CS 35L Software Construction Lab
Week 9 – Change Management
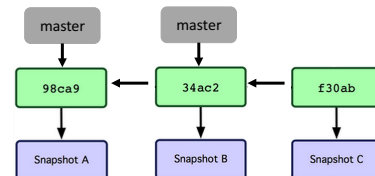
Final Information

- Monday, December 11, 2017
- 8 AM – 11 AM
- Open book and open note
- 50% of course grade
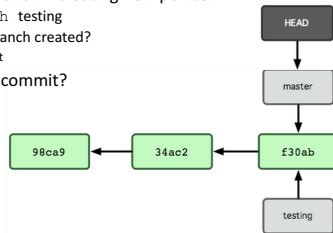- No electronic devices

Git Branch

- A pointer to one of the commits in the repo (head) + all ancestor commits
- When you first create a repo, are there any branches?
  - Default branch named 'master'
- The default master branch
  - points to last commit made
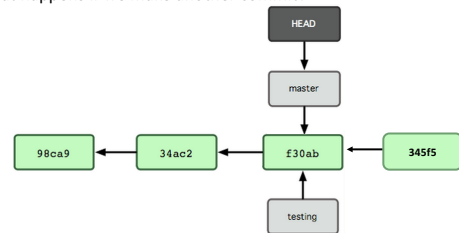 - moves forward automatically, every time you commit

Where Is Master?

New Branch

- Creating a new branch = creating new pointer
  - $ git branch testing
  - Where is new branch created?
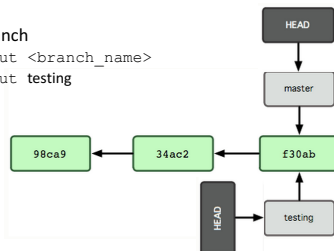   - Current commit
- Where is current commit?
  - HEAD

```
                      HEAD
                       ↓
                     master
                       ↓
98ca9 ← 34ac2 ← f30ab
                       ↑
                    testing
```
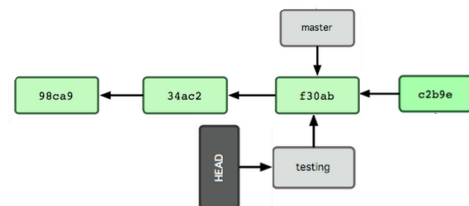
New Commit

- What happens if we make another commit?

```
                      HEAD
                       ↓
                     master
                       ↓
98ca9 ← 34ac2 ← f30ab ← 345f5
                       ↓
                    testing
```

Switching to New Branch

- Check out new branch
  - $ git checkout <branch_name>
  - $ git checkout testing

```
                      HEAD
                       ↓
                     master
                       ↓
98ca9 ← 34ac2 ← f30ab
              HEAD →  ↑
                    testing
```

Commit After Switch

```
                     master
                       ↓
98ca9 ← 34ac2 ← f30ab ← c2b9e
                       ↑
              HEAD → testing
```
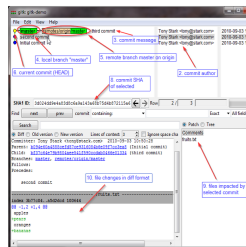
Why Branching?

- Experiment with code without affecting main branch
- Separate projects that once had a common code base
- 2 versions of the project

Homework 9

- Publish patch you made in lab 9
  - Create a new branch "quote" off of version 3.0
    - Branch command + checkout command (git branch quote v3.0; git checkout quote)
    - $ git checkout v3.0 -b quote
  - Use patch from lab 4 to modify this branch
    - Patch command
    - $ patch -pnum < quote-3.0-patch.txt
  - Modify ChangeLog file in diffutils directory
    - Add entry for your changes similar to entries in ChangeLog
  - Commit changes to the new branch
    - $ git add . $ git commit -F <Changelog file>
  - Generate a patch that other people can use to get your changes
    - $ git format-patch -[num] --stdout > formatted-patch.txt
  - Test your partner's patch
    - Check out version 3.0 into a tmp branch
    - Apply patch with git am command: $ git am < formatted-patch.txt
    - Build and test with $ make check
    - Make sure partner's name is in HW9.txt for #8

Gitk

- A repository browser
  - Visualizes commit graphs
  - Used to understand the structure of the repo
  - Tutorial: http://lostechies.com/joshuaflanagan/2010/09/03/use-gitk-to-understand-git/

Gitk

- SSH into the server with X11 enabled
  - ssh -X for OS with terminal (OS X, Linux)
  - Select "X11" option if using putty (Windows)
- Run gitk in the ~eggert/src/gnu/emacs directory
  - Need to first update your PATH
    - $ export PATH=/usr/local/cs/bin:$PATH