PLC: Homework 0 [50 points]

Due date: Wednesday, August 30th, 9pm
2 extra-credit points if you turn in by Tuesday, August 29th, 9pm

About This Homework

For this homework, you will

1. get started using Subversion, including downloading several Subversion repositories to whichever computer you will be using for the work of this class (your own, or a lab computer)
2. get started with Haskell under Emacs, by either installing and configuring it on your computer (optional) or configuring it on the CS lab Windows machines, and running some simple commands

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 via personal subversion repositories we have created for each person in the class. Checking out this repository, and several others, is part of the assignment (see below). So you have to follow the steps in Section 1 first, before you can submit.

When you have your personal repository set up, create a directory within it called `hw0`. Make sure you do a "Subversion Add" on this directory itself, to add it to Subversion (see the instructions in the "using-subversion.txt" file linked in Section 1). Then turn in your solution by copying the required files to this directory, adding them using Subversion Add (again, see the subversion instructions linked from Section 1 below), and then doing Subversion Commit on your personal repository directory. The required files for this assignment are:

- `repos-screenshot.YYY`,
- `haskell-screenshot.YYY`,
- `hash.txt`, and
- `haskell-mode-screenshot.YYY`

You will not literally have `YYY`, but rather the appropriate filename suffix for whatever type of image file you are giving us (e.g., `jpg`, or `pnm`).

Checking that you have submitted correctly. Simply go to the URL for your personal repository using a web browser. The browser will ask for your login credentials, which will just be your HawkId and HawkId password. Then you will see what is currently checked in to the repository in your directory. If you do not see your `hw0` directory and the required files, then your submission has not been checked in (feel free to ask us for help in this case).

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 Resources tab of the Resources page on Piazza, for the locations and times for office hours.

1 Downloading Subversion Repositories [20 points]

All homework, including this one, will be submitted using the Subversion version-control system. The following document describes how to install and use subversion (username and password are both guest if prompted):

`https://svn.divms.uiowa.edu/repos/clc/class/3820-fall17/using-subversion.txt`

For this problem, follow the directions in that document to download the following repositories. Note that if you are on a CS lab machine, you will definitely want to store these somewhere on your `H:` drive (like `Documents`, which is stored there), or they will not roam with you to different lab computers.

- The course repository
- Your personal repository

Note that while you can always view a repository in a web browser, to submit to your personal repostitory, you must use a Subversion client.

To prove that you have done this, please save a screenshot image (any standard image format is fine) in a file called `repos-screenshot.YYY`, where `YYY` is whatever extension the screenshot tool on your computer uses when it creates the screen shot (e.g., `repos-screenshot.jpeg`). Your screenshot should just show a directory containing these repositories (e.g., on Windows, under

Windows Explorer), but do try to include some of your desktop background or something else to show that the image comes from your computer.

2 Reading

Read Chapters 1 and 2 of the required book, *Programming in Haskell*, by Graham Hutton.

3 Getting Started with Haskell [30 points]

For the first part of this course, you will be programming in Haskell from within the Emacs text editor. Haskell and Emacs are already installed on the CS lab Windows computers in Room 301 MacLean Hall. Those computers can also be accessed remotely, as explained here:

`https://www.divms.uiowa.edu/clas_linux/services/view.html`

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_2.5.2.msi`

If you use the installer, you should probably restart your computer after you apply it, or Windows may have trouble finding Haskell. After a restart, the test in the next section should work. The installer also includes Agda, which we will be using later in the semester. For now, though, you will just need Haskell and Emacs.

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 whole homework just for this.

3.1 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 hw0`, to ask the Haskell interpreter to load `hw0.hs`. You should see something like this:

```
Prelude> :l hw0
[1 of 1] Compiling Main             ( hw0.hs, interpreted )
Ok, modules loaded: Main.
```

Now type `hashHawkid MY-HAWKID` and hit enter, where `MY-HAWKID` should be replaced with your actual Hawkid, in double quotes. You should see something like this (except with your Hawkid):

```
*Main> hashHawkid "astump"
4599252333964053774
```

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 (don't forget to add it with Subversion add). Also, please put that number into a text file called `hash.txt` in that same `hw0` subdirectory of your personal repository.

Installing `haskell-mode`

We will use `haskell-mode` within Emacs to provide nice features like syntax highlighting for Haskell source code. To install `haskell-mode`, first start Emacs. If you are using a CS Windows machine, then open the file `h:/.emacs`. Otherwise, open `~/.emacs`. If you are on the CS lab machines, create a new directory at the top level of your `h:` drive, called `emacs-packages` (exactly like that), and add this text to your `.emacs` file (this will store `haskell-mode` to your `h:` drive instead of the local disk of the computer):

```
(custom-set-variables
 '(package-archives
   (quote
    (("gnu" . "http://elpa.gnu.org/packages/")
     ("melpa-stable" . "http://stable.melpa.org/packages/"))))
 '(package-user-dir "h:/emacs-packages"))

(eval-after-load "haskell-mode"
   '(define-key haskell-mode-map (kbd "C-c C-y") 'haskell-compile))
```

If you are on your own computer, you instead just need to add this text:

```
(custom-set-variables
 '(package-archives
   (quote
    (("gnu" . "http://elpa.gnu.org/packages/")
     ("melpa-stable" . "http://stable.melpa.org/packages/")))))
```

```
(eval-after-load "haskell-mode"
   '(define-key haskell-mode-map (kbd "C-c C-y") 'haskell-compile))
```

Then restart emacs, and type `Alt-x` and then `package-refresh-contents` and hit enter. Next type `Alt-x` and `package-install`, hit enter, then type `haskell-mode` and hit enter again. This should install `haskell-mode`.

3.2 Testing `haskell-mode` [15 points]

Restart Emacs, and open the file `hw0.hs` in the course repository's `hw0` directory (as above). You should see "`(Haskell Interactive)`" in the Emacs mode line. Now do "Control-c Control-l". This should open a new Emacs window with text that looks like this:

```
Hours of hacking await!
If I break, you can:
  1. Restart:           M-x haskell-process-restart
  2. Configure logging: C-h v haskell-process-log (useful for debugging)
  3. General config:    M-x customize-mode
  4. Hide these tips:   C-h v haskell-process-show-debug-tips
>
```

Now type `hashHawkid MY-HAWKID` as above and hit enter. You should see the same thing you saw above. Finally, with your cursor back in the `hw0.hs` Emacs window, type "Control-c Control-y". This tells `haskell-mode` to compile `hw0.hs`. You should see something like this:

```
-*- mode: haskell-compilation; default-directory: "~/plc-priv/hw0/" -*-
HsCompilation started at Tue Aug 15 14:47:31

cd /home/astump/plc-priv/hw0/ && cabal build --ghc-option=-ferror-spans
Building hw0-0.1.0.0...
Preprocessing executable 'hw0' for hw0-0.1.0.0...
[1 of 1] Compiling Main             ( hw0.hs, dist/build/hw0/hw0-tmp/Main.o )
Linking dist/build/hw0/hw0 ...

HsCompilation finished at Tue Aug 15 14:47:32
```

Take a screenshot `haskell-mode-screenshot.YYY` showing your version of this output, and save it (again doing subversion add) to the `hw0` subdirectory of your personal repository.

Don't forget to do a "subversion commit" on that `hw0` subdirectory of your personal repository, to push your new files over to the subversion server (where we can then download them for grading).