Git and GitHub Exercise

In today's exercise, you'll work with two different repositories on GitHub to perform tasks that are commonly done by web developers.

In the first part of the exercise, you'll make a fork (your own copy) of a repository that you don't have write (push) permissions on, add a file to your copy of the repository, and then request that your new file be added into the the original repository. This is typically how you will add files to a repository that someone else controls. It gives the owner of the repository a chance to review and approve your changes before merging them into the codebase.

In the second part of the exercise, you will have write (push) access to a shared repository—which will is more likely to happen if you're in a small workgroup without a single person approving updates to the code. (This is likely to be the case when you're working on group projects here at RIT.) You'll be editing the same file that the rest of the class is editing, which means there's the potential for conflicts if you don't follow instructions!

Pre-Class Preparation

In week 1, I asked you to set up an account on GitHub (if you didn't already have one), and to apply for the GitHub Student Developer Pack so that you'll be able to have private repos. Today's exercise also assumes that you've completed sections 1-3 of the GitHub for Web Designers tutorial on Lynda.com that was assigned for today's class. If you haven't, please do that **before** attempting the exercise!

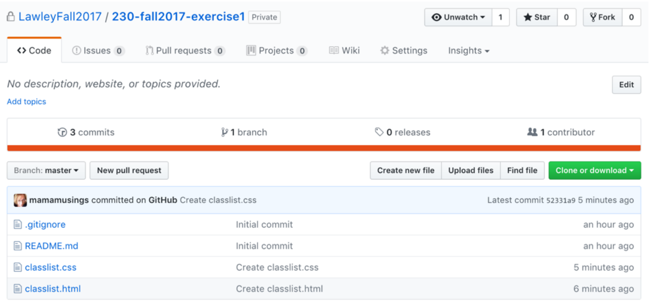
Because the files we're using for this exercise are in private repositories in the LawleyFall2017 organization, you need to be part of the 230-fall2017-students team in that organization to access them. If you posted your GitHub username to the #introductions channel in Slack in week 1, you should have been added to that team. If you haven't done that yet, you'll need to ask the professor or the TA to do that now. (When you're added, you should receive an email from GitHub with a link to join the team.)

The instructions for this exercise assume that you're using Visual Studio Code as your editor. If you're using Brackets or another editor, you'll need to install and teach yourself how to use any necessary extensions to provide Git support.

GIT AND GITHUB EXERCISE

Part 1: Contributing to a Repository When You Don't Have Write (Push) Access

1. Go to https://github.com/LawleyFall2017/230-fall2017-exercise1 .
   You should see four files: an HTML file called classlist.html, a CSS file called GitHub-exercise.css, a .gitignore file that's used to tell your client what types of files not to sync to the server, and a README file that explains the purpose of the repository.



No description, website, or topics provided.

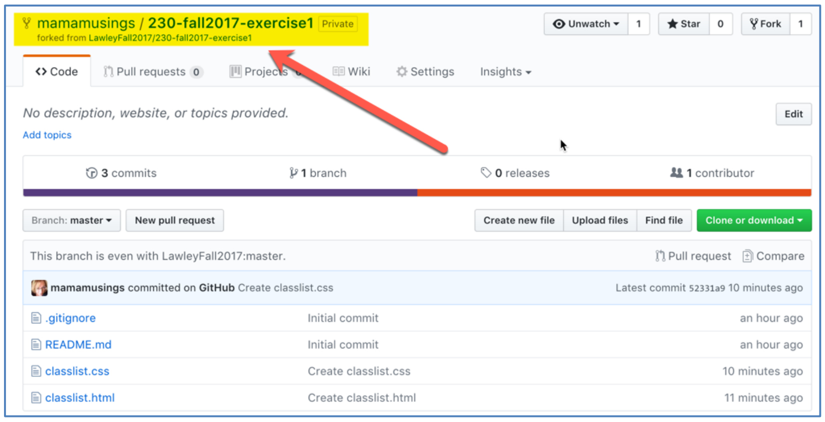
mamamusings committed on GitHub Create classlist.css — Latest commit 52331a9 5 minutes ago

.gitignore	Initial commit	an hour ago
README.md	Initial commit	an hour ago
classlist.css	Create classlist.css	5 minutes ago
classlist.html	Create classlist.html	6 minutes ago

2. You are going to to create a "fork" of the repository on GitHub. A fork is a new repository in your own GitHub account that is a copy of an existing repository. A common reason to make a fork is to take a project that someone else owns in a new direction. For instance, a faculty member at another university might want to use my course repository on GitHub as a starting point for their own class. They could fork the class repository so that they have the basic structure and content, and then edit the files to reflect their own class details. The process of forking leaves a pointer back to the original, which is a helpful way to let people see where content originates, and how it has changed.

   A fork can also be used if someone who doesn't have access to modify a repository wants to make changes—for instance, a programmer who wants to help fix a bug on a big open source project. In that case they would make a fork of the code, fix a bug or add a feature, and then send a request to the original owner to pull in their changes. This is referred to as a pull request.

   In the top right corner of the page, click the button labeled "Fork" (if prompted, select your personal account). This will create an exact copy of the exercise1 repository in your account. If your GitHub Education Pack was approved, your repository should also be set to private. You should be taken to the new repository page. While the content is the same you can see that the path to the repository now has your user name rather than LawleyFall2017.



No description, website, or topics provided.

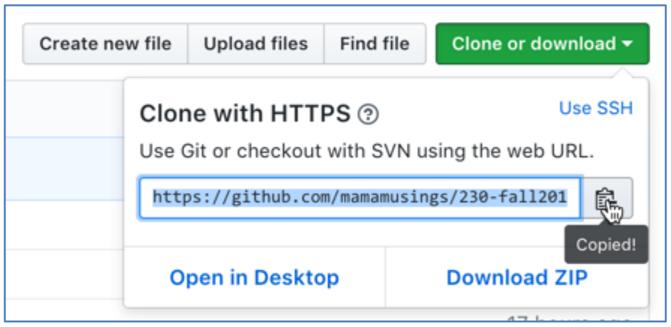
This branch is even with LawleyFall2017:master.

mamamusings committed on GitHub Create classlist.css — Latest commit 52331a9 10 minutes ago

.gitignore	Initial commit	an hour ago
README.md	Initial commit	an hour ago
classlist.css	Create classlist.css	10 minutes ago
classlist.html	Create classlist.html	11 minutes ago

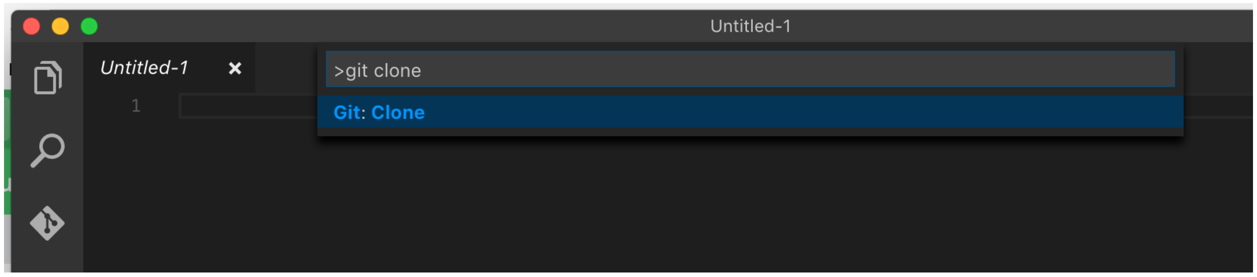
Git and GitHub Exercise

3. Now you're going to clone your GitHub repository to your local machine, so you can edit the files locally. Copy the CLONE URL from your new repository (the one under your account, not the one in LawleyFall2017), using the "Clone or Download" button. If you hit the little clipboard button next to the URL, it will copy the entire URL to your clipboard for pasting.



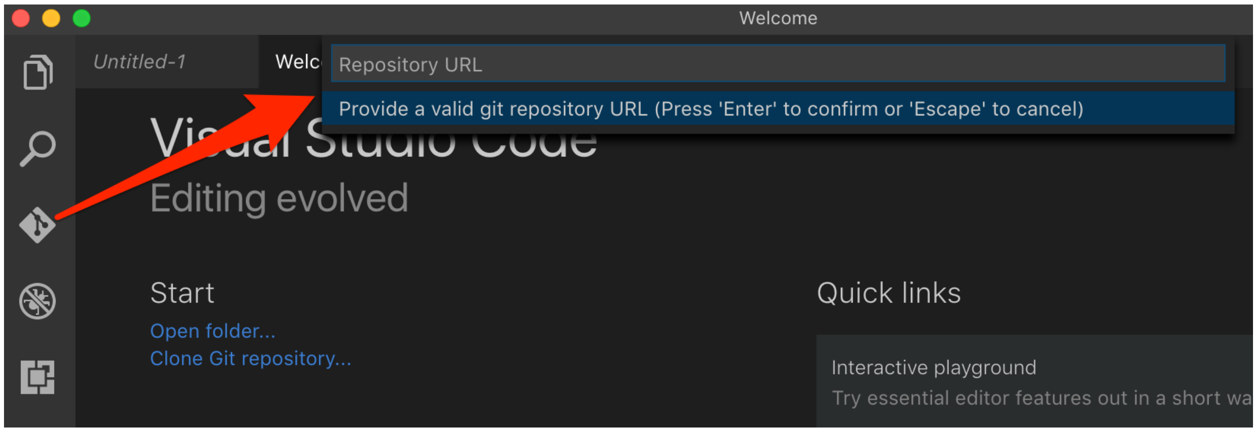
Clone with HTTPS
Use Git or checkout with SVN using the web URL.
https://github.com/mamamusings/230-fall201

4. Launch Visual Studio Code (or, if it's already open, choose "New Window" from the File menu). Choose "Clone Git Repository..." from the welcome screen. (Or press F1 for the command menu, and type Git Clone. You should see the Git: Clone command. Press enter.)



```
>git clone
Git: Clone
```

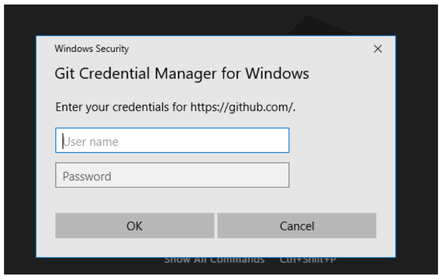
Visual Studio Code will now prompt you for a valid Git repository URL, which is what you just copied from GitHub. Paste the URL in here.



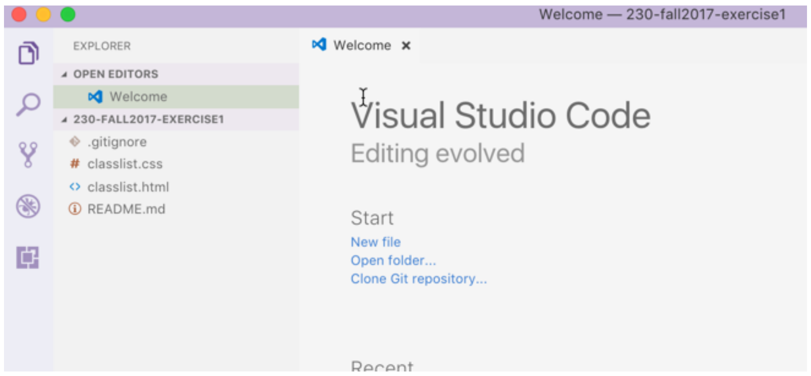
Repository URL
Provide a valid git repository URL (Press 'Enter' to confirm or 'Escape' to cancel)

Hit enter once you've pasted in the URL, and VS Code will prompt you for a parent directory. It will create a new folder with the repository name in the folder you designate. If you don't know how to specify a directory path, it's easiest to just take the default, and then move the folder later if necessary.

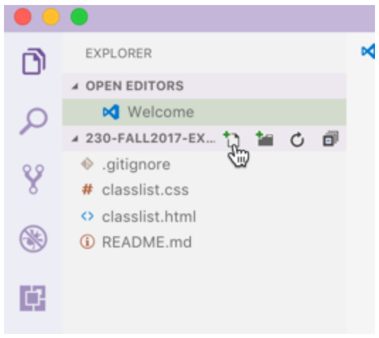
Since you created a private repository, VS Code will then prompt you for your GitHub username and password, so it can access your files. (If you've enabled 2-factor authentication on GitHub, this step may not work properly, and you might have to authenticate using the command line, as explained in the assigned GitHub tutorial. Ask me for help with this.)

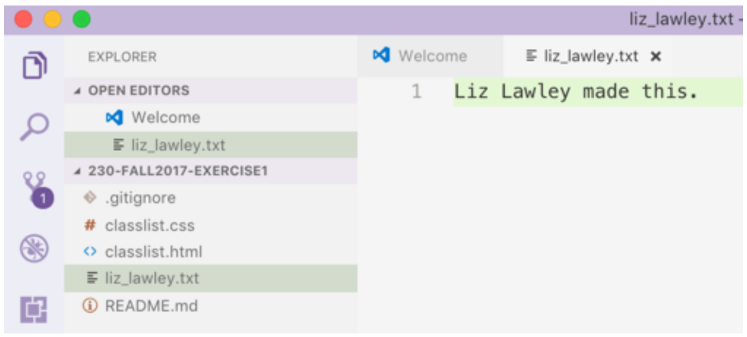

Windows Security
Git Credential Manager for Windows
Enter your credentials for https://github.com/.

If this all works properly, you'll now have an exact copy of your GitHub repo on your local computer, and displayed in VS Code. You'll be asked if you want to open the new repository—say yes. The files from the repository will be displayed in the explorer bar on the left side.

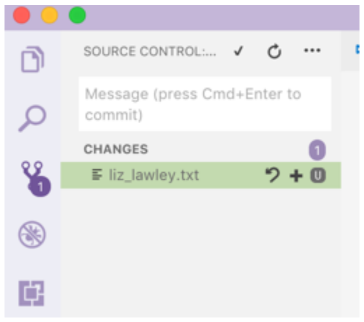


5. Hover over the directory name (230-fall2017-exercise1) in the left sidebar, and click on the "New File" button. Name the file firstlast.txt, where first=your first name, and last=your last name. Add some text to the file, and save it.




```
Liz Lawley made this.
```

6. You should now see a notification on the Git icon in the left sidebar. Click on that icon to see the current status of your repository.



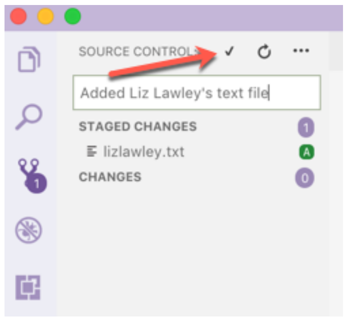
You should see the file you just created listed under "Changes", with a U (for "Untracked") next to the file. The file has been added on your computer, but it has not yet been recorded in your local Git repository. The local repository is the data structure on your machine tracking the history and changes of the files within this folder. You need to explicitly tell git, on your computer, that you want it to keep track of the new file.

The first step is to *stage* this file, which you can do by clicking on the + sign to the right of the file. When you do this, the file will move from "Changes" to a new category of "Staged Changes." (And the "U" for untracked should change to "A" to reflect that the file will be added to the repository.) Stage changes have not been added to the repository yet—that won't happen until you commit the changes.

Why the extra step? When adding a new feature to your code on a collaborative project, it's possible that you will be altering multiple files. (For instance, you might update an HTML file and the CSS file that accompanies it.) You want to make sure that all the related files are updated at once, in case someone else is also working on some of those files, and so that all of your dependencies work.

To address this issue, git waits until you have queued all of the files you want to add in a single update, and then adds them all at once when you commit the files. When someone else syncs their code with the repository, they will get all of the files from your update.

7. We're only adding this one file right now, so it's time to commit it to the repository.

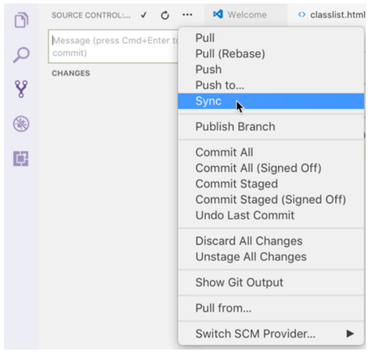


Type your commit message in the box above "Staged Changes." The message should be both brief *and* useful. They should allow other repo users to see at a glance what you changed.

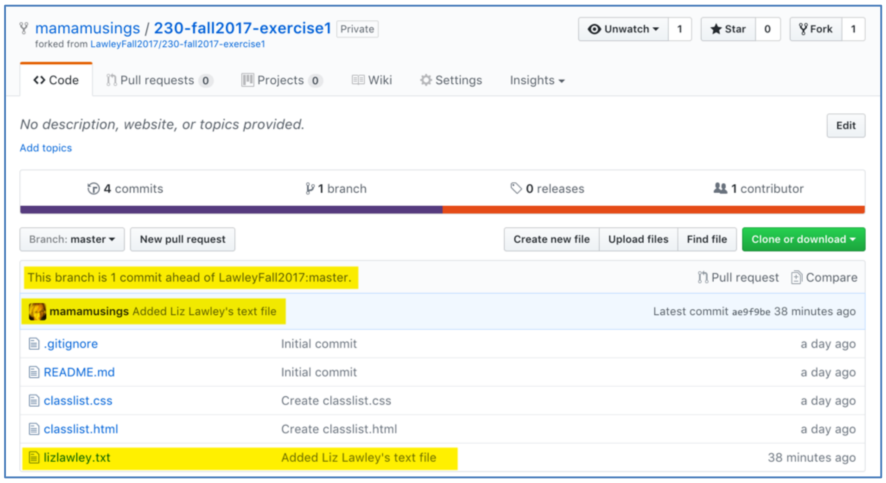
Once you've added your commit message, click the check mark to commit the files to your local copy of the repository.

8. Your changed files are now part of the local copy of the repository on your computer, but they are not yet synced to any other version of the repository. Next you need to sync your changes with the GitHub version of your repository.

   You can see the various options for interacting with the GitHub version of your repository by clicking on the … in the top right corner of the source control sidebar. You can either use "Push" (which pushes your changed files to the GitHub repo), or "Sync" (which pulls the current version of the repo and then pushes your changes). It's good practice to always pull the current version of a repo before you push, so choose Sync. VS Code will send the changes from your commit to the remote repo.



9. Look at your copy of the repository on GitHub (you may need to refresh the page). At the top of the list of files you should see your most recent commit message, and that message will also show up next to the classlist.html file you modified. There will also be a message above the latest commit saying "This branch is 1 commit ahead of LawleyClasses:master", because GitHub keeps track of the differences between the original repository (the one in LawleyFall2017) and your forked copy.



No description, website, or topics provided.

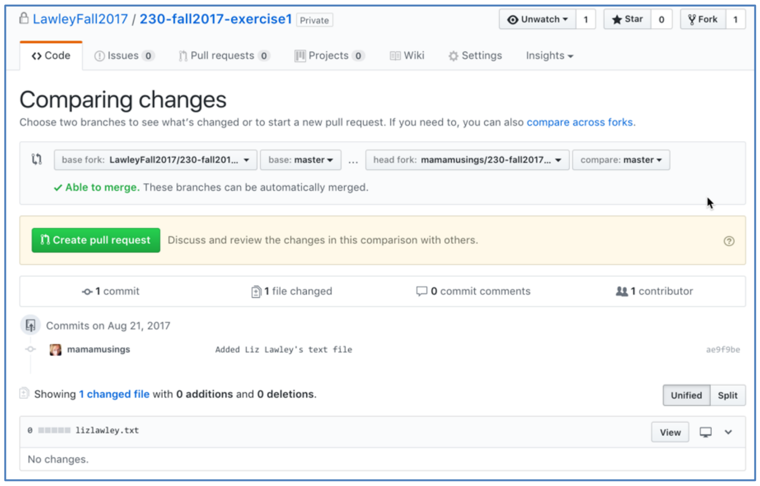
This branch is 1 commit ahead of LawleyFall2017:master.

mamamusings Added Liz Lawley's text file — Latest commit ae9f9be 38 minutes ago

.gitignore	Initial commit	a day ago
README.md	Initial commit	a day ago
classlist.css	Create classlist.css	a day ago
classlist.html	Create classlist.html	a day ago
lizlawley.txt	Added Liz Lawley's text file	38 minutes ago

10. Now go back to the LawleyFall2017 version of the repository (there's a link at the top of the repo saying "Forked from…") Notice that it has not been changed. This is because your "fork" of the repository is entirely separate from the original. Since you don't have permission to write to this repository, you can't push your changes. The only way you could change it is by submitting a pull request, which is what we'll do next.

11. Go back to your personal copy of the repository. Notice that above the line saying "This branch is one commit ahead of LawleyFall2017-master," there's a button that says "Create pull request". Click that, and you'll be taken back to the original repository, with a screen showing whether your proposed changes conflict with any changes made by others. Since you're adding a new file, rather than editing an existing file, there shouldn't have been any conflicts. Now click the green button to create a pull request.



LawleyFall2017 / 230-fall2017-exercise1 Private

Comparing changes

Choose two branches to see what's changed or to start a new pull request. If you need to, you can also compare across forks.

base fork: LawleyFall2017/230-fall201… base: master … head fork: mamamusings/230-fall2017… compare: master

✓ Able to merge. These branches can be automatically merged.

Create pull request Discuss and review the changes in this comparison with others.

1 commit 1 file changed 0 commit comments 1 contributor

Commits on Aug 21, 2017

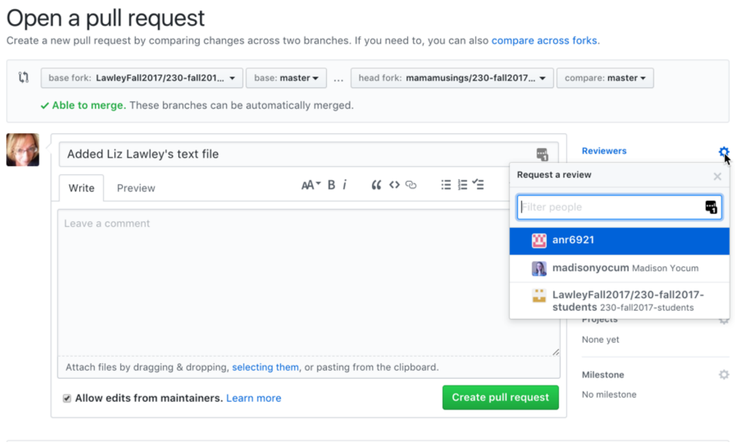
mamamusings Added Liz Lawley's text file ae9f9be

Showing 1 changed file with 0 additions and 0 deletions.

0 lizlawley.txt

No changes.

12. On the screen that follows, you can leave the title as your commit message ("Added Your Name's text file"). Add your TA (madisonyocum for the New Media section, anr6921 for the GDD section) as a reviewer. Then click the green button at the bottom to send the request.



Open a pull request

Create a new pull request by comparing changes across two branches. If you need to, you can also compare across forks.

base fork: LawleyFall2017/230-fall201… base: master … head fork: mamamusings/230-fall2017… compare: master

✓ Able to merge. These branches can be automatically merged.

Added Liz Lawley's text file

Leave a comment

Attach files by dragging & dropping, selecting them, or pasting from the clipboard.

✓ Allow edits from maintainers. Learn more

Create pull request

Reviewers

Request a review

anr6921

madisonyocum Madison Yocum

LawleyFall2017/230-fall2017-students 230-fall2017-students

None yet

Milestone

No milestone

13. Your pull request has now been submitted for approval by the repository administrator(s). Your file will not appear in the repository until it's been reviewed and approved. Once you've submitted the pull request, you can move on to the second part of this exercise.

GIT AND GITHUB EXERCISE

Using Git Part II: Modifying Files in a Shared Repository

1. For this part of the exercise, you'll be using the https://github.com/LawleyFall2017/230-fall2017-exercise2 repository. It is very similar to the first repository—with the exception that the HTML file is called "sharedclasslist.html", and (more importantly) that *everyone in the the class has the ability to directly modify the files*.

2. Instead of making a fork of the repository in your own GitHub account, you're going to clone the shared LawleyFall2017 repository to your computer. Using the process described above in step 3 of Part I, copy the clone URL for the shared repository, and clone it to your local computer in VS Code.

3. Open the sharedclasslist.html file in VS Code, find your name, and make it a link to your igm230 directory on people.rit.edu.

4. Once you've added the link, stage and commit the sharedclasslist.html (and *only* that file!) to your local repository. Make sure your commit message includes your name (e.g. "Added link to Liz Lawley's page")

5. Use the "Sync" command to pull the most recent copy of the file (which may have other student's links already added), and then push your changes. If you've only modified the line with your name, there shouldn't be any conflicts. If you do get a conflict, ask for help before moving on!

6. Once you've successfully synced your addition to the document, take a look at the repository on GitHub, and view the sharedclasslist.html file. You should see your change reflected in the file.

Grading

There are four points for this exercise:

- 1 point for successfully submitting a pull request to the exercise1 repo
- 1 point for the pull request including the correct content
- 1 point for successfully committing changes to the sharedclasslist.html file in the exercise2 repo
- 1 point for properly including a link from your name to your igm230 page on people.rit.edu in sharedclasslist.html