CS431 — Exercise 1

April 4, 2017

Due: Monday, April 10, 2017 before midnight (40 points)

The main goal for this exercise is verifying that you can use `git` successfully and have access to the server for code submission.

Installing and configuring git

1. - If you are using a distribution of Linux, git should be available in your package manager's repositories. For example, on Debian-based distributions you can use `apt-get install git` as `root`.
   - If you are using Windows, you can get a Windows installer from the git website at `http://git-scm.com`. When installing, the defaults should be fine. After the installation, you should have access to a program called git bash that will give you a Unix-like terminal. For the rest of the section, use git bash to run the commands.
   - If you are using a Mac, there should also be an installer available at `http://git-scm.com`.
2. From a terminal or git bash, run the following commands to configure your user:

   ```
   git config --global user.name "Your Name"
   git config --global user.email "youremail@cpp.edu"
   ```
3. Optionally, you can change some of the other settings. For example, you can change the default editor with:

   ```
   git config --global core.editor nano
   ```

   If you'd like to use a different difftool such as Meld, install it then set the following:

   ```
   git config --global diff.tool meld
   git config --global merge.tool meld
   ```

Project Setup and Writing Some Code

1. If you have not done so, register your account at `https://codebank.xyz` using your `@cpp.edu` email address and your bronconame as your username.
2. Log in to `https://codebank.xyz` to verify your account is working properly.
3. From the website, choose to create a new project and name it `CS431-EX1`.

   You must name all projects exactly as shown including case to ensure they are graded correctly.
4. On your local machine, create a folder where ever you plan to store your CS431 materials for this exercise.

5. Open a terminal/git bash and navigate to that folder then run

   ```
   $ git init
   ```

   This initializes an empty git repository in that folder.

6. Now, we need to attach this local repository to the one on the `codebank.xyz` server. To do this, run (replacing name with your bronconame)

   ```
   $ git remote add origin https://codebank.xyz/name/CS431-EX1.git
   ```

7. Next, create a Java file named `ProcessTest.java` that contains a class named `ProcessTest`.

8. Using Java's Process API have your program call the program `java` with no command line arguments.

9. Compile and test your program. You probably won't see any output.

10. Run

    ```
    $ git status
    ```

    and notice it states there are one or two files in red (untracked).

11. Run

    ```
    $ git add -A
    $ git commit -m 'first commit'
    $ git push origin master
    ```

    If you refresh the web page from earlier after creating the project you should now see a new commit has been pushed and you can verify the files contain the expected code.

12. Now, review the Process API to see if there is a way you can add some code that will print out the result of calling the `java` program as a subprocess within your program.

13. After modifying your code, use the following commands to push another commit with the changes:

    ```
    $ git add -A
    $ git commit -m 'code modified to show output'
    $ git push origin master
    ```