CSCI git project for CSCI375

It is crucial that you read and understand these instructions...
think about what you are doing at each step...
pay attention to what you are typing in....
every word of every instruction you type...

Blindly cutting and pasting chunks of instruction into your command window will result in disasters, and you will get no sympathy from the instructor.

.... SLOW DOWN
... READ TO THE END BEFORE YOU BEGIN
... PAY ATTENTION
... AND THINK!

Project documents, code, and other files will be collected through the use of the git version control tool.

The basic idea is as follows, detailed instructions are provided afterwards:

- A central git repository is maintained by the department, with a section within it for csci375.
- The project will have several repositories containing the relevant material the instructor is distributing to the students.
- Each student will also have a central repositories where they maintain their current official submissions for the project.
- At the start of term the students will initialize their own repositories as a copy of the instructors'.
- The student will then create a working copy of each repository within their regular csci account (e.g. on otter), and use that copy to make all their desired changes as they build their solution and create documentation.
  The student will use appropriate git commands (commit, branch, merge, etc) to maintain their local working copy, and as often as desired (prior to the deadline of course) the student can push their most recent version of their work back to the central repository.

  I strongly recommend students become comfortable with the basic use of each of the following git commands:
  add, commit, branch, checkout, merge, tag, revert, rm, show, push, and fetch as well as basic familiarity with *gitk*

- After the deadline passes, the instructor will retrieve the latest version of the students' work from the central repository and use that as the basis for marking the phase.

Specific instructions

Note that the examples below are based on csci375, you will need to make appropriate substitutions in the commands if you are using these instructions for a different course or for another lab/assignment/project phase.

For your individual repositories (for submitting Timesheets, Contributions files, and Evaluations)

1. At the start of the first lab, on otter (or one of the pups or cubs) create a directory for your csci375 material:

   ```
   mkdir ~/csci375
   ```

2. You can see your current central repository information using the command

   ```
   ssh csci info
   ```

   For each repository you have access to, this shows the name of the repository, and one or more of RWC (for Read access - you can fork/clone it, Write access - you can push to it, Creator - you created it)

3. At the start of term create your own fork of it on the central repositories. For the individual portion of the project there are three repositories: **TimeSheets**, **Contributions**, and **Evaluations**. You will create your own fork of each using the commands:

   ```
   ssh csci fork csci375/TimeSheets csci375/$USER/TimeSheets
   ssh csci fork csci375/Contributions csci375/$USER/Contributions
   ssh csci fork csci375/Evaluations csci375/$USER/Evaluations
   ```

   Note that this creates a repository of your own on the central server containing a copy of the instructor's repository, including all the branches.

4. Once you have your central versions set up, it is time to create working copies in your own account. You will need to do *this for each of the above repositories*. This is done as follows for the Timesheets repository:

   ```
   cd ~/csci375
   git clone csci:csci375/$USER/TimeSheets
   cd TimeSheets
   ```

5. Submitting weekly individual submissions. For each repository, do the following:

   - Move to the appropriate repository (i.e., Timesheets)
   - Create the file you are going to submit, and add any content you wish
   - Add the file

   ```
   git add filename
   ```

   - Commit the file:

   ```
   git commit -m "---some message describing the change---"
   ```

   - Push the file:

   ```
   git push
   ```

Other useful commands:

```
git rm filename if you want to get rid of a file (-r for directory)
git mv oldname newname if you want to move or rename a file or
directory
git add -u (so any files you've removed locally also get removed from
the repository)
```

****ONLY DO THE FOLLOWING IF YOUR INSTRUCTOR HAS UPDATED THE REPOSITORY:****

- Add the instructor's repository as an upstream so you can fetch updates from the instructor if necessary. This example shows how to do it for the TimeSheets repository.

```
cd ~/csci375/TimeSheets
git remote add instructor csci:csci375/TimeSheets
```

Type in `ls` to verify that the expected files were copied across.

- ***(Only done if told to!!)*** *How to fetch a new version from the instructor if the instructor tells you an update has been posted and needs to be fetched:*

ONLY DO THESE STEPS IF I EXPLICITLY TELL YOU A NEW UPDATE NEEDS TO BE RETRIEVED - hopefully that will rarely be the case
If an update is issued you can retrieve it using
```
 git fetch instructor
```
You can then see what has changed using
```
git diff HEAD instructor master
```
You can then make any changes needed and merge the instructor's master into yours with
```
git merge instructor master
```