# PLC: Workout 3 [90 points]

Due date: Wednesday, February 12th by midnight

## About This Homework

This assignment is about point-free style and also about representing computer languages as datatypes.

### How to Turn In Your Solution

Please submit your solution via ICON. The required file for this assignment is:

- `Exercises.hs`

**Please use exactly the file names we are requesting.** We will require you to resubmit your homework with a 5-point penalty if the names are not exactly as we are requesting. This is for purposes of grading scripts. It is ok if ICON adds a number to your file name on multiple submission (which is allowed up to the deadline).

### Partners Allowed

You may work alone or with one partner. You should both turn in your solution to the assignment, which we expect will be the same (but is allowed to be different, if you worked together but then you decided to add to your solution – or whatever the scenario). Also, you need to turn in a file called `partner.txt` which lists your partner's Hawkid. This will let us know that you worked with that person (lest we incorrectly think you plagiarized another student's similar submission).

### How To Get Help

You can post questions in the `workouts` section on Piazza.

You are also welcome to come to our office hours. See the course's Google Calendar, linked from the github page for the class, for the locations and times for office hours.

## 1 Reading

Read Chapters 8 and 10 of *Programming in Haskell*.

## 2 Basic exercises [45 points]

Complete the exercises `p1` through `p5` in `Exercises.hs`. I am not providing tests this time, so you will want to test your solutions yourselves.

You may use any coding style you wish for full credit in this section, but in Section 4, you are required to write three out of the five of these problems in point-free style. (So it is likely to your advantage to try using that style from the beginning.)

## 3 Translating from expression type [24 points]

Translate the expressions in `PF.hs` to Haskell functions of the same type. These expressions are of type `E a`, for various types `a`. The `E` type constructor is defined in `E.hs`. An evaluator for `E` is also included, which you can use for testing your solutions.

Add your solutions as `e1` through `e3` to `Exercises.hs`. So `e1` in `Exercises.hs` should have type `Int -> Int`, for example.

## 4 Translating to expression type [21 points]

Pick three of `p1` through `p5` and write expressions of type `E a` for them, for the appropriate type `a`. Name your translation of `p2` (for example) `r2`. So the type of `r2` should be `E (Int -> [Int] -> Int)`, for example. Add these to `Exercises.hs`.