Post-Mortem

Assignment 03

October, 2017

We normally publish the post-mortem for an assignment after it has been marked and released. Here is a list of common errors provided by the graders for assignment 3.

Style and Spacing

• Parameter, function, and constant names should generally be lower case.

General

• Many students used type names as their parameters. For example, “nutri-fact” and “Nutri-Fact” should not be used as parameter names since they are already used for the structure definition and data definition respectively.

Design Recipe

• Purpose statements should meaningfully use each parameter name. For example, a purpose that starts with “(foo x y) consumes a number x and a string y and produces...” is just repeating information that is present in the contract.

• Types in contracts should always be capitalized. This includes user-defined structures like a Nutri-Fact or Hand.

• Any requirements that are present in a data definition do not have to be repeated in a contract for a function that consumes that type.

• The test cases used for your check-expects should be valid tests. This means that any structures used in the tests should follow the data definitions for those types.

• Although we do not require a tests section for helper functions, these functions should still be adequately tested by the functions that use the helper. After running the file, no black highlighting should be present in any functions.

Question 2

• Some students forgot to require that the resize function can only consume positive serving sizes (since otherwise the function would not produce a valid Nutri-Fact).

• Many students did not define constants for the calories/nutrient rates.

• Many students did not consider the possibility that valid-nutri-fact? can consume a make-nutri-fact with improperly typed fields (e.g. (make-nutri-fact 'name false empty “this” “is” “invalid”)).
Question 4

• Some students did not define enough helper functions to simplify their code. Many solutions would benefit from defining helper functions to check if a given hand has a certain attribute.

Ongoing Errors

The following is a list of common errors from previous assignments that were still repeated for assignment 3.

• Constants (and helper functions) should be defined above the design recipe for the function they are used in. Some students still defined their constants and helper functions between their examples and function definition.

• Helper functions require their own purpose, contract, and examples.

• Lines should be less than 80 characters long: excessively long lines should be broken up into multiple shorter lines.

• Some students forgot to leave a blank line before and after function definitions or two blank lines before and after function blocks (design recipes).

• equal? should only be used to compare two values of unknown type. When the types of the arguments are known, use the most appropriate comparison function (e.g., symbol=?).

• To determine if a boolean value is true or false, use that value directly. Expressions like (boolean=? param true) are overly complex.

• Starting a new cond expression in an else clause is unneeded. Instead, directly check for the next condition in the original cond expression.

• Purposes, contracts, and function definitions should not be explicitly labelled.