

Post-Mortem

Assignment 03

February 11, 2020

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

- Constants (and helper functions) should be defined above the design recipe for the function they are used in. Many students defined their constants and helper functions between their examples and function definition, or after the main functions specified in the assignment. Students are advised to refer to pages 18-19 of the style guide for a sample submission that includes constant and helper function definitions.
- Lines should be less than 80 characters long: excessively long lines should be broken up into multiple shorter lines, and make use of DrRacket's auto-indenting features.
- If the `cond` question/answer pair does not fit in one line and causes the code to look too horizontal, students are advised to put the answer component on a new line.
- Many students forgot to leave a blank line before and after function definitions, or included both the function header and part of the function body on the same line.

General

- Remember to check basic test results after submitting to catch simple (but impactful) errors.
- Some students code did not run at all, hence they lost all correctness marks. You are advised to make sure your code runs.
- Some students used disallowed functions, which resulted in them losing a great amount of correctness marks. Please go over the assignment description very carefully before attempting a solution and look over it before submitting to ensure they adhere to the given restrictions.
- Some students still had black highlighting present in their code, which is a clear indication of a lack of thorough testing. Students are advised to - as an absolute minimum - ensure there is no black highlighting in their code once it has been run.
- The names of helper functions should also be clear and descriptive. For example, `helper`, `test`, `function`, and `compute` are very poor choices for helper function names.
- When working with predicate functions, take advantage of the result of boolean expressions. As a rudimentary example, `(cond [(or (= x 3) (= x 4)) true] [else false])` can be simplified to `(or (= x 3) (= x 4))`.

Design Recipe

- For this assignment, many students did not include some or all of the contract requirements for all the main functions that had some, which resulted in a loss of contract correctness marks. Please note that contract requirements are not a separate portion of the design recipe.
- Purposes, contracts, and function definitions should not be explicitly labelled.
- Purposes should begin with a function header (e.g., `(func-name param1 param2 param3)`). These headers should include each of the parameter names used in the function. **This is not the same as referencing parameter names in the purpose's description.**
- **Purposes should meaningfully use each parameter name in the description of the function, and these references to the parameter names should be written exactly as they appear in the function header.**
- Contracts should begin with `func-name:` (including the colon).
- If restrictions are already implied in a data type, they do not need to be included in the `requires:` section. For example, if a function consumes a `Nat`, there is no need to specify that the consumed `Nat` should be greater than or equal to 0.
- Some students forgot to capitalize the type names in their contract (e.g. using `nat` instead of `Nat`).

Question 1

- This question was very well done!

Question 2

- Many students forget to state in the requirement that RGB lists have to contain exactly 3 elements and each element should be between 0 and 255 (inclusive)
- In Part b, the output list is `(listof Int)` instead of `(listof Nat)`, since the output might be `(list -1 -1 -1)`
- Many students did not define constants for the eight colours in Part a and b. You should define those lists as constants to improve readability and to avoid repetition.

Question 3

- This question was very well done!