CSE410 aka CSE306
Software Quality

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall

http://www.cse.buffalo.edu/faculty/alphonce/SP17/CSE410
https://piazza.com/class/iybn33z3aro2p
### Plan
(drop LEX 9/10, EXP 3/4)

<table>
<thead>
<tr>
<th></th>
<th>Tues</th>
<th>Wed</th>
<th>Thur</th>
</tr>
</thead>
<tbody>
<tr>
<td>4/17-4/21</td>
<td>✔</td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>4/24-4/28</td>
<td>git wrap-up?</td>
<td>LEX08 (git)</td>
<td>TBA</td>
</tr>
<tr>
<td>5/1-5/5</td>
<td>debugging in dynamic languages</td>
<td>LPR - I</td>
<td>debugging in dynamic languages</td>
</tr>
<tr>
<td>5/8-5/12</td>
<td>TBA</td>
<td>LPR - II</td>
<td>TBA</td>
</tr>
</tbody>
</table>

- Note: Drop LEX 9/10, EXP 3/4.
EXERCISE

Go to repo from last class and do a 'git pull'. Draw a diagram showing the current state of the repo (but show just the current commit; don't worry about what we did last time).
Create a new branch and then add and commit two or more files to it.
Create another new branch and then add and commit two or more files to it. Update your diagram as you go along, showing the Working Directory, Staging area, and repo contents.
Results: groups 3, 4 & 5

These three groups did not switch back to the master branch between creating the first and second branch:

- `git branch A`
- `git checkout A`
- ...and and commit two files...
- `git branch B`
- `git checkout B`
- ...and and commit two files...
EXERCISE
(do on your own by Thursday)

Go to repo from last class and do a 'git pull'. Draw a diagram showing the current state of the repo (but show just the current commit; don't worry about what we did last time).
Create a new branch and then add and commit two or more files to it.
Create another new branch and then add and commit two or more files to it. Update your diagram as you go along, showing the Working Directory, Staging area, and repo contents.
Switch to one of the non-master branches. Edit and commit one of the files on that branch. Create and commit an additional file on that branch. Merge this branch into master.
EXERCISE

(do on your own by Thursday)

Switch to one of the non-master branches. Edit and commit one of the files on that branch. Create and commit an additional file on that branch. Merge this branch into master.

Let us first switch to branch ‘a’:

`git checkout a`
EXERCISE
(do on your own by Thursday)

Switch to one of the non-master branches. Edit and commit one of the files on that branch. Create and commit an additional file on that branch. Merge this branch into master.

Now edit and commit F2.

```
emacs F2 (call edited version F2')
git add F2'
git commit -m "...">
```
EXERCISE
(don on your own by Thursday)

Switch to one of the non-master branches. Edit and commit one of the files on that branch. Create and commit an additional file on that branch. Merge this branch into master.

Create F5, add, and commit.

```text
emacs F5
```
```
git add F5
```
```
git commit -m "...
```
EXERCISE
(do on your own by Thursday)

Switch to one of the non-master branches. Edit and commit one of the files on that branch. Create and commit an additional file on that branch. Merge this branch into master.

Switch to master.

`git checkout master`
EXERCISE
(do on your own by Thursday)

Switch to one of the non-master branches. Edit and commit one of the files on that branch. Create and commit an additional file on that branch. Merge this branch into master.

Merge a into master.

```
.git merge a
```
Because there was a straight line path from master to a, this is a "FAST FORWARD merge: master is simply moved to point to the same commit as a."
Non-FF merge
(three way merge)

- A new commit is created.
- The branch being merged into is the first parent of the new commit.
- 'master' should always be the first parent; hence switch to master before doing merge.
The non fast-forward merge creates a "merge commit". The branch merged into points to that new commit.
Deleting a branch

`git branch -d a`
Exercise

- Switch to branch b, edit F2 to create conflict (edit same part of F2 as you did on branch a, but change it in a different way), merge b into master (switch to master, then 'git merge b').
STARTING POINT
git checkout b
edit F2 to create conflict with F2'
git add F2"
git commit -m "..."
git checkout master
Blob created with conflict markings.

git merge b
edit F''' to resolve conflict, creating F2^4
git commit -m "..."
Question

- Is there a difference between
  - merging a into b
  - merging b into a