GitHub Classroom

This semester we will be using private repositories in GitHub Classroom to store student source code. To use GitHub Classroom, you must have a GitHub account. This note lists all the steps to set up a GitHub account and the additional steps to access the CSCI571 Google Classroom repositories.

1. Sign up for GitHub

Many students that join CSCI571 already have set up a personal GitHub account. The standard repositories of a “free” GitHub account are “public” and therefore cannot be used to store “private” source code. Nicely, GitHub Classroom allow the CSCI571 instructors to set up “private” repositories accessible only by the individual students and their instructor(s) (Professor, TA and Producers).

If you already have a GitHub account, you can jump to section 2, “Sign up for GitHub Classroom”. Otherwise, we actually recommend you sign up for a “new” GitHub account, by going to:

https://github.com

Click Sign up in to top right of the page, and the following dialog is displayed.

Pick a username, enter your username@usc.edu as the Email address, and create a Password. Then click on Sign up for GitHub. The Welcome to GitHub page is
displayed.

In **Step 2: Choose your plan** leave the **personal plan** selection to “**unlimited public**” and click on **Continue**.
In Step 3: Tailor your experience, select School projects and click on Submit. You will be redirected to the GitHub dashboard.

Feel free to Read the guide and Explore GitHub. You should be receiving an e-mail asking you to verify your e-mail address.

Click on the Verify email address hyperlink. Once you e-mail address is verified, you will receive an additional Welcome e-mail. If you get this, your GitHub account is
2. Sign up for GitHub Classroom

Before signing up for GitHub Classroom, you must associate your USC e-mail address with your GitHub account. If you have followed the instructions above, this is already done. If you already had a GitHub account associated with a different e-mail address, you need to add the USC email address to your Personal Settings. In the GitHub dashboard, click in the dropdown avatar, and select Settings. The Personal Settings page is displayed.
Click the Email tab, enter your USC email address and click Add. Next you should make your USC email address the primary email address. Then Sign out from GitHub.

If you sign up for GitHub for the first time, or you already had a GitHub account, you can now sign up for GitHub Classroom.

To sign up for GitHub Classroom, you need to use an invitation link provided by your instructor. There will be one invitation link for each of the graded exercises, starting with Homework #3.

The invitation links will be provided in Piazza and will look like this:

https://classroom.github.com/a/YHJ3vhk_

Paste the URL of the invitation link in your browser. You will be asked to Sign in to GitHub to continue to GitHub Classroom. Enter your username and password, and click Sign in.
You will now need to **Authorize GitHub Classroom** to access your GitHub account.

Click on **Authorize github**. You will be connected to the **Web Technologies – Fall 2019** GitHub Classroom page, where you will be asked to “Join the classroom roster”.
Select your USC e-mail address by clicking on it. The **Accept … assignment** page is displayed.

Click on **Accept this assignment**. The **Accepted … assignment** page is displayed.

You are now told that you are **ready to go**. The page also warns you that you may receive an e-mail invitation, such as the one shown below.
You can disregard this invitation as you have already accepted. You may also receive an additional e-mail indicating the "a third-party OAuth application has been added to your account. This is just fine.

In the **Accepted … assignment** page shown above, you will find a link where your assignment has been created, as this sample link:

https://github.com/usc-csci571/homework1-roadster2020

Each homework assignment repository will have this URL format:

https://github.com/usc-csci571/

followed by homeworkx- with “x” being the homework number, such as 3, 4, 6, 8, and 9, followed by your GitHub username.

If you click on the URL, you will be taken to the “**Code**” page of your assignment repository. This is the repository where you will store your individual assignment source code. This code will be tested by the MOSS tool for plagiarism discovery.

The repository Code page looks as follows:
The instructions on the **Code** page list the various ways “create” “push” your code from a “local” git repository (local to your laptop or desktop computer), to the GitHub Classroom repository. To be able to do this, you will first have to install Git to your local machine.

### 3. Install and use Git

Instructions to download Git are available here:

[https://git-scm.com/downloads](https://git-scm.com/downloads)

Documentation on Git is available here:

[https://git-scm.com/doc](https://git-scm.com/doc)

There are numerous tutorials online on how to install Git. These are my favorites.
Atlassian’s BitBucket “Install Git” tutorial with step-by-step instruction on how to install Git on a MacOS, Windows and Linux platforms:

https://www.atlassian.com/git/tutorials/install-git

Linode’s “How to Install Git on Mac” tutorial, with instruction on how to install Git with Homebrew, MacPorts, and the Package Installer:

https://www.linode.com/docs/development/version-control/how-to-install-git-mac/

After a successful Git install, you can just follow the create and push instructions in the Code page mentioned above. Here is an example, showing the output you will see on your screen. Notice in **bold** the CLI commands:

```
$ echo "# homework-1-roadster2020" >> README.md

$ git init
Initialized empty Git repository in /Users/marcopapa/Desktop/GitHub571/.git/

$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory.

$ git commit -m "first commit"
[master (root-commit) ff55c43] first commit
 1 file changed, 1 insertion(+)
  create mode 100644 README.md

$ git remote add origin https://github.com/usc-csci571/homerwork-1-roadster2020.git

$ git push -u origin master
Username for 'https://github.com': roadster2020
Password for 'https://roadster2020@github.com':   (enter password or token when using 2FA)
Counting objects: 3, done.
Writing objects: 100% (3/3), 237 bytes | 237.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/usc-csci571/homerwork-1-roadster2020.git
 * [new branch]      master -> master
Branch master set up to track remote branch master from origin.
```

Whenever you add a new file to the local repository you need to add all the above commands, except the first 2 commands, and changing the name of “first commit” to “second commit”, and so on.
Besides the CLI Git commands, a very nice desktop application is provided named GitHub Desktop. GitHub Desktop is available here:

https://desktop.github.com/

GitHub Desktop is available for macOS and Windows. Documentation is available here:

https://help.github.com/desktop/guides/

After installation, the GitHub Desktop app interface looks like this:
The usual steps to follow in the GitHub Desktop app when making changes are as follows:

1. Make changes to local repository (change or add files)
2. Edit summary and Description
3. Commit to Master
4. Push origin

Happy Git.