Find the Project on GitHub

If you haven't already, go to github.com and sign up for an account.

Next, visit GitHub Classroom to clone the class repository here:

Create a Fork of the Class Repository

At the top right of your screen, you should see a **Fork** button. Click on this, and select your personal account as the destination of this "fork".

A Fork is a copy of the original GitHub Class Repository that is tied to the original, but independent. This is your private area within GitHub to work on assignments and submit your work. You are free to work on this fork, and make changes.

To start writing code and modifying your repository, you need to clone your GitHub (hereafter referred to as "remote") repository onto your local machine.

Clone Your GitHub Repository To Your Local Machine

Copy the clone URL of **your repository** from your browser and issue the following command from your EECS Unix account.

```
$ git clone <URL>
```

Note: you can Copy the <URL> from your own fork of the class repository page by clicking the "**Clone or download**" box on the right side of the page.

For example, **my command** might look like.

```
$ git clone https://github.com/aburasa/CS362-001-SP17.git
```

My github username is **aburasa**, and your <URL> should contain your github username.

This will clone the main class repository, creating a copy of both the repository and the individual files in a subdirectory based on the GitHub URL.

You will find a directory on your local machine with the class repository name (i.e., CS362-001-SP17)

Working on the Repository

You can make changes to the remote repository as needed. You can add files, create directories, as you want.

Now, **cd** into the local directory

```
$ cd CS362-001-SP17
```

```
$ ls
```

projects README.md

```
$ cd projects
```

you should be able to see the instructor directory aburasa. You need to create your directory here.

```
$ mkdir youronid
```

```
$ cd youronid
```

```
$ vim README.md
```
#Add README.md file that contains a single line with your name in youronid directory.

Under youronid folder, create your own assignment-1 folder. Look at instructor's folder hierarchy. Add README.md file in the assignment-1 folder and the README file contains "This is my assignment-1 Folder".

To commit youronid, assignment-1 folders and README.md file to a local repository, we must explicitly add them with.

Now go back to projects folder.

```
$cd ..
```

and type this

```
$ git add .
```

```
$ git status
```

Committing the Changes to the Local Repository

When you get to the point where you want to stop working for a while, commit the change to the local repository. Each time you commit, it is important to describe the changes you make.

```
$ git commit -m "Added youronid and assignment-1 folders and README.md with my name"
```

Keep in mind that any changes that made are not part of the local repository until you `commit`

Pushing the to origin (GitHub)

Refresh your GitHub project page and notice that your change hasn't appeared on the website. While the local repository is now updated, it is necessary to push those changes to the GitHub repository

```
$ git push origin master
```

Now go and check **your fork of the class git repository**, your folder should be created under projects folder, just like projects/aburasa, and it contains README.file that has your name. Keep in mind that any commits to a local repository are not made in a remote repository until you commit

Branching

When you are ready to submit your code, you will create a new branch called youronid-assignment-X, where youronid is your onid and X is the number of the assignment. The instructor and TAs will pull from this branch on GitHub and your submission date will be the timestamp of the last commit on that branch. **Do not commit to this branch after the deadline!**

The walkthrough below should be similar for future assignments:

First, check the current branch. We currently only have one branch called master, this is the default branch.

```
$ git branch
* master
```

The `*` indicates the current branch on the filesystem. Let's add a new branch and start working on youronid-assignment-X.

```
$ git branch aburasa-assignment-1
$ git branch
  aburasa-assignment-1
* master
```

Note that `branch` does not switch the current branch, so any edits you make after its creation are applied to the master branch. The `checkout` command is used to switch branches

```
$ git checkout aburasa-assignment-1
```

```
$ git branch
* aburasa-assignment-1
  master
```

Now, edit the projects/youronid/assignment-1/README.md file to read "This is my assignment-1 submission!" and then commit the change.

```
$ git add README.md
```

```
$ git commit -m "Changed assignment 1 readme"
```

To submit the branch to github

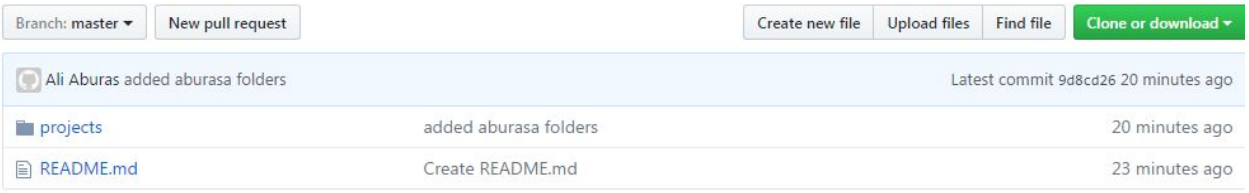
`$ git push origin aburasa-assignment-1` # Change aburasa to your onid

Refresh your GitHub page - in the upper left hand corner, click the **branch** dropdown: you should see a new `youronid-assignment-1` branch - this is your submission - you may want to double check the submitted files.

Switch back and forth between the `youronid-assignment-1` and `master` branches on your Github project page to ensure the projects/youronid/assignment-1/README.md is different.

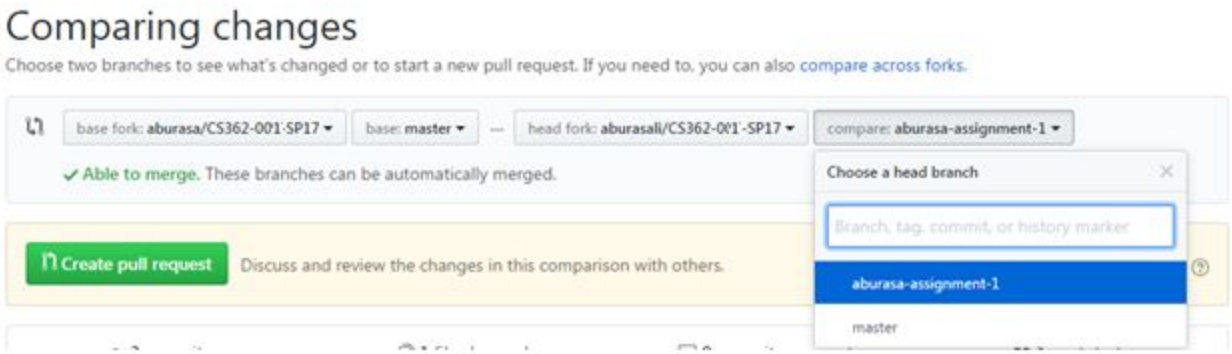
Notifying the Instructor of Changes

When you are finished with your work, you will send the instructor/TAs a “Pull Request” informing them you have changes you have made to your “fork and branch”, and you want them to review those changes.

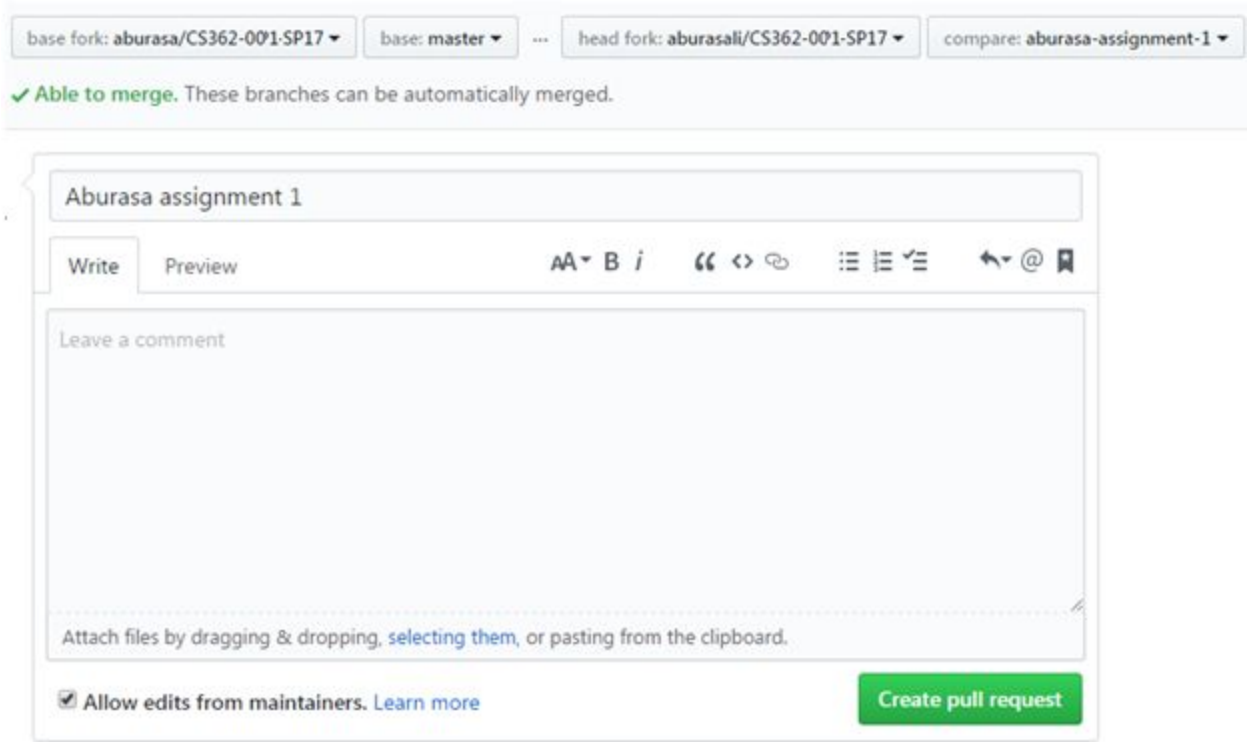


Open up your GitHub page and navigate to your forked project page. On the left side of this page, you should see a button “**New pull request**”.

Click on this button, and you should see a message box indicating you have made changes.



Choose the branch you want to create a pull request for (i.e., aburasa-assignment-1) . Then, click on “**Create pull request**” and you will see a box where you can add comments. Give your pull request a title and write a brief description of your changes.



When you submit this form, the instructor will get that message and can reply or grade the work.

Click on “Create Pull Request” when you are done.

Get the latest folders and files from the class repository by running

```
git remote add upstream "https://github.com/aburasa/CS362-001-SP17.git"
git checkout master              # make sure we're on the master branch
git fetch upstream             # pull any information about changes in upstream
git merge upstream/master -m "Sync" # merge new files
```

You should now have new folders and files in your `CS362-001-SP17` directory