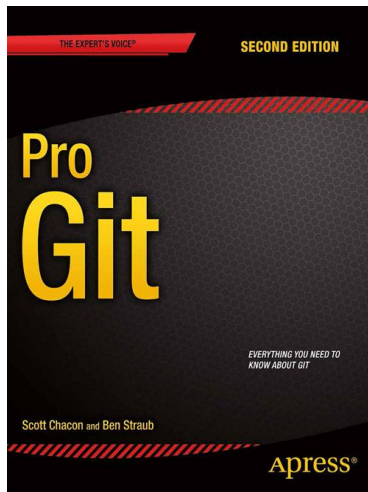


Git remote repositories

On GitHub

Attribution

- Slides attributed to Dr. Cyndi Rader. The original version can be found at <http://eecs.mines.edu/Courses/csci306/ChapterNotes.html>
- Figures and info for this lecture come primarily from Pro Git, available: <http://www.git-scm.com/book/en/v2>
- Material is used in accordance with the Creative Commons Attribution Non Commercial Share Alike 3.0 license.



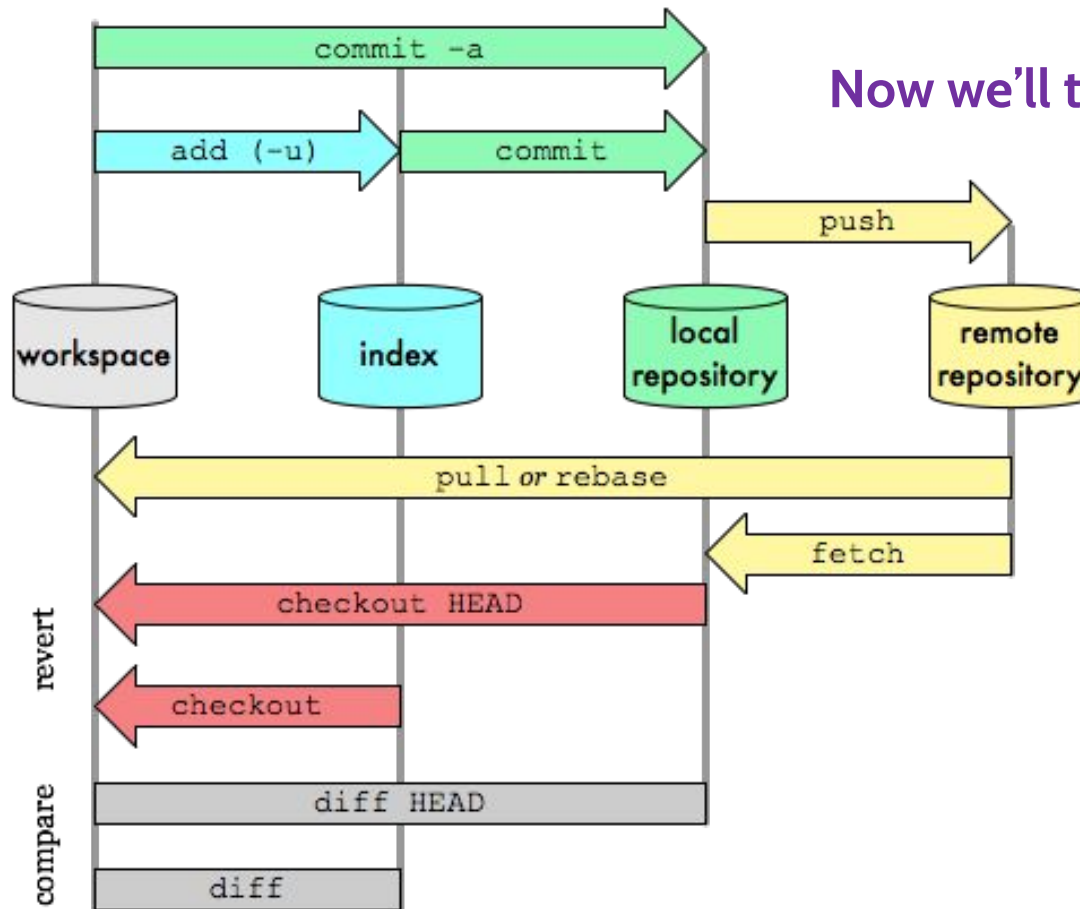
Book available online for free at: <http://git-scm.com/book/en/v2>



The Big Picture

Git Data Transport Commands

<http://osteele.com>



Now we'll talk about this part

<http://blog.mikepearce.net/2010/05/18/the-difference-between-git-pull-git-fetch-and-git-clone-and-git-rebase/>

GitHub

- Largest open source git hosting site
- Public and private options
 - Public: everyone can see, but must be a collaborator to modify files
 - Private: only those you choose as collaborators can see/update
- Public repos often used for large open source or to distribute libraries etc.





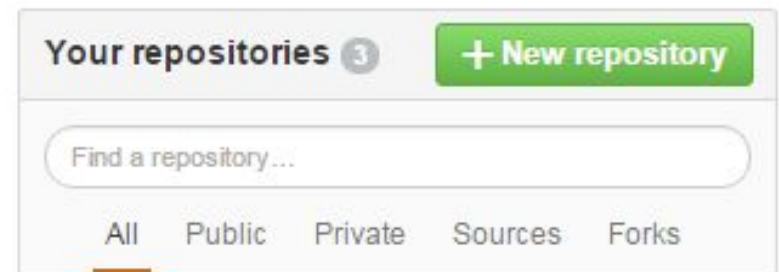
Getting an account

- Set up a user account
 - Public account is free
 - Remember your password!
- Get an educational account
 - Students get 5 private repos
 - After your account is created, send a discount request (may take a week or so to get a response)
 - Discount button on: <https://education.github.com/>

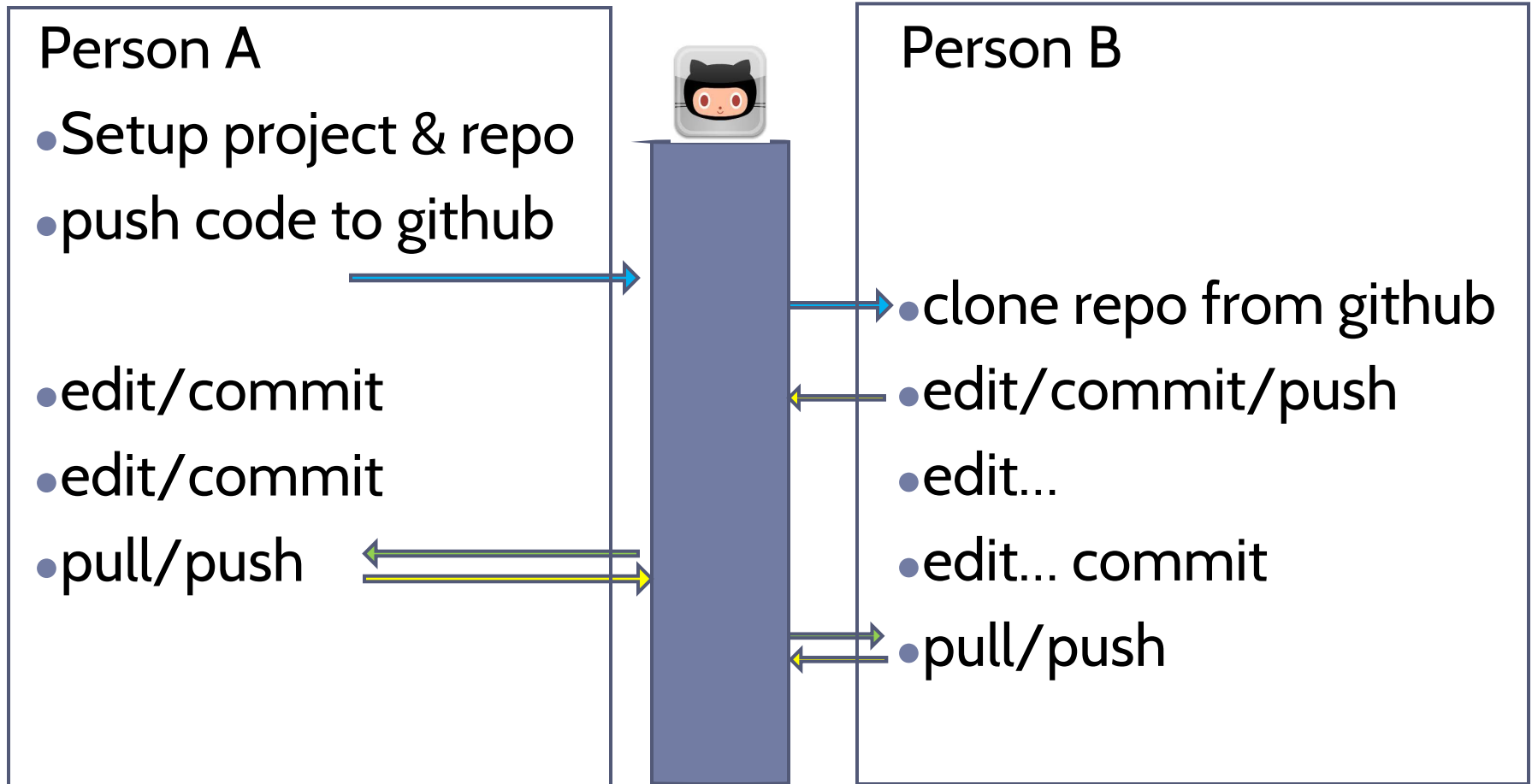
A screenshot of the GitHub sign-up form. It features three input fields: 'Pick a username', 'Your email', and 'Create a password'. Below the password field is a note: 'Use at least one lowercase letter, one numeral, and seven characters.' A green button labeled 'Sign up for GitHub' is positioned below the form. At the bottom, a disclaimer states: 'By clicking "Sign up for GitHub", you agree to our terms of service and privacy policy. We will send you account related emails occasionally.'

Remote repo on GitHub

- Remote repo is a *bare repository*, no working directory
- Four transfer protocols
 - https is the most simple (but requires login for transfer)
 - Repo URL: https://github.com/[username]/[repo-name].git
 - git protocol
 - https://github.com/
 - [username] userID
 - /[repo-name]. repository
- You should always copy the repo URL from the GitHub page to avoid typos
- Setting up the remote repo
 - Create repo on GitHub
 - Add collaborators if any



Typical workflow with remote repo



This is just the flow, specific commands on following slides.

It's also possible to create your project first on github, then clone (i.e., no git init)

Setup a project and local repo

NetBeans	Git client
Create a Java Project	
	Initialize local repo in the project's root directory
Create .gitignore	
	Check status
	Stage .gitignore
	Commit changes with message "Initial commit"



Upload the initial commit to remote repo

- Add a “shortname” for your GitHub repository
 - `$ git remote add [shortname] [url]`
 - `$ git remote add origin https://github.com/[username]/[repo-name].git`
 - **origin** is the standard short name for the remote repo
 - Copy/paste the repo URL from GitHub to avoid typos (in Git Bash press Insert)
- Upload your initial commit onto GitHub
 - `$ git push -u [remote-name] [branch-name]`
 - `$ git push -u origin master`
 - Note: Use option `-u` to establish tracking between local and remote branch



Saving GitHub credentials

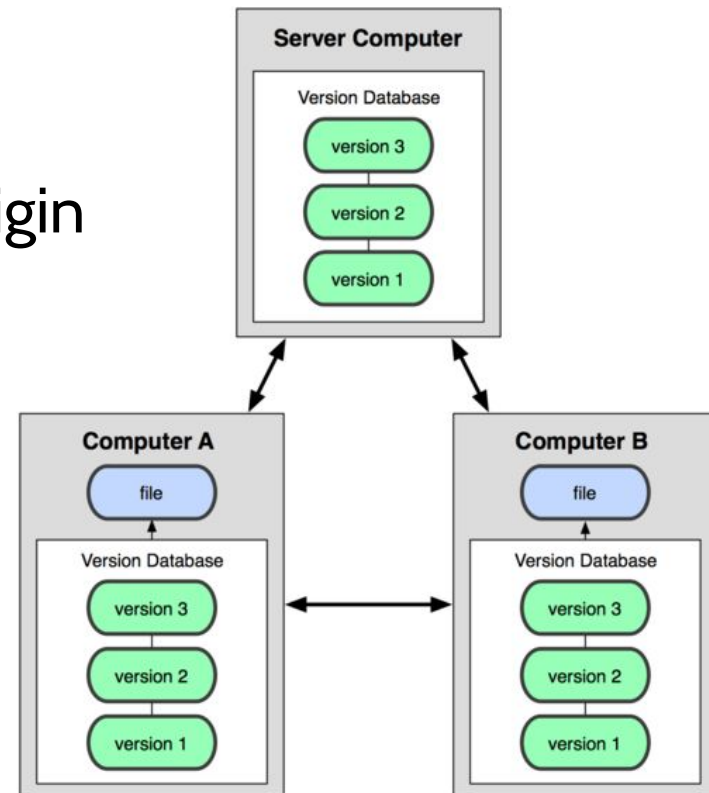
- To save (and avoid typing over and over again) credentials needed for git clone, fetch, pull, or push command, you can make use of the *credentials helper* as follows:
- Win
 - `$ git config --global credential.helper wincred`
- Linux
 - `$ git config --global credential.helper cache`
 - `$ git config --global credential.helper "cache --timeout=3600"`
- Mac
 - `$ git config --global credential.helper osxkeychain`

See: <http://stackoverflow.com/questions/5343068/is-there-a-way-to-skip-password-typing-when-using-https-github>



Creating a local repo by cloning remote

- `$ git clone [url]`
- `git clone` makes a local copy of the remote repository under the name `origin` (i.e. `$ git remote add origin [url]` is done automatically)
- `git clone` also establishes tracking between the local and remote repos (no need to use `-u` option in `git push`)
- Example:
`$ git clone https://github.com/hannum/git-intro.git`



Hands-on with cloned repo

- Get yourself added as a collaborator in a GitHub repo of a NetBeans project created by your team mate
 - Need to be a collaborator to push changes
 - Settings -> Collaborators
- Get the URL of the repo
- Start git client
- Clone the repo to your NetBeans projects folder
- Change directory to the local repo
- Check status
 - Note that you are on the branch “master”



Hands on: making changes locally

- Create the project in NetBeans:
File -> Open Project -> [cloned_project]
 - Note: If the remote repo would contain only Java Sources:
File -> New Project -> Java Project with Existing Sources -> ...
- Make a change to one of the files in NetBeans
- Go back to git client
 - NetBeans comes with Git but command line is good to know as it allows you to version control any type of directories
- Check status
- See what has been changed (git diff)
 - compare the working directory with the stage
- Commit the changes with a meaningful message



Hands on: uploading to remote repo

- Check status to verify that there is nothing to commit and the working directory is clean
- Upload the master branch to remote repo
 - `$ git push [remote-name] [branch-name]`

```
$ git push origin master
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (5/5), done.
Writing objects: 100% (7/7), 809 bytes | 0 bytes/s, done.
Total 7 (delta 2), reused 0 (delta 0)
To https://github.com/hannum/GitIntro.git
   9eb70a5..3016fbb  master -> master
```

- Check the status

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```



Synchronizing local repo with remote

- Two options:
 1. Fetch - merge
 2. Pull
- Pull combines fetch and merge
- If you're careful to avoid conflicts, and you're working on the master branch, easier to just pull
- Be very careful not to make conflicting changes!
- Discuss and co-ordinate before making (major) changes
- Eventually there will be conflicting changes and you will need to merge with conflicts (this will be covered later)



Hands on: downloading changes

- Change back to your original directory
- Check status and note that you are on branch “master”
- Pull changes from the remote repo where your team mate has pushed some changes (in branch “master”)
 - `$ git pull [repository] [branch]`
 - `$ git pull origin master` (params can be omitted if branches

```
$ git pull origin master
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/hannum/GitIntro
 * branch                master      -> FETCH_HEAD
   3016fbb..ae5fbd0      master      -> origin/master
Updating 3016fbb..ae5fbd0
Fast-forward
 README.md | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

```
$ git status
On branch master
Your branch is up-to-date with 'origin/master'.
nothing to commit, working directory clean
```


Hands on: push new changes to remote

- The pull you did ownloaded the latest revision of files from remote branch and merged them in the local branch, as well as checked out the branch
 - Latest versions of the files (the changes made by your team mate in this case) are now in your working directory
- Make some additional changes in the project
- Stage the changes
- Commit
- Push to remote



Hands on: push new changes to remote

- If someone has pushed new changes while you are working on yours, the push does not go through.

```
$ git push origin master
To https://github.com/hannum/GitTest.git
 ! [rejected]          master -> master (non-fast-forward)
error: failed to push some refs to 'https://github.com/hannum/GitTest.git'
hint: updates were rejected because the tip of your current branch is behind
hint: its remote counterpart. Integrate the remote changes (e.g.
hint: 'git pull ...') before pushing again.
hint: see the 'Note about fast-forwards' in 'git push --help' for details.
```



Hands on: checking changes in remote

- You can check if there are changes in the remote:
 - `$ git fetch origin`
 - `$ git status`

```
$ git fetch origin
remote: Counting objects: 3, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/hannum/GitTest
 842a12a..e4b02cb  master    -> origin/master

hannutam@ETY-B227-HANNUT MINGW64 ~/Documents/NetBeansProjects/GitTest (master)
$ git status
On branch master
Your branch is behind 'origin/master' by 1 commit, and can be fast-forwarded.
  (use "git pull" to update your local branch)
nothing to commit, working directory clean
```

- Also:
 - `git log master..origin/master` //Note the order of params
 - `git diff master..origin/master`
-



What does fetch do?

- `$ git fetch origin`
- Remote branches are *not direct links* to remote repositories
- They are read-only copies of remote branches, stored in your own repository
 - `.git -> refs -> remotes`
- This means that you must do another fetch to access new updates





Examples

To work on your own

Example of using cloned repo

- Assume I have a GitHub repo named GUI2.git

- Clone it:

```
Cyndi@CYNDI-PC /c/csm_temp/help
$ git clone https://github.com/CyndiRader/GUI2.git
Cloning into 'GUI2'...
remote: Counting objects: 8, done.
remote: Compressing objects: 100% (7/7), done.
remote: Total 8 (delta 0), reused 8 (delta 0)
Unpacking objects: 100% (8/8), done.

Cyndi@CYNDI-PC /c/csm_temp/help
$ ls
Demo  GUI2
```

- Create project in NetBeans:
 - If repo is a NetBeans project: File -> Open Project -> GUI2.git
 - If it contains only Java Sources: File -> New Project -> Java Project with Existing Sources -> ...



Push example

- Make a change to one of the files, check the changes

```
$ git diff
diff --git a/src/GUIDemo.java b/src/GUIDemo.java
index e57d6c8..fd99d1f 100644
--- a/src/GUIDemo.java
+++ b/src/GUIDemo.java
@@ -7,7 +7,7 @@ public class GUIDemo extends JFrame {
     public GUIDemo()
     {
         setSize(new Dimension(300, 400));
-        setTitle("My Font Viewer");
+        setTitle("My Font Viewer!");
         DisplayPanel dp = new DisplayPanel();
         add(dp, BorderLayout.CENTER);
         JPanel controls = createControlPanel(dp);
```

- See status (I already made one commit)

```
Cyndi@CYNDI-PC /c/csm_temp/help/GUI2 (master)
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 1 commit.
#
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   src/GUIDemo.java
#
no changes added to commit (use "git add" and/or "git commit -a")
```

Push example continued

- Commit changes – updates local repo

```
Cyndi@CYNDI-PC /c/csm_temp/help/GUI2 (master)
$ git commit -am "Updated title of GUI"
[master 1a325f8] Updated title of GUI
1 file changed, 1 insertion(+), 1 deletion(-)

Cyndi@CYNDI-PC /c/csm_temp/help/GUI2 (master)
$ git status
# On branch master
# Your branch is ahead of 'origin/master' by 2 commits.
#
nothing to commit (working directory clean)
```



Push example, continued

```
Cyndi@CYNDI-PC /c/csm_temp/help/GUI2 (master)
$ git push origin master
Username for 'https://github.com': CyndiRader
Password for 'https://CyndiRader@github.com':
Counting objects: 12, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 773 bytes, done.
Total 9 (delta 4), reused 0 (delta 0)
To https://github.com/CyndiRader/GUI2.git
    f275b73..1a325f8  master -> master
```

```
Cyndi@CYNDI-PC /c/csm_temp/help/GUI2 (master)
$ git status
# On branch master
nothing to commit (working directory clean)
```



Fetch example - On original machine

- Fetch, merge

```
$ git fetch
remote: Counting objects: 9, done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 9 (delta 4), reused 9 (delta 4)
Unpacking objects: 100% (9/9), done.
From https://github.com/CyndiRader/GUI2
   f275b73..1a325f8  master    -> origin/master
```

```
$ git merge origin/master
Updating f275b73..1a325f8
Fast-forward
 .gitignore      | 4 ++++
 src/GUIDemo.java | 2 +-
 2 files changed, 5 insertions(+), 1 deletion(-)
 create mode 100644 .gitignore
```

fast-forward covered under branching



Pull example – on original machine

- If you're careful to avoid conflicts, and you're working on the master branch, easier to just pull.
- Make some changes on original machine
- Don't forget to commit! (see push below, didn't work)

```
Cyndi@CYNDI-PC /c/csm_classes/cs306/JavaCode/Practice/GUI2 (master)
$ git status
# On branch master
# Changes not staged for commit:
#   (use "git add <file>..." to update what will be committed)
#   (use "git checkout -- <file>..." to discard changes in working directory)
#
#       modified:   src/GUIDemo.java
#
no changes added to commit (use "git add" and/or "git commit -a")

Cyndi@CYNDI-PC /c/csm_classes/cs306/JavaCode/Practice/GUI2 (master)
$ git push origin master
Username for 'https://github.com': CyndiRader
Password for 'https://CyndiRader@github.com':
Everything up-to-date
```



Pull continued

- Note typo. git merge origin/master BUT git push origin master.
- Merge is specifying the branch to merge (in this case master branch on remote named origin - branches covered in detail later).
- Push is specifying the remote and the local branch.

typo

```
$ git commit -am "Modified size of GUI window"
[master f87024d] Modified size of GUI window
1 file changed, 1 insertion(+), 1 deletion(-)

Cyndi@CYNDI-PC /c/csm_classes/cs306/JavaCode/Practice/GUI2 (master)
$ git push origin/master
fatal: 'origin/master' does not appear to be a git repository
fatal: The remote end hung up unexpectedly

Cyndi@CYNDI-PC /c/csm_classes/cs306/JavaCode/Practice/GUI2 (master)
$ git push origin master
Username for 'https://github.com': CyndiRader
Password for 'https://CyndiRader@github.com':
Counting objects: 7, done.
Delta compression using up to 4 threads.
Compressing objects: 100% (4/4), done.
Writing objects: 100% (4/4), 380 bytes, done.
Total 4 (delta 2), reused 0 (delta 0)
To https://github.com/CyndiRader/GUI2.git
1a325f8..f87024d master -> master
```

Pull continued – on clone machine

- git pull origin master
- git pull (assumes origin, master)

```
$ git pull
remote: Counting objects: 4, done.
remote: Compressing objects: 100% (2/2), done.
remote: Total 4 (delta 2), reused 4 (delta 2)
Unpacking objects: 100% (4/4), done.
From https://github.com/CyndiRader/GUI2
   1a325f8..f87024d  master    -> origin/master
Updating 1a325f8..f87024d
Fast-forward
 src/GUIDemo.java | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)
```

