

PLC: Workout 1 [90 points]

Due date: Wednesday, January 29th by midnight

About This Homework

You will get started with Haskell, and solve some basic programming problems in Haskell.

How to Turn In Your Solution

Please submit your solution via ICON. The required files for this assignment are:

- `screenshot.png`
- `Basic.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.

No Partners for This Assignment

For other assignments in the class, you will be allowed to work with a partner, but for this one, you must turn in a solution by yourself. This is to make sure that you can get started using Haskell.

How To Get Help

You can post questions in the `hw1` 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 1 and 2 of the required book, *Programming in Haskell*, by Graham Hutton.

2 Getting started with Haskell [20 points]

For the first part of this course, you will be programming in Haskell. I will be using emacs in class, which is already installed on the MacLean Windows machines, but you can use any text editor, like Notepad++.

Installing Haskell and Emacs

While you are welcome to use Haskell and Emacs from the CS lab Windows machines, I do encourage you to install these on your own computer if possible. Haskell and Emacs are freely available for Windows, Linux, and Mac. For Windows, we have created a single installer for everything you need, here:

<http://homepage.divms.uiowa.edu/~astump/agda/Agda2.6.0.1.v1.msi>

If you use the installer, you should probably restart your computer after you apply it, or Windows may have trouble finding Haskell.

If you are installing on Mac or Linux (or for some reason do not want to use our installer), then I recommend you install the Haskell Platform (from www.haskell.org). This works on all three major platforms. You would then separately need to install Emacs. On Mac, try <https://emacsformacosx.com> for Emacs. Note that we have not had good luck with running Emacs within the terminal on Mac. You should download a proper standalone application for Emacs on Mac.

If you have problems getting Haskell and Emacs working on your computer, please come see us in office hours, and we will help you (but try it seriously first yourself).

2.1 Testing Haskell [20 points]

To make sure that Haskell is working correctly, you first have to open up a terminal (Mac, Linx) or command shell (Windows). On Windows, you can start a command shell by searching for `cmd` and then selecting “Command Prompt”. Now navigate to your copy of the course repository (you can use `dir` and `cd` on Windows, or `ls` and `cd` on Mac/Linux), and into the `hw0` directory. Now type `ghci` and hit return. This should start the Haskell interpreter (if it does not, it likely means your path is not set correctly to include the directory where Haskell is installed on your computer). Then type `:l Basic`, to ask the Haskell interpreter to load `Basic.hs`. You should see something like this:

```
antioch:~/plc-priv/hw1$ ghci
GHCi, version 8.6.5: http://www.haskell.org/ghc/  :? for help
Prelude> :l Basic
[1 of 1] Compiling Basic          ( Basic.hs, interpreted )
Ok, one module loaded.
*Basic>
```

Now type `hello` and hit enter. You should see something like this:

```
*Basic> hello
"Welcome to Haskell!"
```

Please take a screenshot of this interaction called `screenshot.png`. This is just to demonstrate that you were able to start up `ghci` (whether on your computer or a lab machine).

2.2 Haskell exercises [70 points]

Complete the exercises in `Basic.hs`. There are tests in `PublicTests.hs` you can use to help make sure your code is correct. I suggest: uncomment the commented-out problems (p1 through p20) in `Basic.hs` and the corresponding test in `PublicTests.hs` one at a time, to try solving the first and testing, then solving the second and testing, etc.