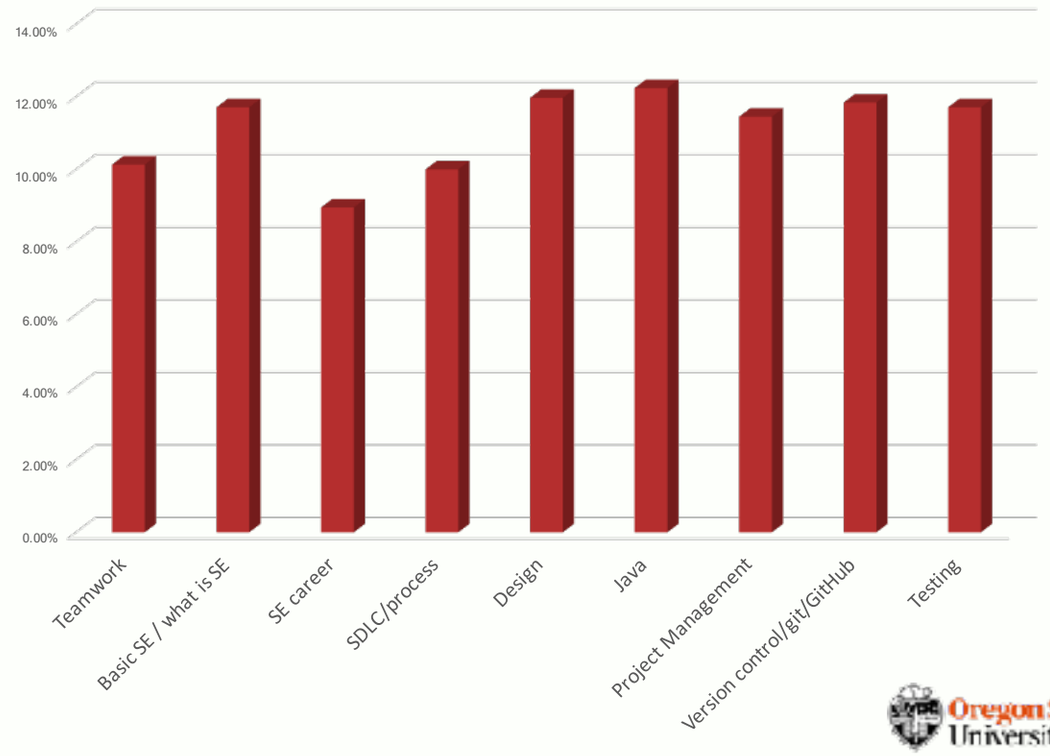
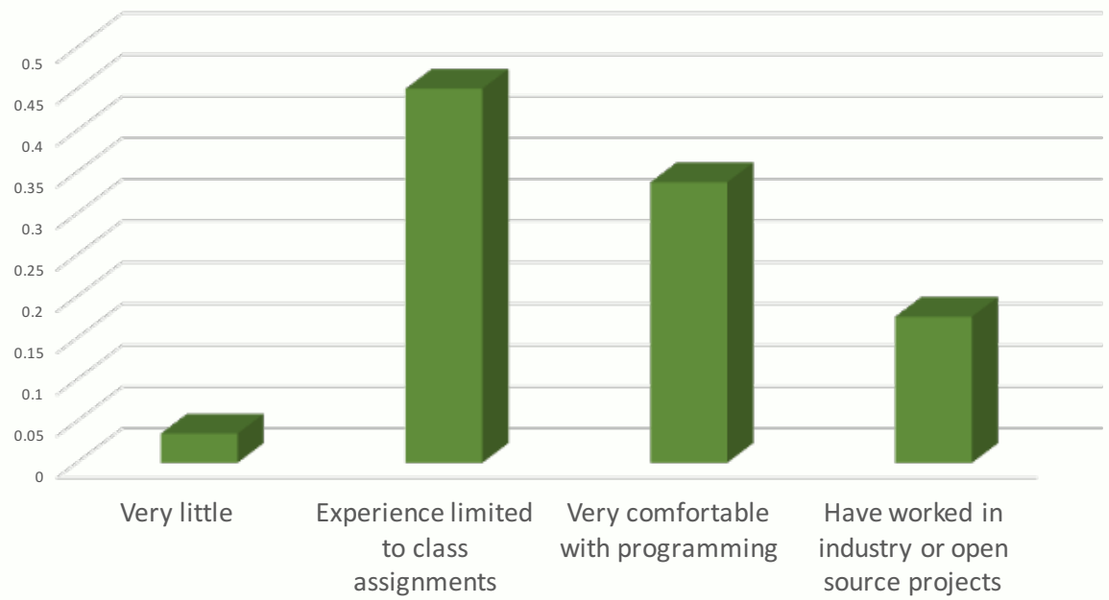
Collaboration

Class expectations



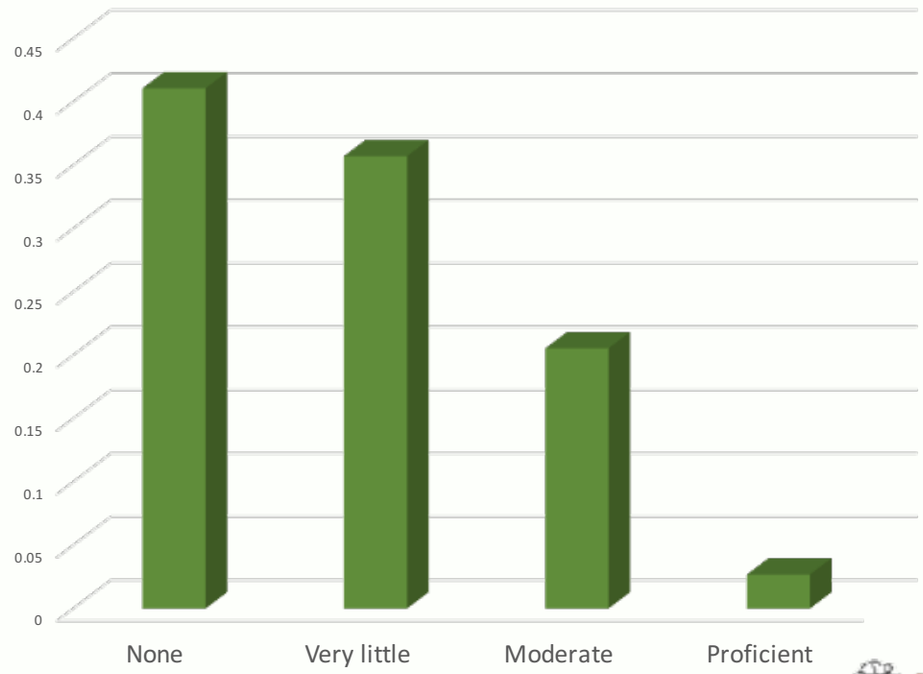
Oregon State University

Programming Experience



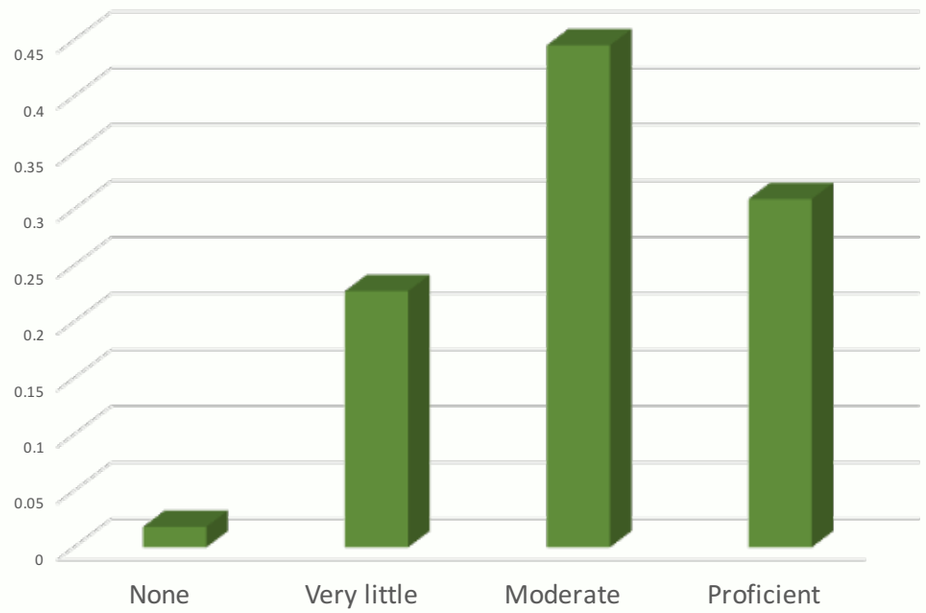
Oregon State University

Java Experience



Oregon State University

Experience Working in Teams



Oregon State University

Announcements

- Team formation deadline is **tomorrow!**
- Sprint 0 deadline is this Friday

This assignment is the only individual assignment; Project Stages 1-4 will be group assignments. Each student is expected to complete this assignment on their own. Please review all instruction steps, and the grading rubric section, before beginning the assignment. Prioritize your efforts accordingly.

1. Sign-up for a GitHub account (preferably using your school email). If you already have a GitHub account, you can skip this step.
2. Post your GitHub username to the Discussion thread titled GitHub Usernames (see thread for instructions on finding your username).
3. Fork the cs361fall2017/project0 repository.
4. Clone the **<your_username>\project0** repository to your local system.
5. Add your name to the students.txt file and commit it to your local git repository.
6. Push the current version of your local git repository to your remote GitHub repository (**<your_username>\project0** repository).
7. Create a new pull request from your remote GitHub repository (**<your_username>\project0** repository) to the cs361fall2017/project0 repository.

Grading Rubric

Item	Points
Posted GitHub username to Discussion thread	5 points

Office Hours

- Anita Sarma: Wed: 11-12
- Nick: Fri: 2-3
- Caius: Thu: 3-4
- Ayda: Fri: 11-12

Oregon State University

This week

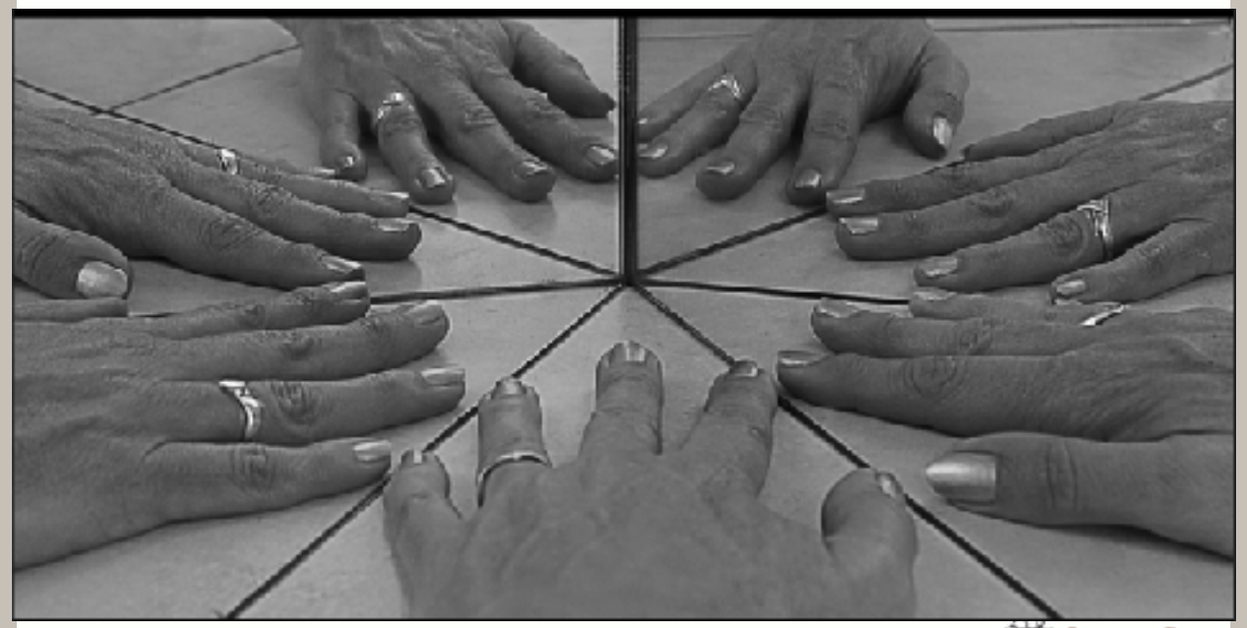
- Collaboration needs
- Version Control Systems – git
- GitHub

Oregon State University

Why teach this, why a central component:

1) you will need this when you go out
2) You will need version control system, even when you work on your own stuff
3) This is something you need to know for your interviews
4) Important to note the distinction between Git and GitHub

Professional Software development



Oregon State University

Dog house -> large project; long living project

Collaboration + rolling back changes

Handling changes

While working on a program you spend the entire day making changes and now it no longer compiles, or has a bug. **Can you retrace your steps to an earlier working code ?**

Handling changes

You are working on a team project. (Don) Joe has made changes to a part of the program but you are still working with his old code. You test your new code with the old code, everything works fine. But, not when you combine it with the new one – it doesn't compile. **Where was the bug introduced?**

Handling changes

You are working on a team project working on writing a complicated test plan. You come back in the morning and you find someone overwrote your changes…**who made the changes? What was changed?....**

Oregon State University

Collaborative software development involves …

…Socio-technical dependencies

Oregon State University

What is version control system

…identify, organize, and control changes to the software … development and maintenance

- Keeps records of your changes
- Allows for collaborative development
- Allows you to know who made what changes and when
- Allows you to revert any changes and go back to a previous state
- Build and release management

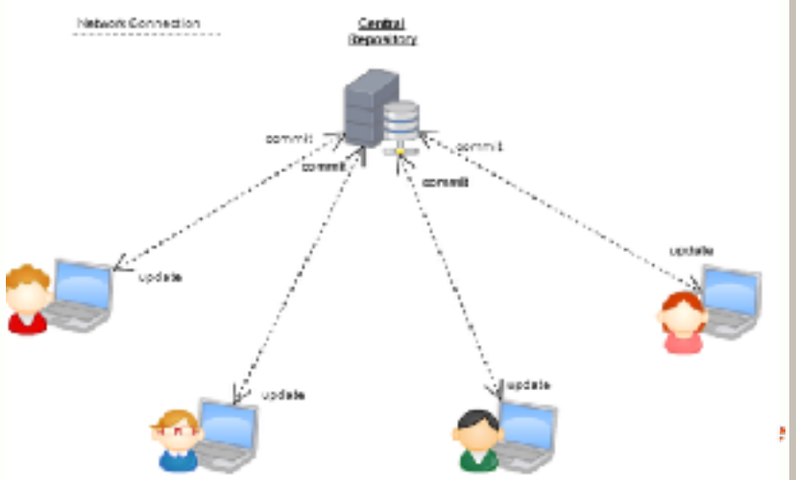
Oregon State University

Types of Version Control Systems

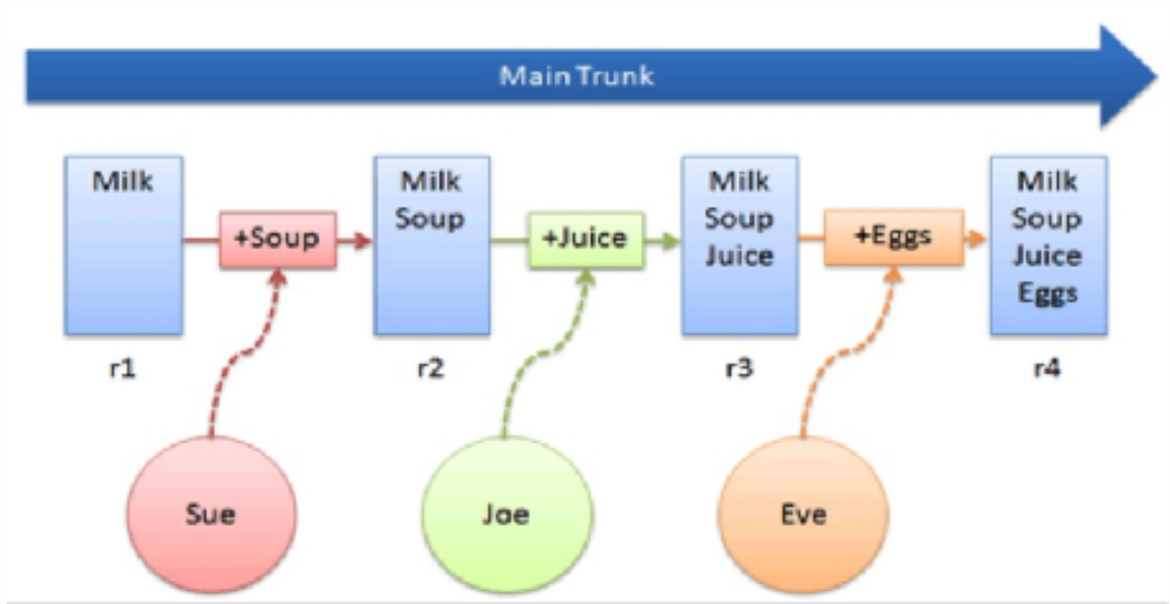
- Make
- **Centralized VCS**
  - RCS, SCCS, ….
  - CVS, Subversion (SVN)
  - Clearcase, Perforce
- **Distributed VCS**
  - Git
  - Mercurial (Hg), Bazaar (Bzr)

Centralized VCS

- Requirements:
  - Internet,
  - Server up & running to commit against to
  - **Improve your code**: *"Commit Early, Commit Often"*



Centralized VCS



Types of Version Control Systems

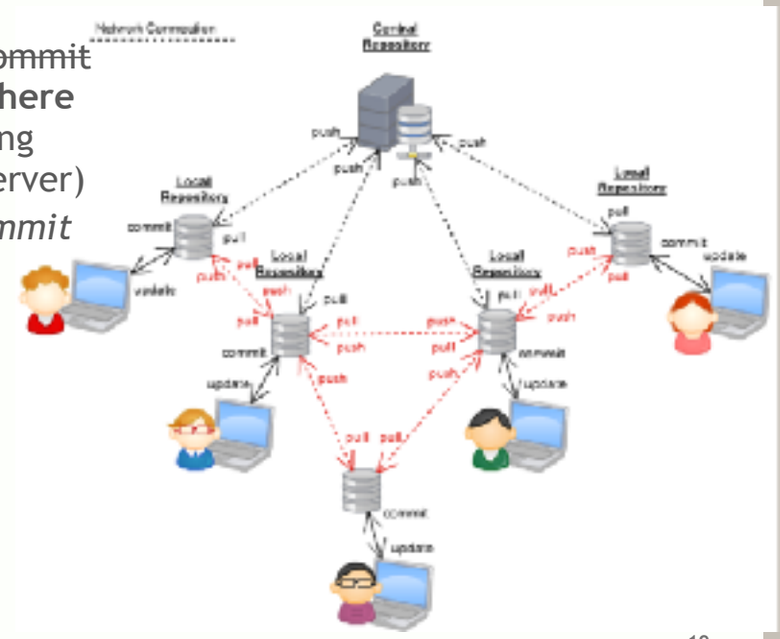
- Make
- **Centralized VCS**
  - RCS, SCCS, ….
  - CVS, Subversion (SVN)
  - Clearcase, Perforce

- **Distributed VCS**
  - Git
  - Mercurial (Hg), Bazaar (Bzr)

Decentralized VCS

Requirements:

- ~~Internet~~,
- ~~Server up & running to commit against to~~, **commit anywhere** (except pushing and pulling changes from a remote server)
- **Improve your code**: *"Commit Early, Commit Often"*

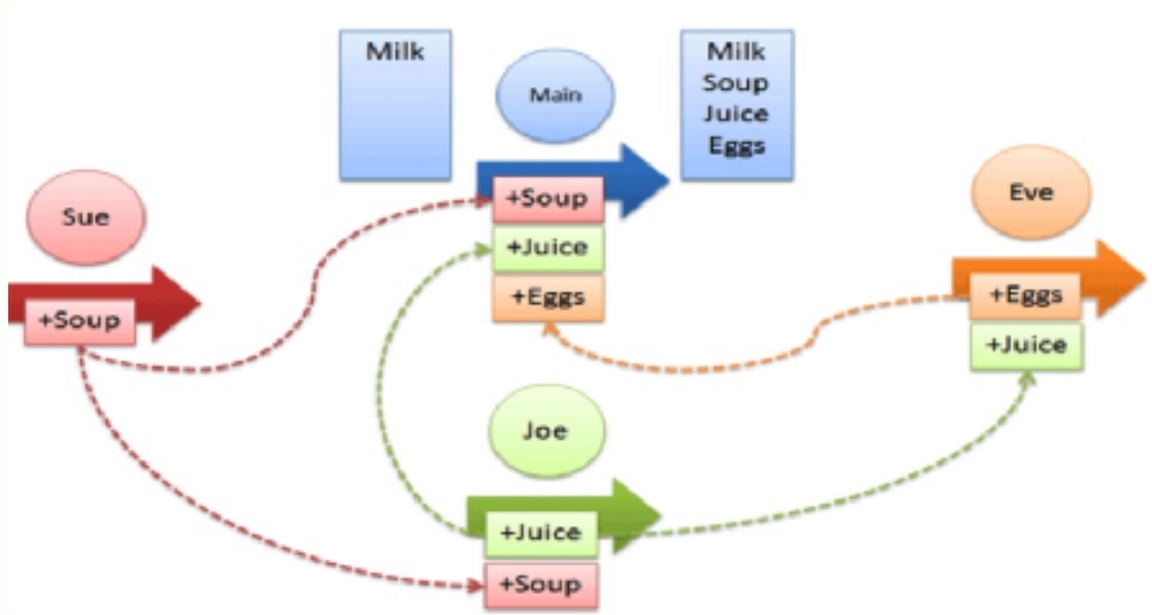


Q: How can you be committing, if you are not committing to a server?

Entire Git repo lives in your file-system - all the meta data.

But, you do need to push to a remote server — so, that the rest of the world can have access

Decentralized VCS



Jeremy Coates
, Founder & Chief Executive Officer (CEO) at Magma Digital Ltd

Git vs. GitHub

- A distributed VCS
- Started in 2005
- Created by Linus Torvalds to aid in Linux kernel development

- A web-based git repository hosting site
- Started in 2007, released in 2008
- Calls itself social coding
- As of April 2016: 14M users, 35M repositories

git

Key Concepts: Snapshots

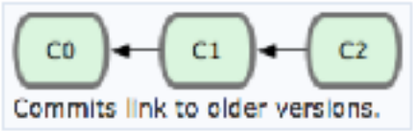
- The way git keeps track of your code history
- Essentially records what all your files look like at a given point in time
- You decide when to take a snapshot, and of what files
- Have the ability to go back to visit any snapshot
- Your snapshots from later on will stay around, too

Oregon State University

Key Concepts: Commit

- The act of creating a snapshot
- Essentially, a project is made up of a bunch of commits

- Commits contain three pieces of information:
  1. Information about how the files changed from previously
  2. A reference to the commit that came before it - called the "parent commit"
  3. A hash code name. Something like: f2d2ec5069fc6776c80b3ad6b7cbde3cade4e



Oregon State University

Key Concepts: Repositories (1)

- Often shortened to ‘repo’
- A collection of all the files and the history of those files
- Consists of all your commits

Oregon State University

Key Concepts: Repositories (2)

- Can live on a local machine or on a remote server (GitHub!)
- The act of copying a repository from a remote server is called cloning
  - Cloning from a remote server allows teams to work together

Oregon State University

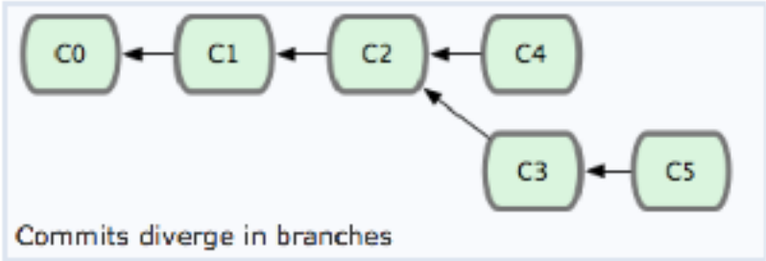
Key Concepts: Repositories (3)

- The process of downloading commits that don't exist on your machine from a remote repository is called pulling changes
- The process of adding your local changes to the remote repository is called pushing changes

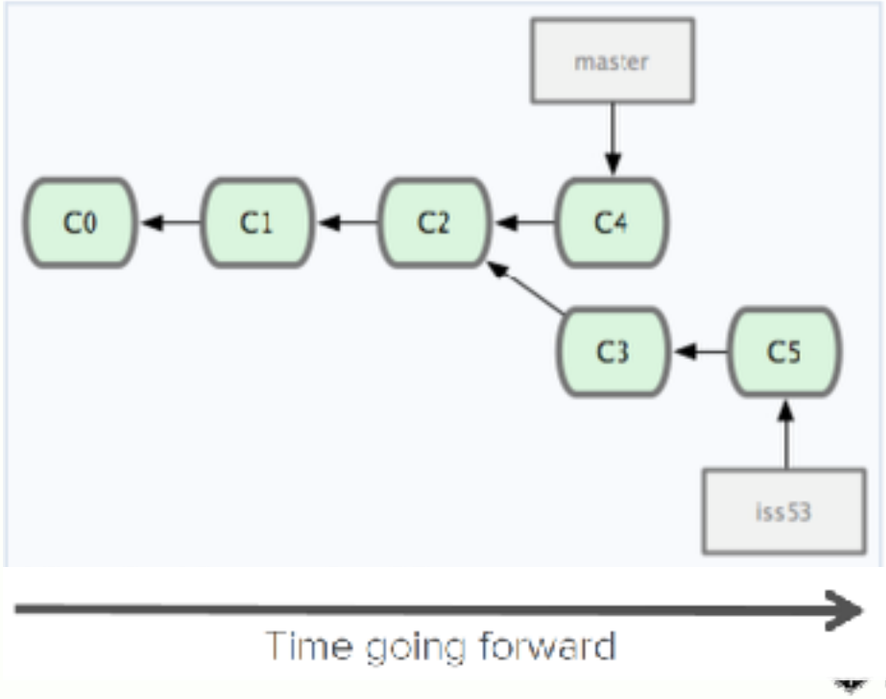
Oregon State University

Key Concepts: Branches

- Allows you to work in parallel
- All commits in git live on some branch
- But there can be many, many branches
- The main branch in a project is called the master branch

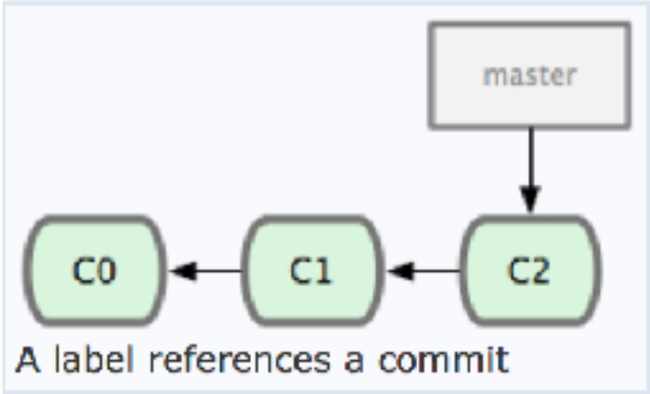


Key Concepts: Branching off of the master branch

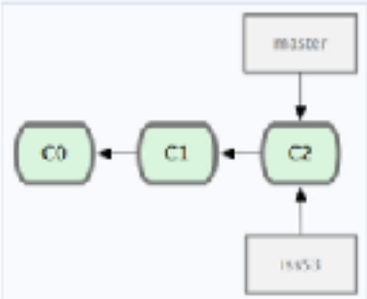


Oregon State University

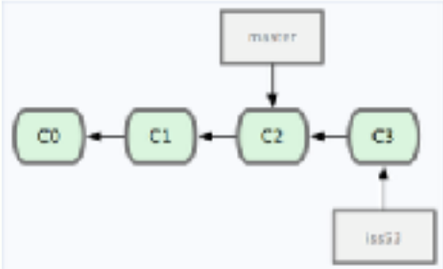
Unpacking Branching: current state



Unpacking Branching: creating a branch, switching to it



1: A new branch is created



2: A commit is added to the branch

Making a branch, and committing it in:

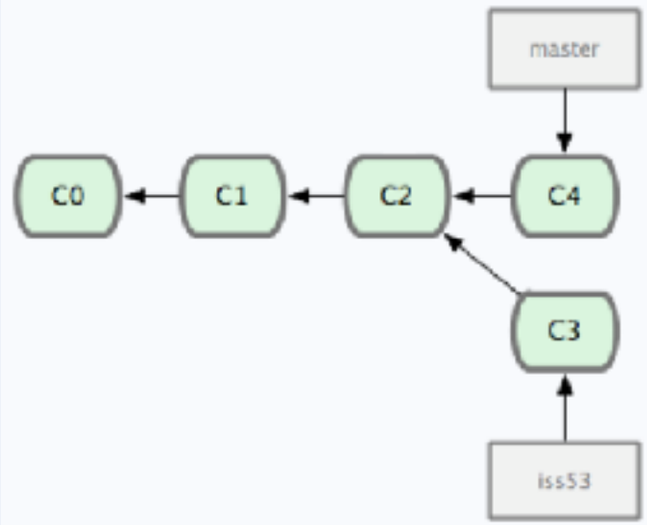
```
git branch iss53        # create the branch (see figure 1)
git checkout iss53      # switch to it

...                     # edit a file

git commit -a           # commit the change in the new branch
```

Oregon State University

Unpacking Branching: switching back to MASTER



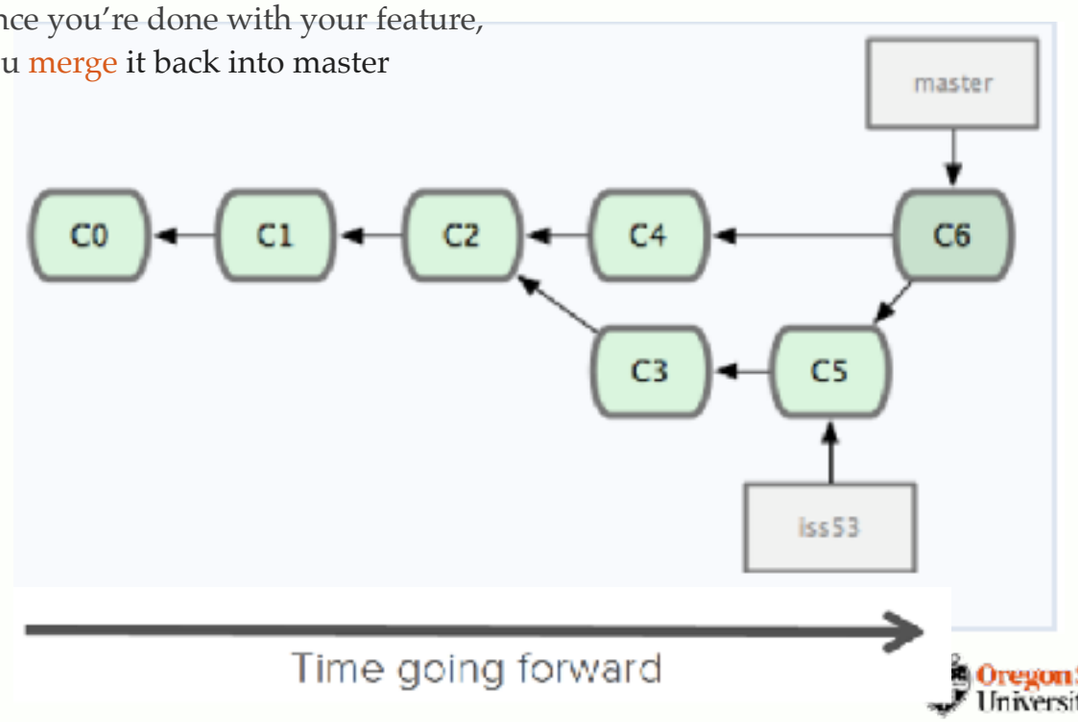
Result after making a new commit in the master branch

Switching back to master:

```
git checkout master
```

Key Concepts: Merging

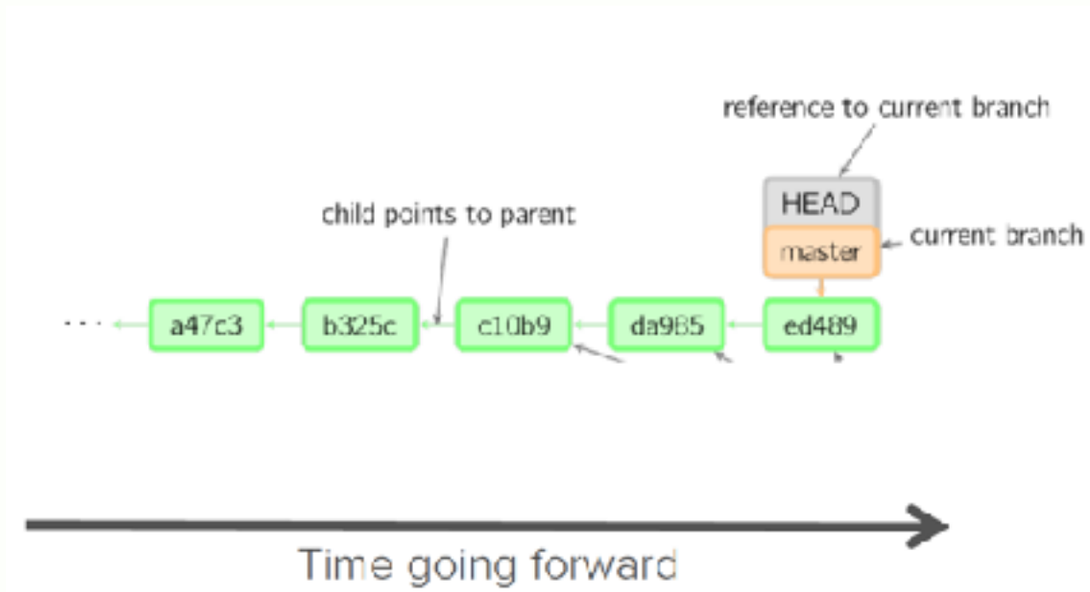
Once you're done with your feature, you merge it back into master



So, what does a typical project look like?

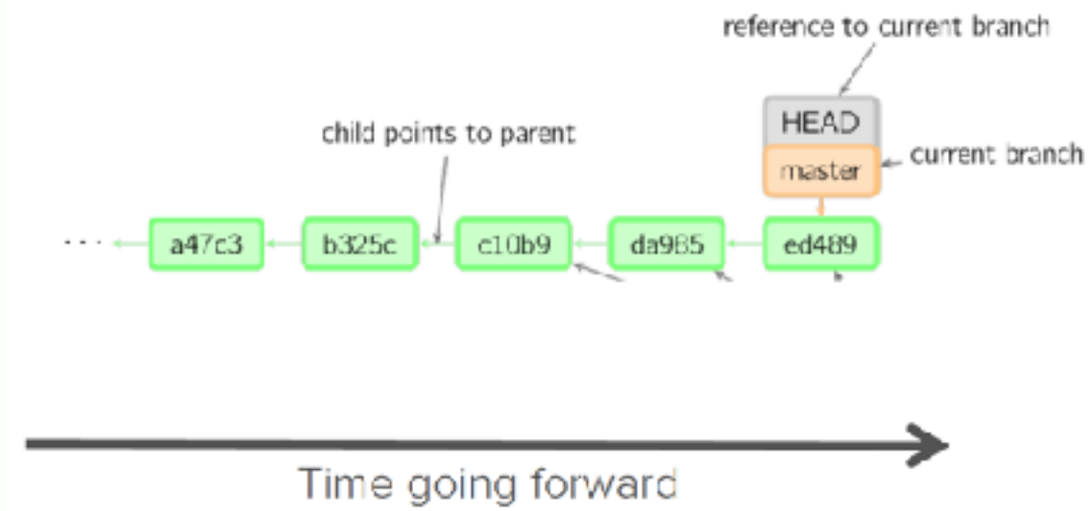
- A bunch of commits linked together that live on some branch, contained in a repository
- Following images taken and modified from:
  - http://marklodato.github.io/visual-git-guide/index-en.html
  - Also a good tutorial!

So, what does a typical project look like?



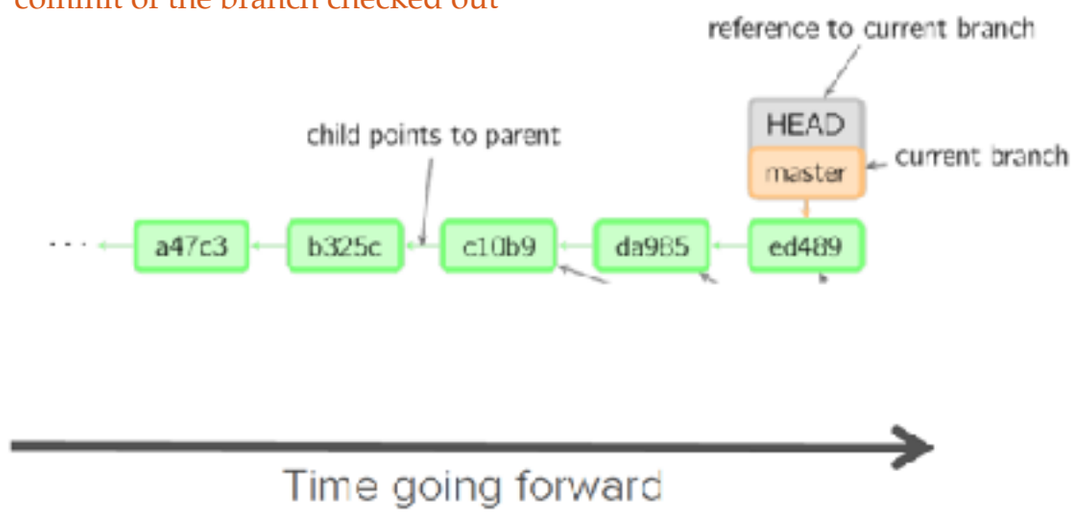
So, what is MASTER?

The main branch in your project
Doesn't have to be called master, but almost always is!



So, what is HEAD?

A reference to the most recent commit of the branch checked out



HEAD: refers exclusively to the currently active head (generally the tip of the branch)

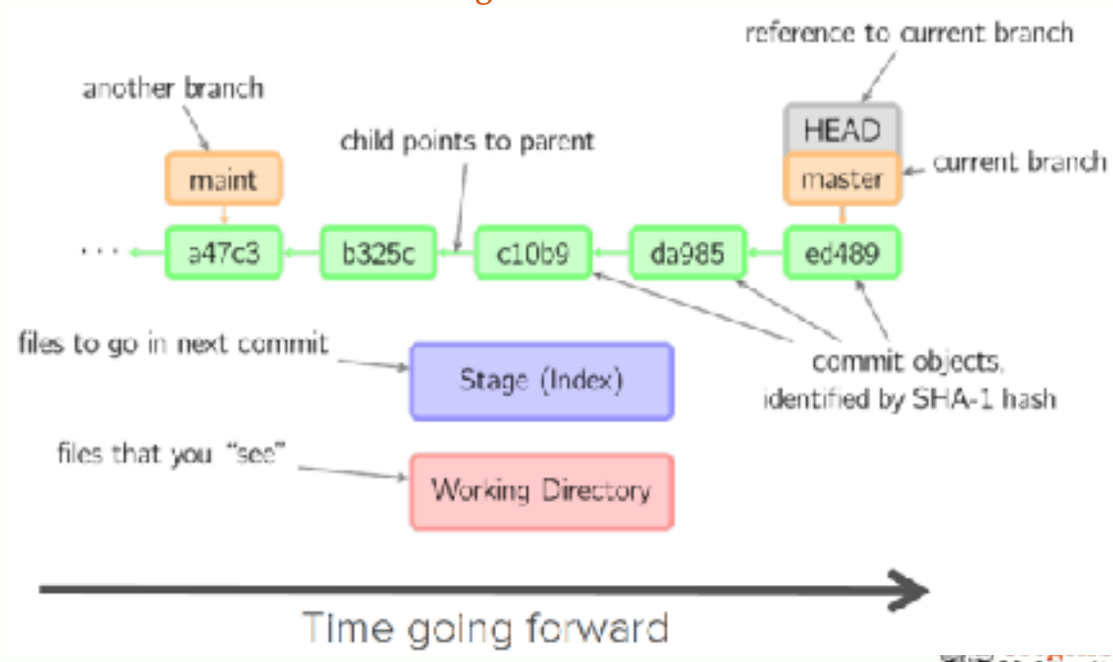
Key Concepts: How do you make a commit anyway?

- There are a lot of ‘states’ and ‘places’ a file can be
- Local on your computer: the ‘working directory’
- When a file is ready to be put in a commit you add it onto the ‘index’ or ‘**staging**’
- The process:
  - Make some changes to a file
  - Use the ‘git add’ command to put the file onto the staging environment
  - Use the ‘git commit’ command to create a new commit’

Oregon State University

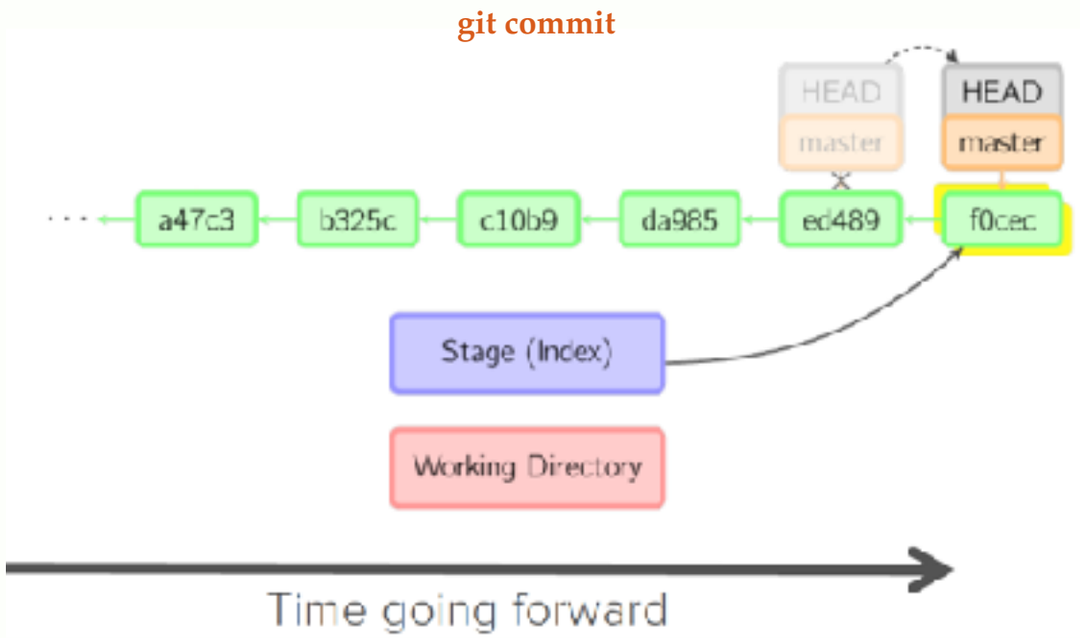
Key Concepts: How do you make a commit anyway?

git add

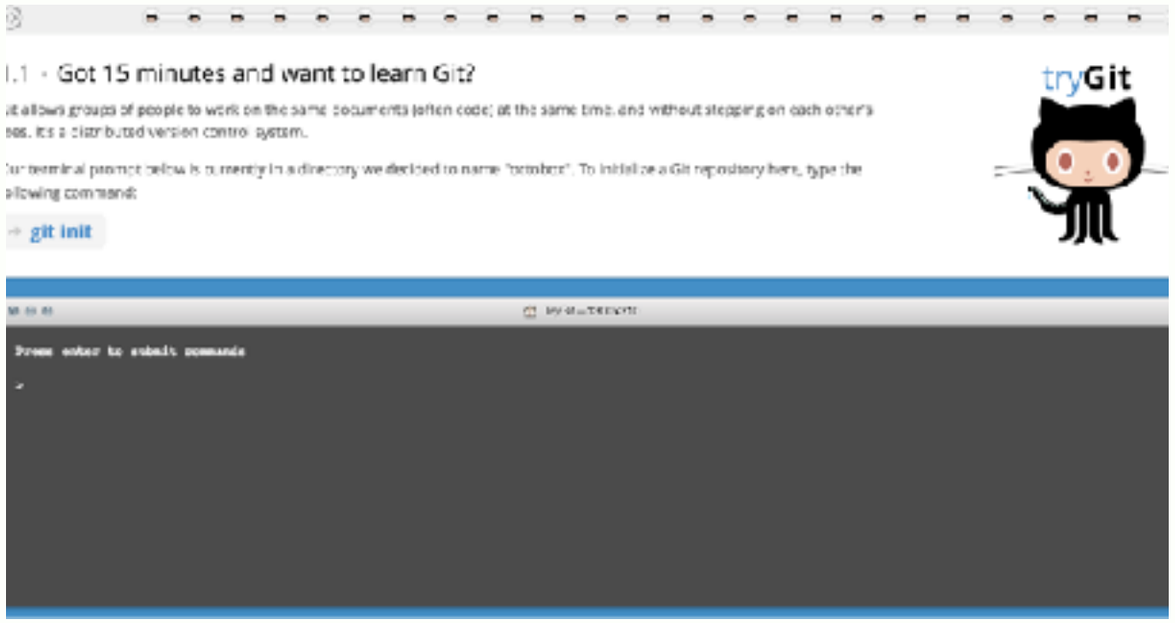


git add

Key Concepts: How do you make a commit anyway?



Demo



Other concepts

- Conflict
- Backport
- Cherrypick

Useful, step by step guide: http://codingdomain.com/git/