

Post-Mortem

Assignment 01

January 28, 2019

We normally publish the post-mortem for an assignment after it has been marked and released. **Note that design recipe components were not marked for assignment 1, as they were not required for this assignment.** Here is a list of common errors provided by the graders for assignment 1.

Style and Spacing

- Lines should be less than 80 characters long: excessively long lines should be broken up into multiple shorter lines.
- Some students did not use DrRacket's auto-indenting features and had inconsistent indentation. You may use Ctrl-I or cmd-I for auto-indentation.
- Many students did not leave a blank line before and after function definitions or two blank lines before and after function blocks.

General

- Some students named their functions differently from what was specified in the assignment. These students have lost all their correctness marks for these functions. The basic tests results did notify them that the required function has not been defined, so it is extremely important that you submit early and often and check your basic tests results to ensure that this does not happen again in the future.
- Some Racket functions, such as '+' and '*', can consume more than 2 arguments.
- Many students used parameter names that were not meaningful (e.g., `a`, `m`, `f`) or used ambiguous/unclear names (e.g., `fe`, `mt`).
- Many students used excessively long parameter names (e.g. `the-overall-assignments-grade` instead of something that is shorter and still meaningful, like `assignments-grade`). The style guide clearly indicates to not use filler words such as 'a', 'an' and 'the'.
- Many students did not follow the naming conventions discussed in the style guide for constant and parameter names. In particular, names should contain a dash between each word, they should not use any capital letters (with the exception of special constants or proper nouns), and they should not use any underscores.
- Some students named their constants ambiguously (e.g. `(define assign 0.2)` instead of a clear name like `(define assns-weight 0.2)`) which is highly discouraged.
- Some students submitted code that did not run at all. Students are advised to make sure their code runs on their local machine.

Question 2

- In part (a), some students swapped the order of multiplication/division in their translation (e.g. $(\ast 4/3 (\text{expt } r 3) \text{ pi})$ instead of $(\ast 4/3 \text{ pi } (\text{expt } r 3))$, which is not a literal translation.
- In part (b), some students failed to define a constant for G: (define G 6.674e-11).
- In part (b), some students wrote a decimal equivalent in their definition of the constant G, which was not a direct translation.
- In part (c), some students did not define constants for phi and psi as directed to.
- In part (d), some students did not give a direct translation of the fraction, and simply divided $e^{\pi\sqrt{\frac{2n}{3}}}$ by the denominator of the given fraction.

Question 3

- Some students used different values than what was given in the question, which led to severe loss of correctness marks.
- Many students did not define constants for the conversion ratios.
- Some students defined overly complex constants that did not reflect the derivation of the value.
- In part (b), many students did not multiply or divide by the correct conversion factors, and thus did not pass most of the correctness tests.

Question 4

- Many students did not define constants for the weights associated with each grade component.
- In part (b), a few students forgot to account for the final exam weight.