Lab 0 Pt 2: SVN

Due: Tuesday, July 3rd, 11:59pm

***NOTE**: This lab is based on a lab originally written by John Reppy for CMSC 15100 (and updated and edited by a variety of instructors and TAs throughout the years)*

1.
    At this point you should have your GitHub account and repository set up.
    Change to your desktop directory.
    Check out a local copy of your repository with
    svn checkout **https://github.com/yourRepoAddress**
    At this point, if you don't have a subversion for some reason, please let the TA know.
    Type ls to make sure the local copy did indeed arrive.
    Change into the local copy of your repository.
    Type touch "This is a test." > test.file.
    Type svn add test.file.
    Type svn commit test.file -m "Adding a test file.".
    Type svn mkdir lab0_v2. This will create a directory lab0_v2 inside your repository where you can store your work for this part.

2.
    Make and push a test.txt file into your subversion repository, you need to add it first, then commit it. Note that once you've added a file, you do not need to add it again, although you will (usually) commit it again many times. Also notice how you don’t need to push or pull every time unlike Git.

    To do this, change into the root directory of your repository at the command line (you can use pwd to verify that you are in the correct directory). Then type these instructions:

    ```
    $ svn add lab1/test.txt
    $ svn commit lab1 -m "committing test file"
    ```

3.
    Getting out of trouble in the shell is a critical skill. Therefore, we ask you to type some commands that will not succeed, to gain some experience with (minor) shell mishaps.

    Try to determine the result of each of the following commands before typing them:

    ```
    $ cd ~
    $ ls foo
    $ mkdir foo
    $ ls foo
    $ touch foo/FILE1.empty
    $ ls foo
    ```

Explain (to yourself) the different results you observed typing the same command ls foo three times.
Now type this:

```
$ rm FILE1.empty
$ cd foo
$ rm FILE1.empty
$ ls foo
$ cd ..
$ ls foo
```

Why does one rm command fail and the other succeed?
Finally, try this sequence of instructions:

```
$ touch foo/FILE1.empty
$ ls foo
$ rm FILE1.empty
$ rm foo/FILE1.empty
$ rmdir foo
$ cd ..
$ rmdir foo
```

4.

Look online for more SVN commands and why some people chose this over Git.
If you have completed this part of the lab, you're free to use either SVN or Git for your repository, both are equally efficient and useful in their own way.