Assignment 0: Warmup
Due: 11:59pm, Tuesday, January 25

1 Part 1: Haskell

This course will require a substantial amount of Haskell coding, starting quite soon. You therefore need to get re-acquainted (or aquainted, if you're taking CS 131 this semester) with Haskell.

A good starting point is to review the tutorial "Learn You a Haskell for Great Good", found at http://learnyouahaskell.com, up through and including Chapter 8. There are other Haskell links on the course webpage as well.

2 Part 2: Subversion and Historical Context

All homework assignments will involve use of the subversion source-control system. Most of you are already familiar with subversion. A brief summary of subversion appears as an appendix, but if you will certainly benefit by referring to other resources (e.g., the book available at http://svnbook.red-bean.com).

1. Change to the non-publicly-readable directory where you would like to have all your homework and lab files. Execute the command:

   `svn checkout https://svn.cs.hmc.edu/cs132s11`

   to create a directory cs132s11; this is your working copy of the class repository. Your password will be your HMC CS password (the one used for logging into knuth.cs.hmc.edu, wilkes.cs.hmc.edu, and the Macs in the CS labs).

2. Read the four PDF documents in the doc subdirectory from earliest to latest.

3. In the other subdirectory (which is private to you), modify a0/summaries.txt to contain:

   • Your name
   • For each paper, an exactly 42-word response. This can be a summary of the most important or most interesting point(s) and/or a question that arises in reading the paper.

4. Submit your changes to the summaries.txt file by running `svn commit`.

If there are any problems with the assignment or the subversion repository, please send e-mail to cs132help@cs.hmc.edu.
A Subversion Brief Summary

- There is a single central repository, containing every version of everyone's work for this semester of CS 132. (Access controls will ensure that you see only your own work, of course.)

- The central repository is located somewhere away from prying eyes. You can get a working copy (a local copy of your part of the repository) by running the command

  `svn checkout https://svn.cs.hmc.edu/cs132s11`

  When you run this command in some appropriate (non—publicly-readable!) location you will get a working copy in a subdirectory named cs132s11. (You could even create several working copies by running this command repeatedly in different places. This might be useful if you are working on different computers with different home directories, but most students will run the checkout command just once all semester.)

- Make whatever changes you want to the files in your working copy. The command

  `svn commit`

  creates a new version of the code and stores it permanently in the central repository. You will get the chance to add a short message describing your changes. You can also specify the message on the command line, e.g.,

  `svn commit -m "Finished part 2, and fixed the division bug in part 1"`

  Commit early and often: you can always get any past or current version of your files from the central repository. Also, other people (e.g., grutors, or pair-programming partners) cannot see your changes until they are committed to the repository.

- If other people commit changes to the central repository (i.e., the professor adding new files, or a partner committing new work, or you committing changes made in a different working copy), you can update your working copy to incorporate these changes by running

  `svn update`

  The system may insist that you run update before it will let you commit.
• If two people make changes to the same file in a way that Subversion believes to be in conflict (e.g., making different changes to the same line of a file), you may end up with a conflict. Subversion will not let you commit until the conflict is resolved; see the Subversion documentation.

• If rather than changing existing files you want to add brand new files or directories, you must tell Subversion about them, with the command

```
svn add <filename(s)-to-add>
```

The files/directories must exist before they can be added, and you still need to `svn commit` before the new files/directories appear in the central repository.

• The commands `svn commit`, `svn update`, and `svn add` apply to the current working directory and any subdirectories. Thus, you'll usually want to run these commands in the main cs132s11 directory, to make sure you're covering all files and directories.

• Other extremely useful commands are `svn diff`, `svn status`, `svn log`, `svn revert`, `svn cat`, and `svn blame`. See Subversion documentation for more information.