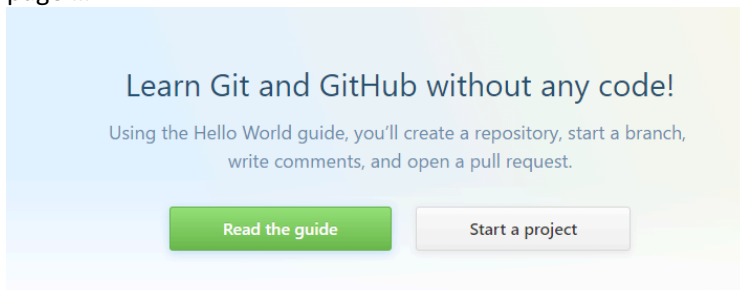# CSC 174 Lab 4 – Github Setup

Due: Tuesday, September 27, 2016

Create a Github Account

- ❑ Go to **github.com** and Sign up for Github
- ❑ After you go through the sign-up process and you get to the Read the Guide / Start a Project page …

Learn Git and GitHub without any code!

Using the Hello World guide, you'll create a repository, start a branch, write comments, and open a pull request.

Read the guide   Start a project

…you can simply **close** that window/tab[1]
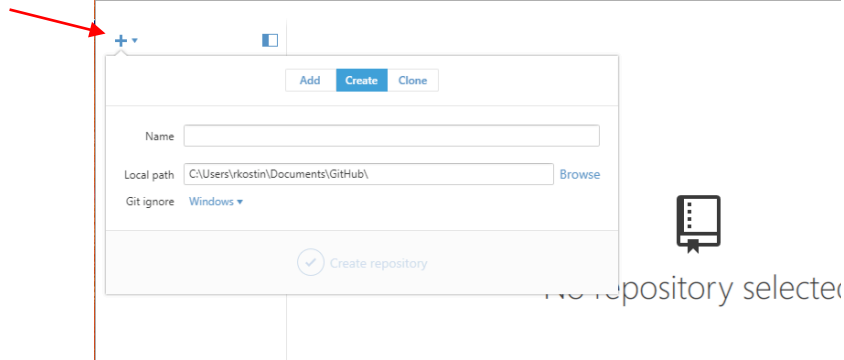
Install the Desktop Client Software

- ❑ Go to **desktop.github.com** and download/install the Github Desktop application
  - Mac users: just drag the .app file to your Applications folder and run it
  - Windows users: run the installation file
- ❑ Start the application (if not already running) and **login** using the credentials you created in the previous step
  - You can skip the step to find local repositories

---

[1] However, the **Read the Guide** tutorial is very good. If you are interested in a career in development, it might behoove you to spend some time learning more about Git and Github than we're going to cover in CSC 174.

Create a New Repository

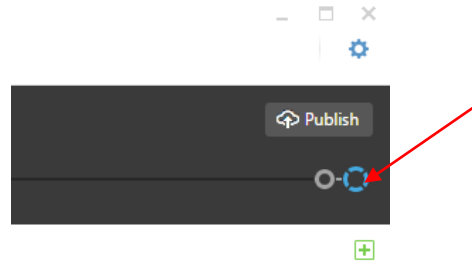❑ Click the + (top left corner) to create a new repository

❑ In the dialog box that appears, create a name - it can be anything, like **csc174-test** or something like that
- CHANGE the "Local path" to point to the location of your local host (WAMP or MAMP server), could be…
  Windows: **c:\wamp\www**
  Mac: **//Users/Applications/MAMP/htdocs**
  …or something else
- Then click the Create Repository button

Create Some Content

❑ Find the new folder in your file system and open it.

❑ In the new folder, create a plain-text file - any file
- Try something simple like a text shopping list or something like that
- Save the file

CSC 174 Lab 4: Github Setup

Commit/Publish Your Repository to Github

❑ Toggle back to the Github Desktop application - make sure you're looking at the "head" of your repository
- Make sure this is clicked (near the top-right)…
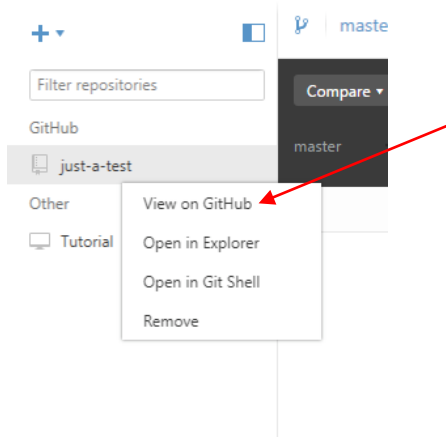

There, you'll see a view of your file

❑ Where it says "Summary" enter the word **Initial**[2]
- (Optional) Where it says "Description" enter some words, like "my shopping list" …or whatever

❑ Click the **Commit** button

❑ Find the **Publish** (or Sync) button (near to the top-right) and click it
- The first time you do this part of the process you'll get a new dialog box asking you to name and describe this "project" - just add a description, like "just testing" …or whatever, then click the Publish button

---

[2] We developers always call our first commit: "Initial"

CSC 174 Lab 4: Github Setup

Find your Project in Github

❑ In the Github Desktop application, right mouse click on the name of your repository in the list on the left side, then select: "View on GitHub"

There (in Github, on the Web) you'll see your Project file(s)

Share with Collaborators (your team members)

Note: I've decided to make this a gradable lab. There's a step in this section you MUST do to make sure you get credit for this lab. (prof k)

❑ While in Github, looking at your project, click the **Settings** menu item (near the top-right)

❑ On the Settings page, click the **Collaborators** tab (near the left side)

❑ On the Collaborators page, you have to find your team members and **Add** them

❑ Also on the Collaborators page, add **rkostin2** (not rkostin) to get credit for this lab

Join Your Team Members' Projects

❑ Check your email for invitations (or use the other tools Github provides) to **Accept Invitation**

❑ While looking at your team member's project in Github (on the Web) click the **Clone or Download** button
  o In the pop-up that appears, select **Open in Desktop**
  o You may get a confirmation dialog box on your computer - go ahead and do it
  o Choose a location - should be inside your localhost (web root) on your computer
  o Click the **Clone** button

Find and Edit Your Team Members' Files

- ❑ Look for your team members' files in your web root on your computer.

- ❑ Open their files - make edits - save your changes

Commit and Publish Your Edits

- ❑ Back in the Github Desktop application, find your team's projects
  - o Make sure you're looking at the "head" of each, where you'll see *your* edits highlighted in red or green

- ❑ For each project, write a summary (and description) and click the **Commit to master** button

- ❑ For each project, click the **Sync** button (near the top-right)

Get the Edits *from* Your Team Members

- ❑ After your team members have made changes to *your* project, in the Github Desktop application, select your project and click the **Sync** button (near the top-right)
  - o You should see the edits your team members made to your file(s)
  - o IF you end up with a "conflict" you need to go into your file system, find the file(s) with the conflict(s) and manually edit (and clean-up the markers)

Remember: Github is only one type of Git-type solution. There are others. But if you don't have a preference Github works nicely (enough).

When you start your Team Project, one of you will create a new repository and add the rest of the team as collaborators. Then the team will "clone" the shared repository so you can all work on the same files together.