type -1) => "Integer. Negative value"`
   - `(data-type pi) => "Number. Non-integer value"`
   - `(data-type 0) => "Zero"`
   - `(data-type 8.0) => "Integer. Positive value"`

   Note:
   - **You may NOT use nested conditions** and y**ou may NOT use helper functions**,
   - You may use and & or,
   - The strings the function might produce must match those provided in the question exactly in order to get any marks - any typographical error in the string will be marked as incorrect.

3. A shoe store carries six shoe styles. During a shoe sales event, customers who bought three shoes within a certain period were given a discount of *2%* on the first shoe purchased, *5%* on the second shoe, and *10%* on the third shoe. An additional discount of *5%* of the original price is applied to extra small and extra large shoe sizes. Extra small and extra large shoes are represented with "X", while other shoe sizes are represented with "Y".

   Each shoe is represented by a single character, for example, the ballet shoe is "B", converse is "C", galesh is "G", jelly is "J", loafers is "L", and moccasin is "M". Prices are as follows: "B" is $10, "C" is $40, "G" is $70, "J" is $25, "L" is $90, and "M" is $120. You may assume all shoes are in stock.

   The three shoes bought by a customer is represented by a string, `shoes`, and shoe sizes are represented by another string, `sizes`. Length of `shoes` is three, and length of `sizes` is three. The first character represents the first shoe bought, the second character represents the second shoe, and the third character represents the third. Characters in `sizes` correspond to the characters in `shoes`.

   Write a Racket function `shoe-cost` that consumes two strings, `shoes` and `sizes`, and produces the total amount paid by the customer.

For example:

- `(shoe-cost "CGJ" "XYY") => 126.2`
  - Discount on shoes "CGJ" (40*2%)+(70*5%)+(25*10%) = 6.8
  - Discount on sizes "XYY" (40*5%)+(70*0%)+(25*0%) = 2
  - (40 + 70 + 25) – (6.8 + 2) = 126.2
- `(shoe-cost "LBM" "YXY") => 205.2`
  - Discount on shoes "LBM" (90*2%)+(10*5%)+(120*10%) = 14.3
  - Discount on sizes "YXY" (90*0%)+(10*5%)+(120*0%) = 0.5
  - (90 + 10 + 120) – (14.3 + 0.5) = 205.2

4. For this question, you will perform step-by-step evaluations of Racket programs by applying substitution rules until you either arrive at a final value or you cannot continue. You will use an online evaluation tool that we have created for this purpose. To begin, visit this webpage: https://www.student.cs.uwaterloo.ca/~cs115/stepping/

Note:

- You will need to authenticate yourself using your Quest/WatIAm ID and password.
- Once you are logged in, **try the "Warmup questions" and complete the problems under Assignment 3.**
- You can re-enter a step as many times as necessary until you get it right, so keep trying until you completely finish every question.
- All you have to do is complete the questions online, we will be recording your answers as you go, and there is no file to submit.
- Note however that you are not done with a question until you see the message **Question complete!**
- You should see this once you have arrived at a final value and clicked on "simplest form" (or "Error", Depending on the question).
- You should not use DrRacket's Stepper to help you with this question for several reasons. First, as mentioned in class, DrRacket's evaluation rules are slightly different from the ones presented in class, but we need you to use the evaluation rules presented in class. Second, in an exam situation, you will not have DrRacket's Stepper to help you, and there will definitely be step-by-step evaluation questions on at least one of the exams.
- The basic tests for this assignment will tell you whether or not we have a record of your completion of the stepper problems.
- *There will be no automatic basic test mail sent to UWaterloo email upon finishing this question. However, you may request for a basic test or you could submit another file to receive the basic test for this question.*