Assignment 10

Interpreter in Racket

In this assignment we will build an interpreter in Racket. You will find detailed instructions in the file: https://github.com/skiadas/ProgLangAssignments/blob/master/assignment10sub.rkt

There is no "parser" for this interpreter. You would write your sample programs by directly writing a combination of surface and core language expressions. Also you will notice that the surface language expressions are not a whole separate layer, and there's no desugar method. The surface language constructs are regular Racket functions or macros, that take an expression and turn it into another expression.

You are not allowed to use any of the mutation features in Racket. But you may otherwise use Racket features we may not have talked about. There are really few instances where this should be needed, though. For the most part we have covered all that you will need for this assignment.

Correctness of the solutions, including whether the code loads, will count for 15 points. Your code MUST load with no errors and have the correct types. You should make sure that the compile steps described in the README in the assignment file work out without problems.

Style-wise, pay particular attention to the names you use for your variables, and to the appropriate vertical-aligning of related items.

Another 5 points will come from style issues, your tests and use of GitHub, including Milestones and Labels, issues and closing those issues via commits. Keep creating issues for the various parts of the assignment. assigining them to milestones and tagging them with labels. **Create new milestones for each assignment**.

1. You already have created a link to the instructor's remote repository. You need to download the updated version of the instructor's repository and merge the changes into yours. The following should do that:
   - Type: `git fetch instr`
   - Type: `git merge master instr/master`. This may give you a editor window to edit a commit message. You don't need to edit it, just "save" (writeOut) and exit/close the document. (This is probably not your prefered editor. See this page[1] for how to set up your favorite editor for use in these merge situations. I would recommend setting it to SublimeText).
   - If it reports no problems, you're ready.
   - If it reports merge conflicts, you will need to resolve those first, then make a commit. this page[2] has some instructions on how to do that. Or contact me.
2. Create new milestones for components of this assignment as described in the assignment's documentation file.

[1] https://help.github.com/articles/associating-text-editors-with-git/
[2] https://help.github.com/articles/resolving-a-merge-conflict-from-the-command-line/

3. Throughout your work, create and close issues via commits as described in the instructions above and in previous homeworks. Make sure to assign each issue to the correct milestone and to give them the correct labels.
4. There are two files that you need to work with: assignment10sub.rkt contains the source code, while assignment10tests.rkt contains the tests.