Git 101
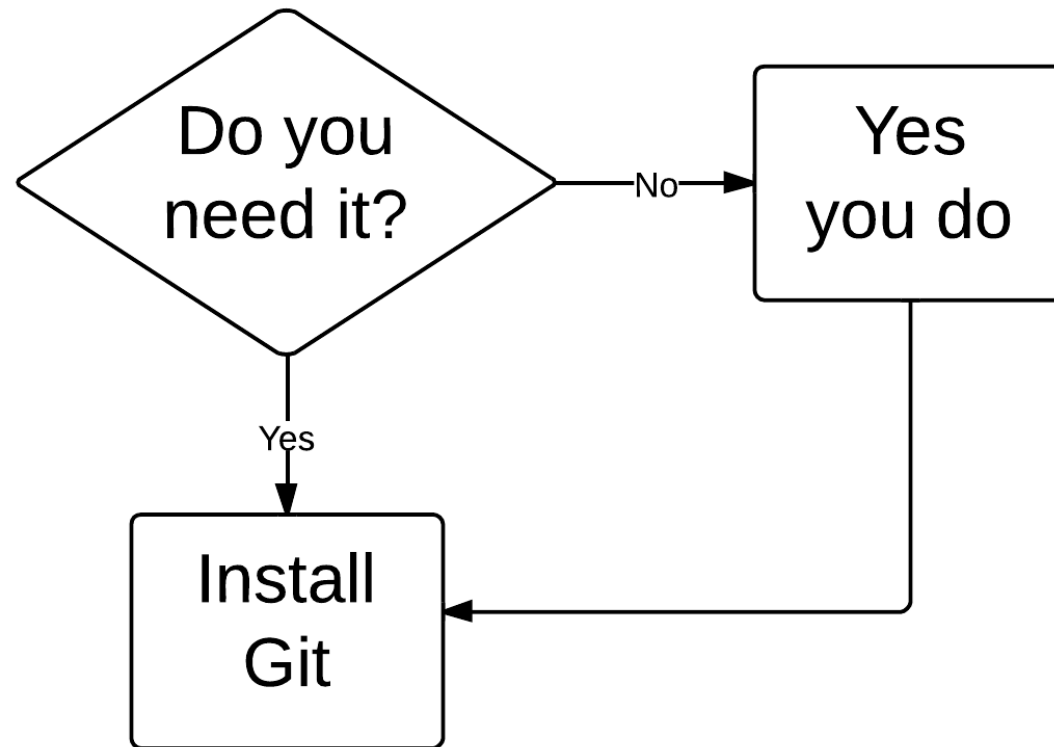
OR: GIT FOR GITS

DAVID MUCKLE

Version Control Flowchart

What is Git?

- VCS (Version Control Software)
  - CVS, Mercurial, Git
- Keep track of changes
- Revert back to old changes
- Keep multiple versions of a project
- Merge versions
- Synchronize projects between machines
- Keep your project "in the cloud"

History of Git

- ▶ Use in Linux development
  - ▶ Replacing BitKeeper
- ▶ Use online
  - ▶ GitHub, BitBucket

The name "git" was given by Linus Torvalds when he wrote the very
first version. He described the tool as "the stupid content tracker"
and the name as (depending on your way):

 - random three-letter combination that is pronounceable, and not
   actually used by any common UNIX command. The fact that it is a
   mispronunciation of "get" may or may not be relevant.
 - stupid. contemptible and despicable. simple. Take your pick from the
   dictionary of slang.
 - "global information tracker": you're in a good mood, and it actually
   works for you. Angels sing, and a light suddenly fills the room.
 - "g*dd*mn idiotic truckload of sh*t": when it breaks

The Git Basics

- git init
  - git clone
    http://example.com/repo
    - Use http or ssh
- git add somefile.thing
- git status
- git commit -m 'Comment'
- git push
- git pull
- git branch branchname
- git checkout branchname
- git merge

	David Muckle	a81acfe		Cleaned up for submission	3 days ago
	David Muckle	37523f2	M	Merged homerun into master	3 days ago
	David Muckle	2466872		Finished readme	3 days ago
	David Muckle	8f3f553		Possible fix for R	3 days ago
	David Muckle	59248fa		Fix elim check	3 days ago
	David Muckle	0aa038d		Readme progress	3 days ago
	David Muckle	0bff397		Added comments and check for input with only one team	3 days ago
	David Muckle	5d1af3f		Added better formatting	3 days ago
	David Muckle	0a7ba1a		Added exceptions	3 days ago
	David Muckle	38f0ef8		Finished isEliminated and certificateOfElimination	4 days ago
	David Muckle	0ab946c		Added readme	4 days ago
	David Muckle	a32ea20	M	Merged homerun into master	4 days ago
	David Muckle	d8d5d10		Added cert stuff for trivial	4 days ago
	Nick Bardjis	c66e74d		Constructed two halves of FlowNetwork	4 days ago
	Nick Bardjis	9eae7e4		Constructed two halves of FlowNetwork	4 days ago
	Nick Bardjis	41f8a31		Started mathematical elim	5 days ago
	David Muckle	acc6cc8	M	Merged homerun and master	5 days ago
	David Muckle	e60f47a		Fixed merge	5 days ago
	Nick Bardjis	638fefd		Update	5 days ago
	David Muckle	a5834ef		Added trivial elimination	5 days ago
	Nick Bardjis	d51b31c		Added constructor	5 days ago
	David Muckle	4312d1b		Added project property files	2016-12-09
	David Muckle	45e88f0		Added skeleton	2016-12-09
	David Muckle	093a437		Initial commit	2016-12-09

git init and git clone

- git init for a new project
  - Run in a folder with all your project materials
  - Creates a .git folder within your project folder, DON'T TOUCH THIS
- git clone for an existing project
  - Clone the project from some online source like BitBucket or Github.
  - git clone https://github.com/username/project
  - git clone ssh://git@github.com/username/project

Aside: SSH and Git

- SSH can be used to facilitate cloning and other stuff with online projects
- Uses a keypair for authentication
  - Public key and private key
  - `ssh-keygen` to make a new keypair
  - Put your public key on things you want to "unlock" with your private key
  - Keep your private key *safe*
- Put your public key on Github or Bitbucket to clone and interface with projects without having to enter your username and password every time

git add

- Tell Git to start "tracking" a file
  - Can have tracked and untracked files in a repository
  - Only tracked files will be saved by Git
- `git add .`
  - Tell Git to add all files in your repository, including files in other folders
- `git add *`
  - Add only the files in the folder you're currently in
- `git add filename`
  - Just add one specific file
- `git rm filename`

git status

- Tells you what's tracked, untracked, what's ready to commit, status of upstream remotes, branch…

- Useful after doing any git command to double check everything's good

- A way to get your bearings in a repo

git commit

- Git's "save" button
- Makes a save of all the files you currently have tracked, and changes to anything you've changed and are still tracking
- Needs a comment to go with it
  - `git commit -m 'Commit comment'`
  - Otherwise a text editor like vim or nano will open up
    - This is annoying. Just use –m.
- Git creates a unique hash value of your commit to refer to later
  - CS160 people, what does that mean? ☺

git push

- Upload your changes to an online "remote"
- If you cloned from an existing "remote," you don't need to set up a remote
  - `git push -u origin master`
- `git remote add origin https://bitbucket.org/username/repo`
  - If you started a project locally using `git init`
- Git might ask you some config settings.
  - Defaulting to "matching" is fine

git pull

- Download new changes from online
- Useful to do before working on a project, to avoid conflicting changes
  - Awesome if you're working with another person
- Will default to the information that you used during `git clone` or `git push`

git branch and git checkout

- `git branch branchname`
  - Creates a separate copy of your project from the current point in time
  - Git creates the 'master' branch by default
  - Need to create a new branch on the remote as well
- `git checkout branchname OR git checkout commithash`
  - Switch to a branch OR revert back to an old commit
  - Make sure to commit your changes before switching branches, else you will lose said changes
    - `git stash`
      - Makes a temporary commit. Where does it go? Use `git stash list`
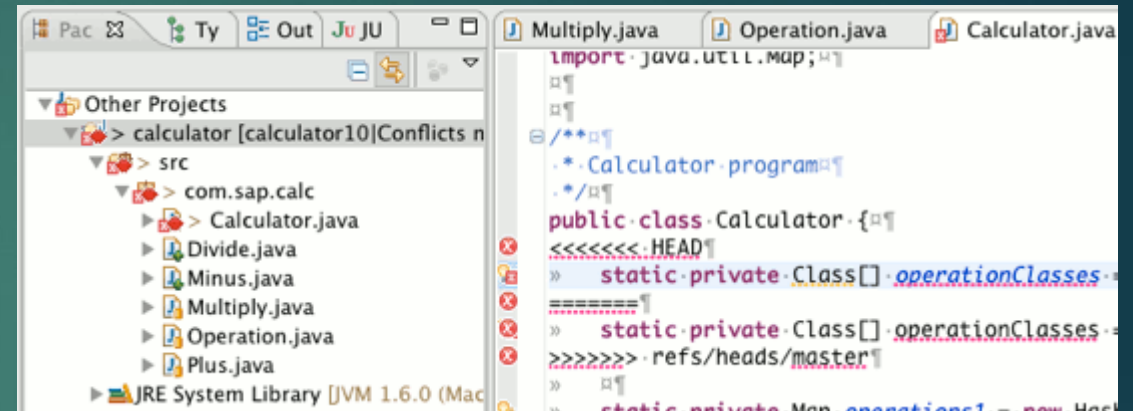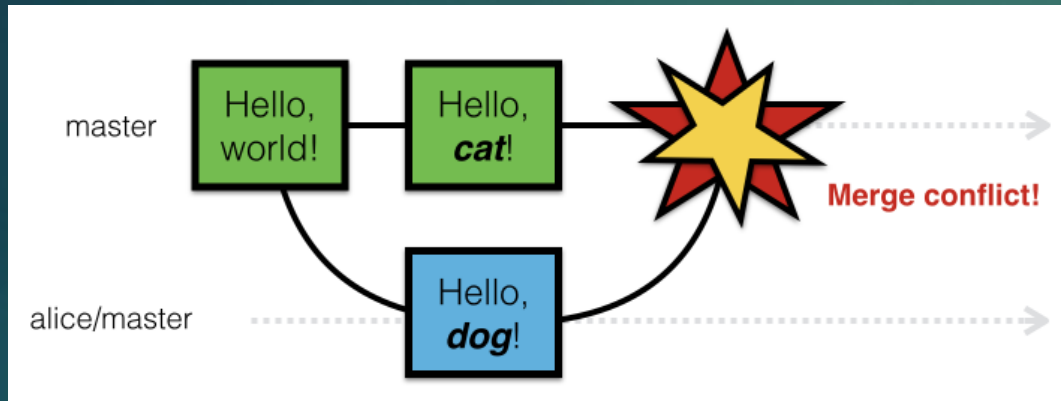   - `git stash apply` to go back to a stash

git merge

- Merge two (Or more!) branches together
  - Happens automatically if you do a pull while having local commits not in the remote
  - If you're on the `master` branch and do `git merge dev`, it will pull in the changes from `dev` and apply them to `master`
  - Also good to use `-m 'Comment'` with as well, same as `git commit`
- Useful for working on stuff that doesn't work yet, then merging it into a branch where everything does work
- Still keeps the branch you merged changes from
- Can be done with more than one branch (Octopus merge)
  - Linux kernel once did a 66 way merge!
- What happens if two files have a different line in both branches?

Other commands

- `git diff commithash commithash` OR `filename filename`
  - Show the difference between two files, two commits, two branches…
  - Nice and colorized!
- `git blame filename`
  - Show, line by line, who did what in a file
  - Find out who to ~~blame for a bug~~ thank for a feature
  - Also shows commits something happened in
- `git reflog`
  - A "time machine" of all things you've done in Git

What to do if Git breaks
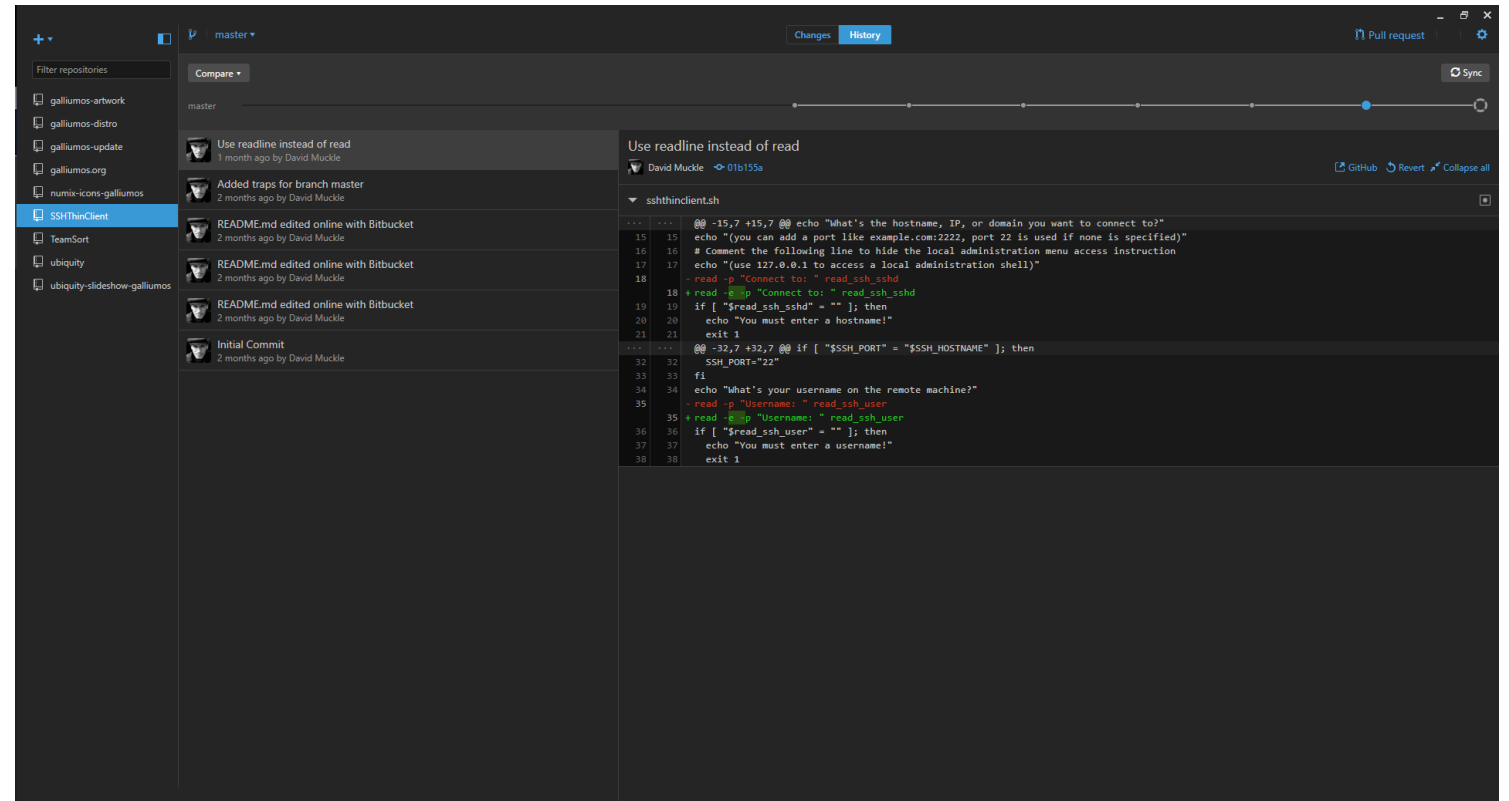
Resolving Merge Conflicts

- Stay calm
- Read the marked changes
- Talk with your partner(s)
- Figure out what to keep
- Edit accordingly
- Push changes

Graphical Clients

- Integration with IDEs
  - Eclipse, Netbeans, IntelliJ
- Standalone Clients
  - GitHub Client, Atlassian SourceTree, Git GUI

Suggested Setup for Pair Programming

- Use BitBucket
- Create repository with private access
- Add members to repository

OR

- Create team with partner
- Create repository under team

Using with your Clark CS Account

▶ Create a repository on system
▶ Commit changes
▶ On local computer, `git clone ssh://username@csgateway.clarku.edu/home/username/pathtorepo`
▶ Won't work with pair programming

Can I use Git for other things?

- Absolutely!
- Essays
- Cover letters
- Image editing
- Recipes
- Program settings (dotfiles)
- This PowerPoint
- Aside: Git works best with text files, but can still be used with non-text files

Other Git Resources

- Oh Shit, git! http://ohshitgit.com
  - Git will break. Here's how you can fix it
- Gitless http://gitless.com/
  - A nice "porcelain" to use with Git
- Github
  - The hub… For Git… It's in the name folks
  - Has some helpful docs and is *the place* for open source projects
- Atlassian (BitBucket)
  - Has some more docs, slightly better formatting than Github
- `git command --help`
  - Pulls up a webpage or "man" page on any git command, complete with all the options you can use.

Go forth and Commit!