CMSC 22300 **Functional Programming** **Handout 1**
Spring 2012 **March 29, 2012**

Phoenixforge and Subversion

Phoenixforge

We are using the CS department **phoenixforge** server to manage homework submissions. The server hostname is `https://phoenixforge.cs.uchicago.edu`.

Each of you has a personal phoenixforge project called *user*`-223`, where *user* is your CNetID. Associated with this project is s Subversion (svn) repository where your homework assignments will be submitted. You can access your phoenixforge project by loading the above URL in your browser and logging in with your CNetID and password, but you do not need to do that in order to use your svn repository.

Using Subversion

Your phoenixforge project has an associated Subversion repository on the server where you will keep the source code of your homework assignments. To *checkout* a copy of a assignment called "`homework1`," run the following command:

```
svn checkout https://phoenixforge.cs.uchicago.edu/svn/user-223/homework1
```

where *user* is replaced by your CNetID. On your first checkout, you should be prompted for your password. It will assume you are using the username of the account executing svn. If your CNetID is different from your login name, just press enter at the password prompt and it will then ask you for your phoenixforge username (*i.e.* your CNetID) and then your corresponding CNetID password. If everything checks out, a *working* directory called `homework1` will be created in the current directory. All the files related to your assignment should live in this directory.

Now suppose you create a new file called `foo.sml` in your `homework1` directory. In order for Subversion to keep track of it, it needs to be added to the repository. You do this using the following command:

```
svn add foo.sml
```

You should see a message like:

```
A         foo.sml
```

This response indicates that `foo.sml` has been scheduled to be added to your repository, but the file will actually be added only when you commit your changes. To do so, type the following command:

```
svn commit -m "added foo.sml"
```

to add the file permanently to the repository. If you omit the `-m` option, you will be prompted to enter a log message in your default editor. After you have entered the commit command, you will see a message like the following:

```
Adding         main.sml
Transmitting file data .
Committed revision 1.
```

Changes you make to your files are recorded in the repository every time you do a `svn commit`. Before you make changes to your files, you can ensure that you have a current version, by running `svn update`. You will want to run update once your assignment has been graded to get a copy of the graded (and commented) version of your code that will have been added to your repository by the grader.

Not all the files in your assignment directory need to be in the repository. For example, you should not put your executable files in the repository — these can always be recreated (hopefully!) by compiling the source. In particular, you do not need to add the files in the `.cm` subdirectory that are created by the SML/NJ compiler when you run `CM.make`.

The `svn diff` command is used to compare the version of a file (or files) in your working directory with the version(s) in the repository. If no files (or options) are specified, all working files are compared to their last committed versions, otherwise only the specified files are compared. There are also flags to compare other versions, see the man pages or the online manual for details.

Other useful svn commands are:

svn status This lists files in the current directory and subdirectories that have been modified (M) but not committed, or have not been added to the repository (?).

svn delete *filenames* This deletes the designated files.

svn move *path1 path2* This moves the file designated by *path1* to the location/name specified by *path2*.

The Subversion home page is at `http://subversion.apache.org/`. Full official documentation is available at `http://svnbook.red-bean.com/`.