

211hw1_S11

Homework 1

Due: 10:00am, Fri Jan 21, 2011

100 points

For this assignment, set the DrScheme Language to `Beginning Student with List Abbreviations` (under `How to Design Programs`). Your assignment will be graded using the specified language level. If you use a different language level, your code may not work when it is graded.

- **Carefully follow the [Sample Solution to a Programming Problem](#)** in the [Scheme HW Guide](#). Less than half of the credit for each programming problem is based on the correctness of your code as determined by our test cases. Most of the grade is based on how well you follow the *design recipe*.
- Do the following problems from the book (HTDP):
 - 8.3.1 (15 pts)
 - 8.3.2 (15 pts)
 - 9.3.3 (14 pts)
 - 9.5.2 (14 pts)
 - 9.5.7 (14 pts) [Note: toy prices are numbers. See 9.5.6.]
 - 10.1.5 (14 pts)
 - 10.1.7 (14 pts)
- Notes:
 - Be sure to do all the assigned parts of each problem. In the book, each problem ends with a block icon. Online, each problem ends with a hand icon.
 - In problem 9.5.7, the book asks you to write a function that checks for the empty list input as an error and throws an aborting error in the case. (The purpose statement should document this behavior!) In the newest version of DrScheme, the `error` function take a single argument *not two arguments as documented in the book*. We recommend using a string (text enclosed in double quotation marks like `"This is sample string"`). You can test the error throwing behavior of a function using `check-error` which is documented in the DrScheme Help Desk.
 - Read chapter 8 very carefully before doing the problems from that chapter. This chapter is *not* padded with lots of wordy examples. In writing hand evaluations for problems 8.3.1 and 8.3.2, follow the same format as the evaluation examples given in the [Scheme HW Guide](#).
 - To follow the design recipe, you must write down the *purpose*, *contract*, at least 5 well-chosen examples, write the code for the function, and include illustrative test cases for the function (using at least the examples you developed ahead of time). You can bundle the examples and test cases together as a block of `check-expect` invocations. Your chosen examples should illustrate the output you expect, and the test cases should produce the actual output (they should not be commented). If the function processes an inductive type, make sure that your examples include the base case(s) and sample inductive cases.

Submit your homework using the [Comp 211 Owlspace page](#).