### Schedule

**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><strong>4/17-4/21</strong></td>
<td>git</td>
<td>(open)</td>
<td>git</td>
</tr>
<tr>
<td><strong>4/24-4/28</strong></td>
<td>TBA</td>
<td>LEX08</td>
<td>TBA</td>
</tr>
<tr>
<td><strong>5/1-5/5</strong></td>
<td>debugging in dynamic languages</td>
<td>LPR - I</td>
<td>debugging in dynamic languages</td>
</tr>
<tr>
<td><strong>5/8-5/12</strong></td>
<td>TBA</td>
<td>LPR - II</td>
<td>TBA</td>
</tr>
</tbody>
</table>
git

pointer to the current branch
$ more head
ref: refs/heads/master

-rw-r--r--  1 alphonce  staff   23 Apr 17 13:26 HEAD
drwxr-xr-x  2 alphonce  staff   68 Apr 17 13:26 branches
-rw-r--r--  1 alphonce  staff  328 Apr 17 13:26 config
-rw-r--r--  1 alphonce  staff   73 Apr 17 13:26 description
drwxr-xr-x 12 alphonce  staff  408 Apr 17 13:26 hooks
-rw-r--r--  1 alphonce  staff  137 Apr 17 13:26 index
drwxr-xr-x  3 alphonce  staff  102 Apr 17 13:26 info
drwxr-xr-x  4 alphonce  staff  136 Apr 17 13:26 logs
drwxr-xr-x  7 alphonce  staff  238 Apr 17 13:26 objects
-rw-r--r--  1 alphonce  staff  107 Apr 17 13:26 packed-refs
drwxr-xr-x  5 alphonce  staff  170 Apr 17 13:26 refs
$ git ls-files

README.md
<table>
<thead>
<tr>
<th>Mode</th>
<th>User</th>
<th>Group</th>
<th>Size</th>
<th>Date</th>
<th>Content</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>123</td>
<td>Apr 17</td>
<td>HEAD</td>
</tr>
<tr>
<td>d</td>
<td>alphonce</td>
<td>staff</td>
<td>68</td>
<td>Apr 17</td>
<td>branches</td>
</tr>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>73</td>
<td>Apr 17</td>
<td>description</td>
</tr>
<tr>
<td>d</td>
<td>alphonce</td>
<td>staff</td>
<td>408</td>
<td>Apr 17</td>
<td>hooks</td>
</tr>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>137</td>
<td>Apr 17</td>
<td>index</td>
</tr>
<tr>
<td>d</td>
<td>alphonce</td>
<td>staff</td>
<td>102</td>
<td>Apr 17</td>
<td>info</td>
</tr>
<tr>
<td>d</td>
<td>alphonce</td>
<td>staff</td>
<td>136</td>
<td>Apr 17</td>
<td>logs</td>
</tr>
<tr>
<td>d</td>
<td>alphonce</td>
<td>staff</td>
<td>238</td>
<td>Apr 17</td>
<td>objects</td>
</tr>
<tr>
<td>-r</td>
<td>alphonce</td>
<td>staff</td>
<td>107</td>
<td>Apr 17</td>
<td>packed-refs</td>
</tr>
<tr>
<td>d</td>
<td>alphonce</td>
<td>staff</td>
<td>170</td>
<td>Apr 17</td>
<td>refs</td>
</tr>
</tbody>
</table>
```bash
$ ls -l objects
total 0
  drwxr-xr-x  3 alphonce  staff 102 Apr 17 13:26 25
  drwxr-xr-x  3 alphonce  staff 102 Apr 17 13:26 39
  drwxr-xr-x  3 alphonce  staff 102 Apr 17 13:26 9c
  drwxr-xr-x  2 alphonce  staff  68 Apr 17 13:26 info
  drwxr-xr-x  2 alphonce  staff  68 Apr 17 13:26 pack
  drwxr-xr-x  3 alphonce  staff 102 Apr 17 13:26 info
  drwxr-xr-x  4 alphonce  staff 136 Apr 17 13:26 logs
  drwxr-xr-x  7 alphonce  staff 238 Apr 17 13:26 objects
  drwxr-xr-x  5 alphonce  staff 170 Apr 17 13:26 refs
```
$ git cat-file -t 25b4
ccommit
$ git cat-file -t 9ce9
tree
$ git cat-file -t 39af
blob
$ git cat-file -p 25b4

tree 9ce959348ab4c2e6b6154939b4b1acc0504a649
author Carl Alphonce <alphonce@buffalo.edu> 1492449992 +0000
committer Carl Alphonce <alphonce@buffalo.edu> 1492449992 +0000

README.md created online with Bitbucket
$ git cat-file -p 39af

# README#

This README would normally document whatever steps are necessary to get your application up and running.

### What is this repository for? ###

* Quick summary
* Version
* [Learn Markdown](https://bitbucket.org/tutorials/markdowndemo)

### How do I get set up? ###

* Summary of set up
* Configuration
* Dependencies
* Database configuration
* How to run tests
* Deployment instructions

### Contribution guidelines ###

* Writing tests
* Code review
* Other guidelines

### Who do I talk to? ###

* Repo owner or admin
* Other community or team contact
pointers to commits

-rw-r--r--  1 alphonce staff  23 Apr 17 13:26 HEAD
drwxr-xr-x  2 alphonce staff  68 Apr 17 13:26 branches
-rw-r--r--  1 alphonce staff  328 Apr 17 13:26 config
-rw-r--r--  1 alphonce staff  73 Apr 17 13:26 description
drwxr-xr-x 12 alphonce staff  408 Apr 17 13:26 hooks
-rw-r--r--  1 alphonce staff  137 Apr 17 13:26 index
drwxr-xr-x  3 alphonce staff 102 Apr 17 13:26 info
drwxr-xr-x  4 alphonce staff 136 Apr 17 13:26 logs
drwxr-xr-x  7 alphonce staff 238 Apr 17 13:26 objects
-rw-r--r--  1 alphonce staff 107 Apr 17 13:26 packed-refs
drwxr-xr-x  5 alphonce staff 170 Apr 17 13:26 refs
$ ls -l refs
total 0
  drwxr-xr-x  3 alphonce staff  102 Apr 17 13:26 heads
  drwxr-xr-x  3 alphonce staff  102 Apr 17 13:26 remotes
  drwxr-xr-x  2 alphonce staff   68 Apr 17 13:26 tags
  drwxr-xr-x  3 alphonce staff  102 Apr 17 13:26 info
  drwxr-xr-x  4 alphonce staff  136 Apr 17 13:26 logs
  drwxr-xr-x  7 alphonce staff  238 Apr 17 13:26 objects
  drwxr-xr-x  5 alphonce staff  170 Apr 17 13:26 refs
  -rw-r--r--  1 alphonce staff  107 Apr 17 13:26 index
  -rw-r--r--  1 alphonce staff  328 Apr 17 13:26 config
  -rw-r--r--  7 alphonce staff   73 Apr 17 13:26 description
  -rw-r--r--  1 alphonce staff  137 Apr 17 13:26 index
  drwxr-xr-x 12 alphonce staff  408 Apr 17 13:26 hooks
  drwxr-xr-x  3 alphonce staff  102 Apr 17 13:26 info
  drwxr-xr-x  4 alphonce staff  136 Apr 17 13:26 logs
  drwxr-xr-x  7 alphonce staff  238 Apr 17 13:26 objects
  drwxr-xr-x  5 alphonce staff  170 Apr 17 13:26 refs
  -rw-r--r--  1 alphonce staff  107 Apr 17 13:26 index
  -rw-r--r--  1 alphonce staff  328 Apr 17 13:26 config
  -rw-r--r--  7 alphonce staff   73 Apr 17 13:26 description
  -rw-r--r--  1 alphonce staff  137 Apr 17 13:26 index
Possible states of a file

- Modified
- Unmodified
- Staged
- Untracked

Actions:
- Edit
- Commit
- Add
commit preserves contents
(accidental removals can be recovered from)
understanding 'reset'

Create file
git add file
git commit -m "..."
Create file
git add file
git commit -m "..."
edit file
git add file
`git commit -m "..."`
Let's take this as our starting point
Moves the branch that HEAD refers to.
RESULT: master is moved; HEAD still refers to master.

git reset --soft HEAD~

HEAD~ is the parent of HEAD
STARTING POINT
git reset --mixed HEAD

-- mixed is default if not specified

Moves the branch that HEAD refers to, and unstages changes to the index.
RESULT: master is moved; HEAD still refers to master.
STARTING POINT
Moves the branch that HEAD refers to, unstages changes, and makes workspace reflect the staging area.

RESULT: master is moved; HEAD still refers to master.

```
git reset --hard HEAD~
```

---

-- hard can result in data loss if workspace had uncommitted files.
understanding branching


- `git branch <name>` creates a new pointer
Let's take this as our starting point.
git branch B
git checkout B

workspace

fileA

index staging

fileA

local repository

HEAD

B

master

commit

tree

fileA
create fileB

git add fileB
git commit -m "..."
git checkout master
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.
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.
Results: groups 1 & 2

These two groups switched back to the master branch between creating the first and second branch:

- git branch A
- git checkout A
- ...and and commit two files...
- git checkout master
- git branch B
- git checkout B
- ...and and commit two files...
Go to repo from git pull

- draw a diagram
- switch to one Edit & commit
- create another
- update your
- as you go
- create a branch
- draw a diagram
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...
For class

Updating current commit
what we did last week)
& commit
add two or more files to it
& commit
add, two or more files to it
showing WD, Staging, Repo>

Non-master branches
& commit, merge in master.
Group 5: "We thrive" "Five n' jive"
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.