PLC: Homework 0 [50 points]

Due date: Wednesday, January 23rd by midnight
4 extra-credit points if you turn in Tuesday, January 22nd by midnight

About This Homework

For this homework, you will

1. clone the class repo from github.uiowa.edu
2. set up a private github.uiowa.edu repo for your work in the class
3. get started with Haskell, by either installing and configuring it on your computer (optional) or configuring it on the CS lab Windows machines, and running a simple command

So mostly this homework requires you to download some things and get some software working. This is not very academic, but every computer scientist should know how to do this. And it will take the pressure off later homeworks if we just devote our first homework to doing this.

How to Turn In Your Solution

For this class, all your work will be turned in through your github.uiowa.edu repository. Creating this repository is part of the assignment (see below). So you have to follow the steps in Section 3 first, before you can submit.

When you have your personal repository set up, create a directory within it called `hw0`. Stage, commit, and push with git to turn in your files. You can go to the web site for your github.uiowa.edu repo to make sure you have gotten your files uploaded. The required files for this assignment are:

- `class-repo.YYY`
- `Hw.hs`
- `haskell-screenshot.YYY`

Please use exactly the file names we are requesting. We will take off 5 points if your files do not have the exact names we are requesting, including capitalization, or are not in a `hw0` directory (with that exact specific name). In the past, we have had problems with students naming their files slightly differently than we have requested in the instructions. For example, someone might have a `Homework 0` directory instead of `hw0`. Please do not do this. Just call the folder `hw0`, exactly as we are asking. Because we will rely on grading scripts for grading later assignments, we need you to get the names exactly correct. In addition to helping us grade, this doubles as a small exercise in attention to detail, which is an important work/life skill.

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 with Subversion and Haskell.

How To Get Help

You can post questions in the `hw0` 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 Cloning the class's github.uiowa.edu repo [15 points]

The github.uiowa.edu repo for the class is here:

`https://github.uiowa.edu/astump/3820-spring19`

The first step is to clone this repo to your computer (or you can use the MacLean lab's Windows computers). If are cloning to your own computer, please see the instructions here:

`https://github.uiowa.edu/astump/3820-spring19/blob/master/uiowa-github-tips.md`

On the MacLean lab machines, GitKraken is not yet installed, so for now you can use the git client that is already there (under "Git" in the list of Windows programs). Try using "Git bash", and then proceed as follows, based on the instructions linked above:

- Run `ssh-keygen` to create a public/private key pair. Add the public key to your github.uiowa.edu user settings as described in our github tips.
- Now you can run `git clone git@github.uiowa.edu:astump/3820-spring19.git` to clone the repo.

Take a screenshot in showing the contents of the `3820-spring19` directory you get from cloning the repo, and call this `class-repo.YYY` (where `YYY` is whatever graphics format you can conveniently get a screenshot in, like jpeg or png).

3 Setting up your github.uiowa.edu repo [15 points]

All homework, including this one, will be submitted through github.uiowa.edu. For this, you need to create your own repo on that server, following these steps (which assume you have already added a public/private key pair as you had to do to clone the class repo in the previous section):

1. Create a new repo called `3820-spring19`. It is critical that you name the repo exactly this, so course staff can clone it.
   (a) Your repo should be private, so other students cannot see your work
   (b) You should add the following users as "collaborators" under the settings for your repo: `ocook`, `rdong6`, and `astump` (these are the Hawkids for the course staff). This is critical so we can actually clone your otherwise private repo.
2. Then clone your repo to your computer or the MacLean lab Windows computer following similar steps to those above for cloning the class repo.
3. Create a subdirectory of your `3820-spring19` directory (cloning creates this), called `hw0`
4. Copy the file `Hw.hs` from the `hw0` subdirectory of the class repo to the `hw0` subdirectory of your repo
5. Edit the file to change the value of `hawkid` to your actual hawkid (as a string literal)
6. Also, copy the `class-repo.YYY` file you created in the previous problem, to your `hw0` subdirectory

4 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.cs.uiowa.edu/~astump/agda/AgdaBundle_2.5.4.2.v4.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). It can be a little frustrating, but that is why we have a (half) homework just for this.

4.1 Testing Haskell [15 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 Hw`, to ask the Haskell interpreter to load `Hw.hs`. You should see something like this:

```
antioch:~/plc/hw0$ ghci
GHCi, version 8.4.3: http://www.haskell.org/ghc/  :? for help
Prelude> :l Hw
[1 of 1] Compiling Hw               ( Hw.hs, interpreted )
Ok, one module loaded.
```

Now type `hawkid` and hit enter. You should see something like this (except with your Hawkid):

```
*Hw> hawkid
"astump"
```

Please take a screenshot of this called `haskell-screenshot.YYY` (where again, `YYY` is something like `jpeg` or `png`), and put that in your `hw0` subdirectory of your personal repository.

5 Completing your submission

Stage, commit, and push the changes to your repo back to github.uiowa.edu.