CSE 341 - Programming Languages - Autumn 2012

Running Haskell

We're using the Glasgow Haskell Compiler, platform version 2012.2.0.0 (which includes version 7.4.1 of GHC). GHC is the *de facto* standard version of Haskell.

On the CSE Undergrad Windows Machines

Open a command window and navigate to the directory where you want to keep your Haskell source files. Run Haskell by typing `ghci` or `ghci MyFile.hs`.

Alternately, if you're just experimenting and don't have a source file, you can run Haskell via the "Start" menu. It's here:
`All Programs / DEV TOOLS & LANGUAGES / Haskell Plaform 2012.2.0.0 / GHCi`

(The "i" in "GHCi" stands for "interactive", as opposed to compiling and producing an executable file.)

You can also run Haskell on a .hs file by double-clicking on the file name (but this may not work correctly if you are importing another module that you defined, since Haskell will look in the current working directory for this file).

There are also local copies of documentation under
`All Programs / DEV TOOLS & LANGUAGES / Haskell Platform 2012.2.0.0 / GHC Documentation`

On CSE Undergrad Linux Machines (attu)

At the command prompt type `ghci` to start it, or `ghci MyFile.hs` to start it and load `MyFile.hs`.

An advantage of running on the lab Linux machines is that we have the emacs Haskell mode package already downloaded -- see the "Using Linux and Emacs in CSE 341" section of Software for CSE 341.

Note that attu has an older version of Haskell (7.0.4). This seems unlikely to affect anything you're going to do in 341 -- but if you get a complaint about a missing package or the like you could try your program on the current version of Haskell on another machine. The earlier problem with HUnit is fixed.

On a Personal Machine

Haskell is available for Windows, Mac, and Linux. Here's the download page:
http://hackage.haskell.org/platform/.

Basic Haskell Commands

A few essential commands:

- `:?` -- list commands
- `:quit` (or `:q`)
- `:load` [filename] (or `:l [filename]`)
- `:reload` -- reload the current file set after you've edited it (or `:r`)
- `:cd` -- change to another directory
- `:type` [expr] -- give the type of an expression (or -- just `:t`)
- `:set +t` tell Haskell to print type information after evaluation

If you just want to try Haskell without loading any files, just start it up. For anything interesting, though,

you'll want to have a file with defintions in it. A simple way to do this is to use your favorite editor to create a file (with the extension `.hs`) in your home directory or a subdirectory. Save the file but leave the editor open on it. Then start up Haskell in that same directory on that file. Try things out. If you want to edit your file, do so and save it, and then back in Haskell reload the file using `:reload`.

The above directions may be all you want for 341; and in any case I'd suggest starting out that way. For students comfortable with emacs (or who are willing to learn) running Haskell from emacs is more convenient in the long run. There are directions for installing and running this at Haskell mode for Emacs.

If you are running on the ugrad linux machines, we've got the emacs Haskell mode packages downloaded already. Just add the following lines to the `.emacs` file in your home directory:

```
(load "/cse/courses/cse341/common/haskell-mode/haskell-site-file")
(add-hook 'haskell-mode-hook 'turn-on-haskell-doc-mode)
(add-hook 'haskell-mode-hook 'turn-on-haskell-indent)
```

Beware the Evil Tab!

Unlike most programming languages, whitespace is significant in Haskell. So that you can see whether things are lining up properly use a fixed-pitch font when editing. Haskell has some rule about how tabs are processed, but I recommend that you avoid tabs in Haskell code and always use spaces instead.

For emacs, if you use the Haskell mode for Emacs, it will avoid using tabs in files. Otherwise, by default, Emacs inserts tabs in place of multiple spaces when it formats a region. Putting the following in your `.emacs` file turns this off:

```
(setq-default indent-tabs-mode nil)
```