The following git(1) commands are used in CS 440. The git(1) SCM is already installed on the CS server.

Create a configuration file to identify the user (you):

1. change to your home folder - "cd"
2. create/edit a ".gitconfig" file - "vim .gitconfig"
3. add the following text:

```
[user]
    name  = Your Name
    email = email@una.edu
```

4. Save the file and quit vim

The .gitconfig file only needs to be created once per user per machine (laptop, desktop, etc.). The .gitconfig file *MUST* be located in your home folder.

Create a new repository:

1. cd into the directory where you wish to create the repo
2. "git init -b main"

Stage (add) a specific file for a future commit (track the file):

1. create the file to be added
2. "git add fname" where fname is the name of the file to stage

Stage (add) everything in the current folder:

1. create the files and folders to be added
2. "git add ."

Commit the staged file(s) to the repo:

1. "git commit"
2. Your default editor will open with information about this commit
   Lines that start with "#" are comments and will not be included
3. Add a short summary line describing the purpose of this commit (max 50 chars)
4. If more info is desired, skip a line (insert a blank line) and add more info
5. Save and quit to commit, quit without saving to abort the commit

Tag the most recent commit:

1. "git tag -a tagname" where tagname is the tag text (e.g. design)
2. Your default editor will open with information about this commit
3. Enter a brief commit message such as "Add design tag"
4. Save and quit to commit, quit without saving to abort the commit

Summary of commands:

Command	Description
git init -b main	create an empty git repository in the current folder
git add fname	tell git to track the current changes* for the file "fname"
git add .	tell git to track the current changes* for everything in this folder
git stage fname	synonym for add
git stage .	synonym for add, (git documentation often uses the word stage instead of add)
git commit	store the staged files in the git repository
git tag -a tagname	label the most recent commit with the friendly name "tagname"

Other useful git(1) commands for a standalone developer:

Command	Description
git status	show the status of the working tree
git diff	show the differences between the working tree and commits
git log	show the commit logs, newest entry to oldest
git restore	restore specified file(s) in the working tree

See the git(1) man pages for additional information about how to use git(1). For example, to read the man page for the "git commit" command:

type "man git-commit"
-OR-
type "git help commit".

Other git(1) man pages may be accessed using the same pattern.

* More specifically, this command will add file(s) to the index. The index is a snapshot of the working tree that will used for the next commit into the repository.