Git / Github setup for MATH 342W

In the software engineering and data science worlds, git is the standard source control program thus it’s worth it to learn now. Further, you’ll be submitting your labs and theory homeworks with git and github.

# MAC Instructions

The first step is to make sure you have git on your computer. It comes preinstalled on most flavors on Linux. For the MAC, it is usually preinstalled. Try by running

git --version

in the terminal. If you see “git version <number>”, then it’s installed. If you do not see this, you can install git by opening the terminal and running

brew install git

If you don’t have brew installed yet, you first install brew by following the instructions at <https://brew.sh> then return and do the above. Verify using git –version that git is successfully installed.

Now open up the default terminal app.

# Windows Instructions

Windows does not have git preinstalled. You should download git for windows at <https://git-scm.com/download/win>. The recommended installer is the “standalone installer” and “64-bit Git for Windows Setup”. Once that’s installed, you should open up the “Git Bash” program which will provide you a terminal (the regular CMD terminal does not work with git)!

# Linux Instructions

You already have everything you need. Just open a terminal!

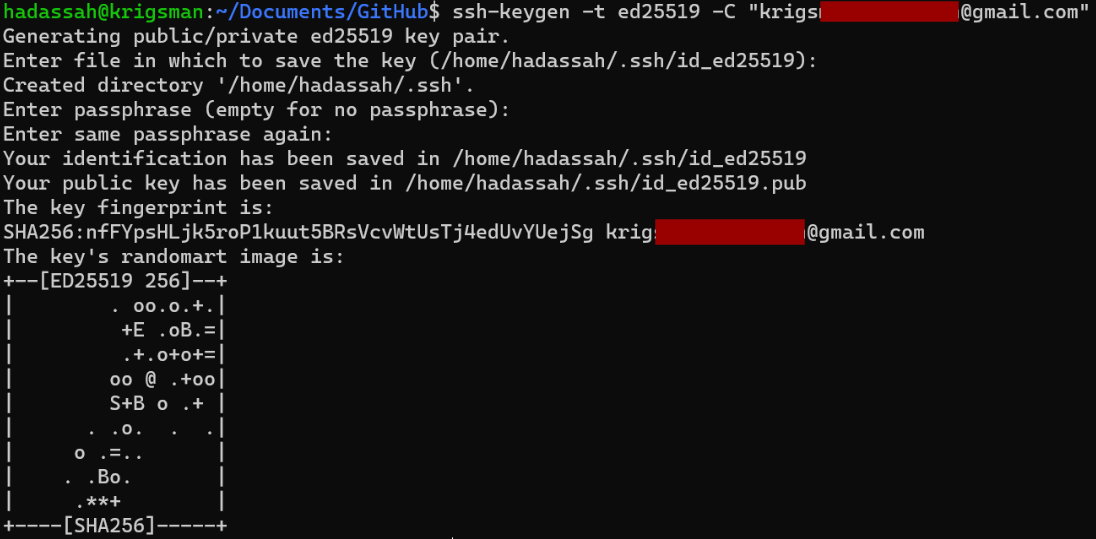
# All Operating Systems Continue Here

In the terminal, navigate to the folder you want to store your class materials using the “cd” command. If you are okay keeping your class materials in your default home folder, then do nothing.

We now setup your own security keys. This can be done by typing in the following command

ssh-keygen -t ed25519 -C "type your email address here"

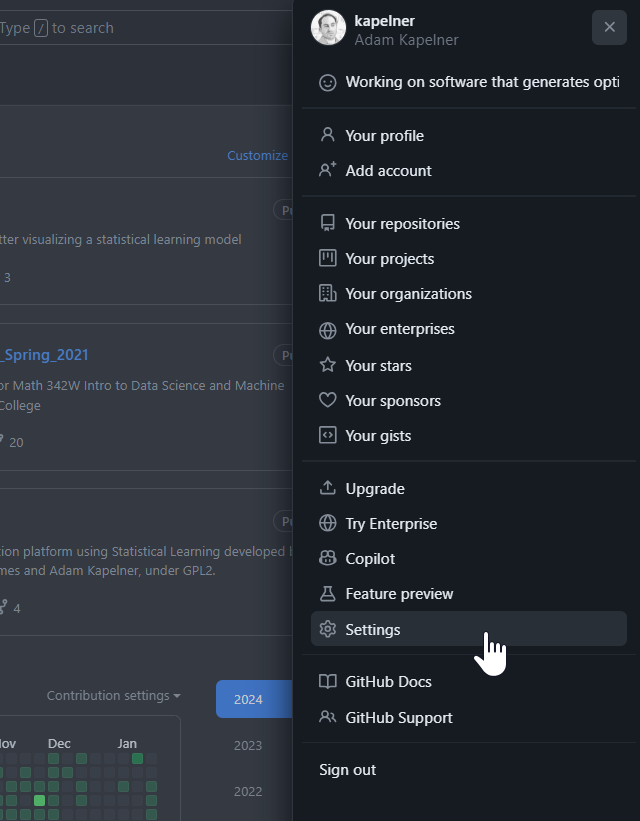
I use no password so I just press enter three times now. This is more convenient, but less secure as if someone gains access to your laptop, they can impersonate your github account and whatever else uses these keys. If the key creation is successful, you should see something like the following message:



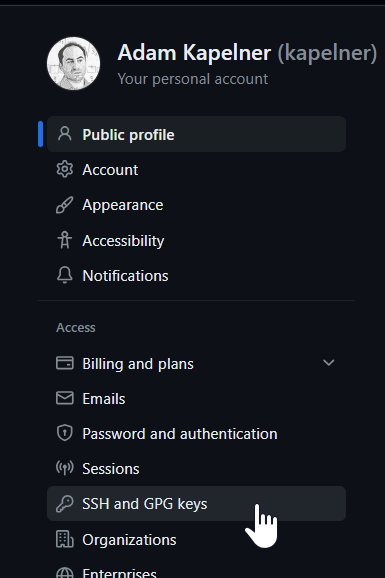
If you don’t already have a github.com account, go to github.com and create one. Use a username that’s appropriate! Employers will see it! Then login to github.com and keep that browser open for the rest of the setup.

Now, send me an direct message on discord with (1) your full name (2) your CUNY ID) and (3) your github homepage link (e.g. mine is <https://github.com/kapelner>). I need this to be message because I don’t know what your handle name to full name is and thus I won’t be able to grade your assignments!

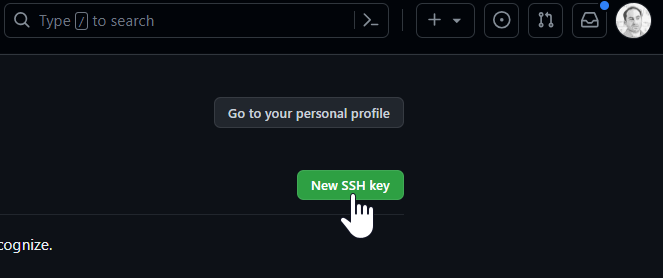
We then add your personal security key to github. You can do this by clicking on your account icon in the top right corner of your account homepage and clicking settings:



Afterwards you’ll click on the “SSH and GPG keys” on the left menu:



Now click the “New SSH key” in the top right



The title is up to you. The key belongs to a computer, so if you have multiple computers, you may want to name this key with something that identifies the computer. I use “my\_lab\_laptop”.

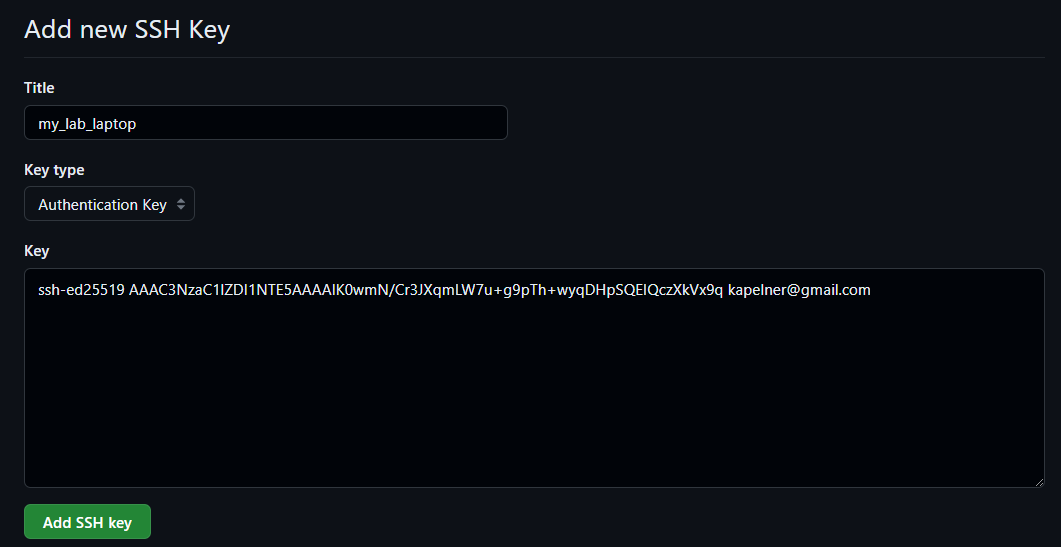
Now you need to open your public key that you generated. I use the following

vi ~/.ssh/id\_ed25519.pub

You should see something like this

ssh-ed25519 AAAC3NzaC1lZDI1NTE5AAAAIK0wmN/Cr3JXqmLW7u+g9pTh+wyqDHpSQEIQczXkVx9q kapelner@gmail.com

Now you need to copy that text ***exactly*** and paste it into the “key” field so the final screen should look like this:

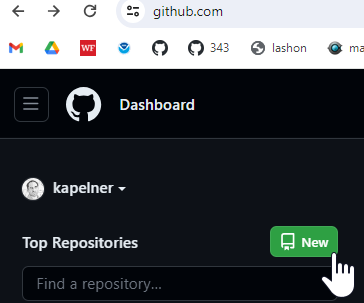
****

Now press “Add SSH key”.

Github is now set up to communicate with the laptop you’re using to follow this tutorial. If you have multiple computers you want to use for this class, you need to repeat the two steps for all those computers (a) key generation, (b) add key to github.

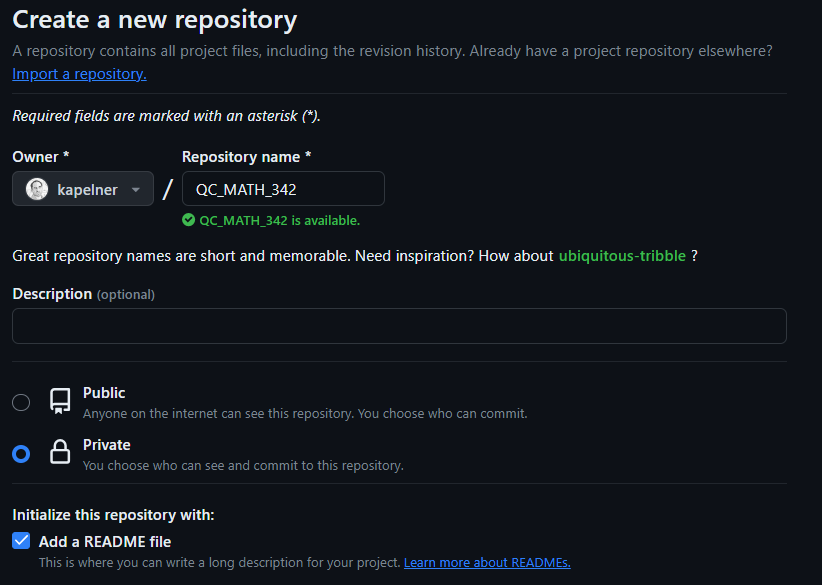
Go back to the terminal window and exit the vim editor by typing “:q” and pressing enter. You should now be back to the plain terminal screen.

Okay now you are ready to create your own repository, the repository that will store all your homeworks and labs. Go to github.com and click the green new button:

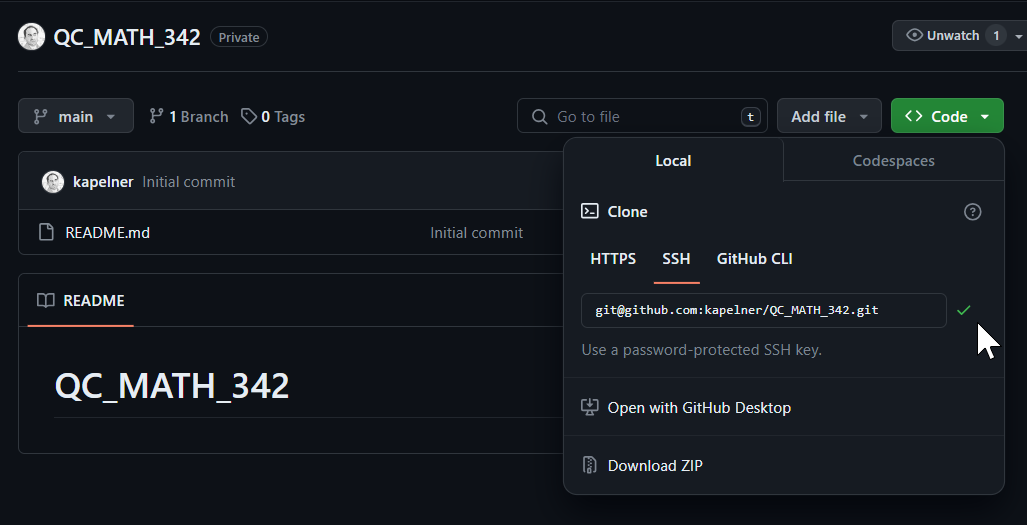


Under “repository name” type “QC\_MATH\_342”. Then click the “private” button (we will do this for now to prevent cheating). You are free to make it public at the end of the semester if you are proud of your assignments.

Then check the box “Add a README file”. That checkbox is required as otherwise you won’t be able to clone. Your screen should look like:



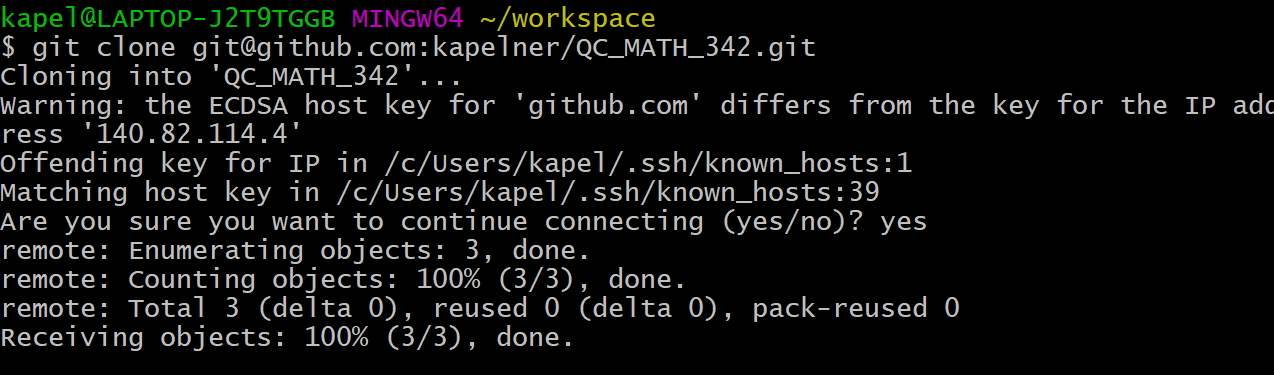
On the bottom, click the green “Create Repository” button. You should now be taken to the repository homepage. As before, we want to clone it. So click the green “Code” button and then the SSH option and click the copy button until you see the green check mark:



Now let’s clone it which will put it on your computer and allow you to add and edit files. Go back to your terminal. Type the following and paste that copied text where it says so:

git clone [paste]

For [paste] you’ll press ctrl + v (or command + v or shift + insert) to paste your ssh link that you just copied from your repo page. My link is “git@github.com:kapelner/QC\_MATH\_342.git” but yours will be different as your username will replace my username, “kapelner”. Press enter to run. If it complains “are you sure…” just type “yes”. You should then see the following:



You now will have your repository folder on your computer. Let’s go inside that folder

cd QC\_MATH\_342

You can now create all the subdirectories for all the class assignments:

mkdir homeworks

mkdir labs

mkdir modeling\_paper

mkdir final\_project

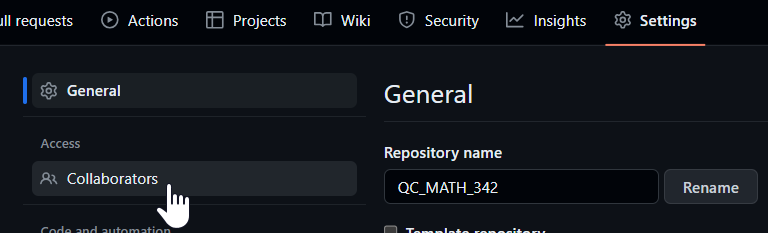
You should also make sure git knows your email address and name by running

git config --global user.name "your full name here"

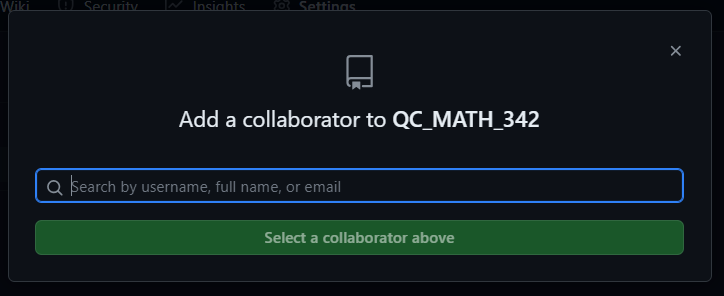
and then running

git config --global user.email "your email address here"

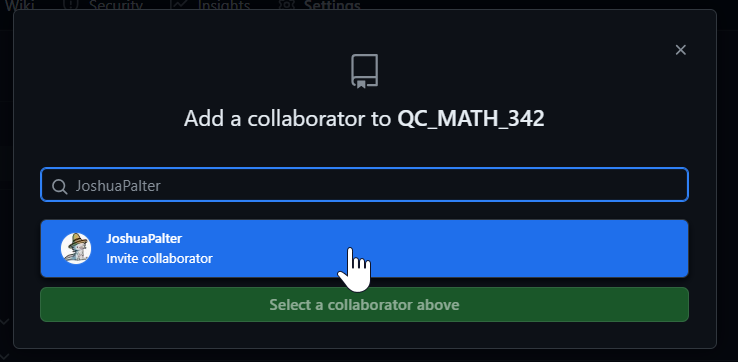
The last step is to add me and the TA as a collaborator on your private repo. This allows us to access your files and grade your assignments. If you don’t do this, you will obviously not get credit for any of your homeworks. On the repository page, click “Settings” and then on the left menu, click “Collaborators”



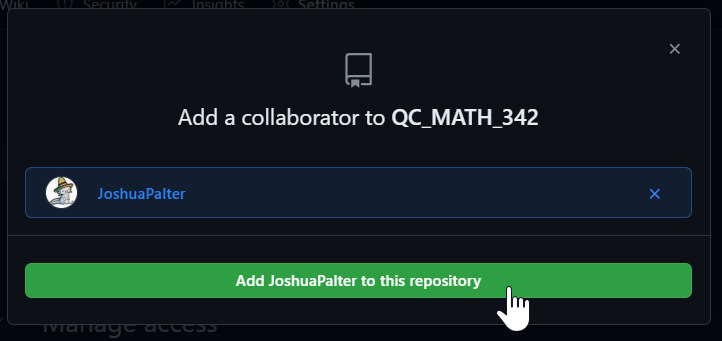
Now it will ask for your password again. Then click the green “Add People” button and you’ll see:



You’ll type in “nebryan” (that’s your TA’s handle) for and click the dropdown account when it comes up. (Note: the screenshot below was for a previous year’s TA who’s handle was “JoshuaPalter”).



And click “Add nebryan” to this repository



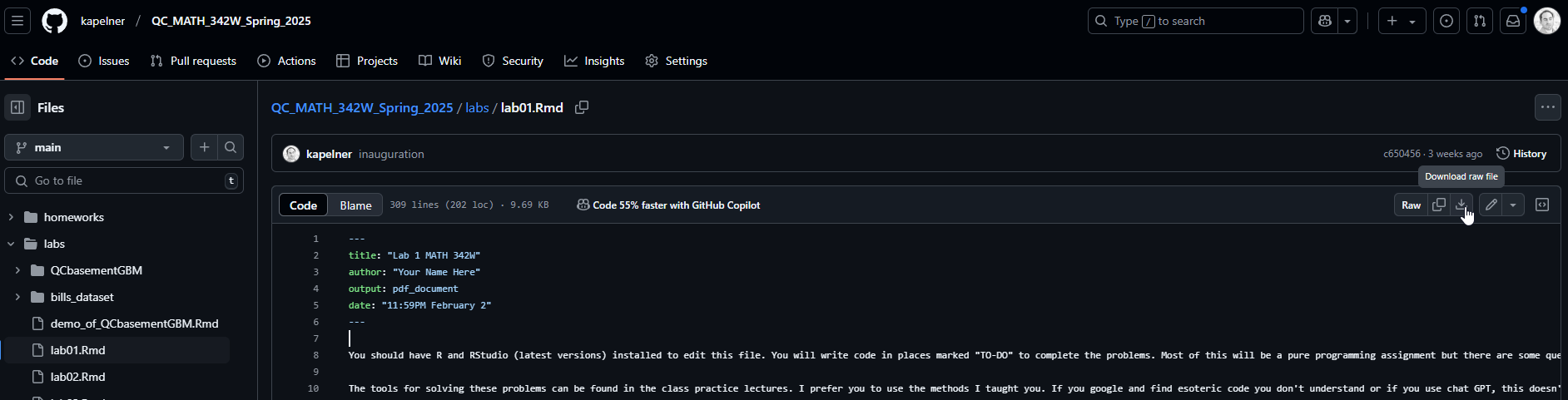
Now repeat this process again to add a second collaborator, me. Type in “kapelner” then select my account and then click “Add kapelner to this repository”.

If done correctly, you should see both of us as “pending Invite”.

Congrats you’re now al set up! There are many great resources about git you can find online if you want to learn more.

# How to Upload your Labs, Homeworks and Writing Assignments

Since you are setting this up at the beginning of the semester, I will use lab 1 as the canonical example. First, go to lab 1 on the course homepage via navigating your browser to <https://github.com/kapelner/QC_MATH_342W_Spring_2025/blob/main/labs/lab01.Rmd> and then download the lab using the “download raw file” icon:



Now copy this downloaded file “lab01.Rmd” into your QC\_MATH\_342/labs folder.

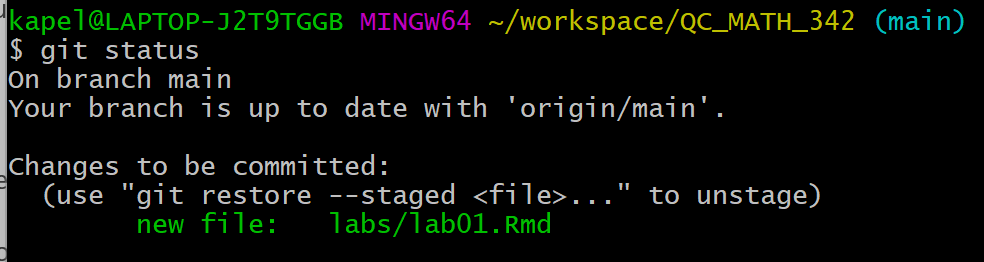
We will now add this file to your git source control repository by running

git add .

from the terminal. The “add” command adds new files. You only need to run it when you create new files, not when you edit old files. You can make sure lab01.Rmd is added by running

git status

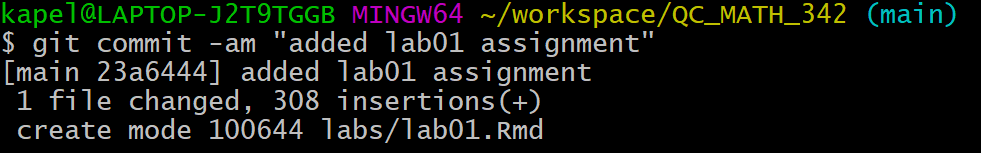
You should see it listed in green. The “status” commands tells you if anything has changed.



Now we will commit this file. Always write comments that make sense.

git commit -am “added lab01 assignment”

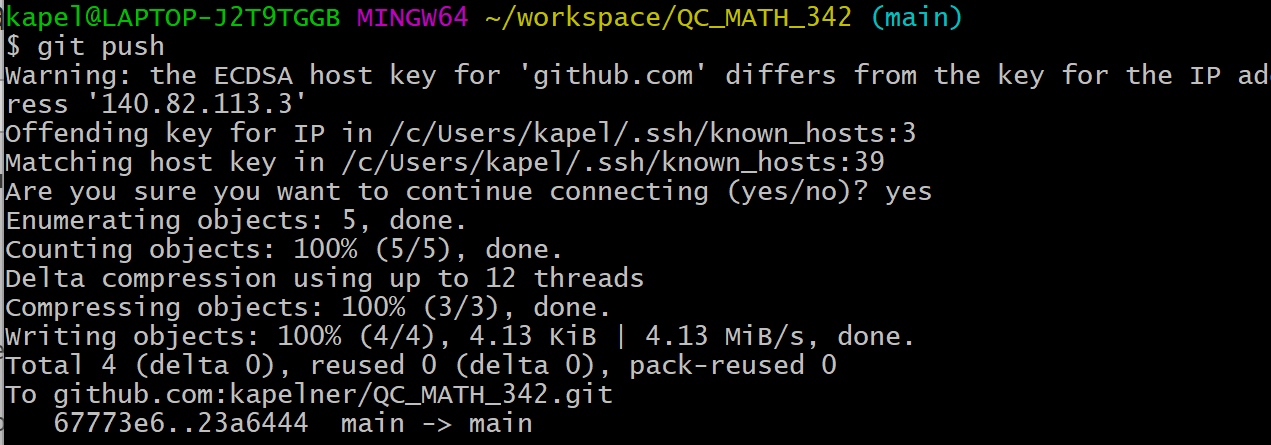
For some reason the quotes don’t copy well from Microsoft Word. Another thing Microsoft gets wrong. So type the command yourself manually if it didn’t work. You should see the following if it ran successfully:



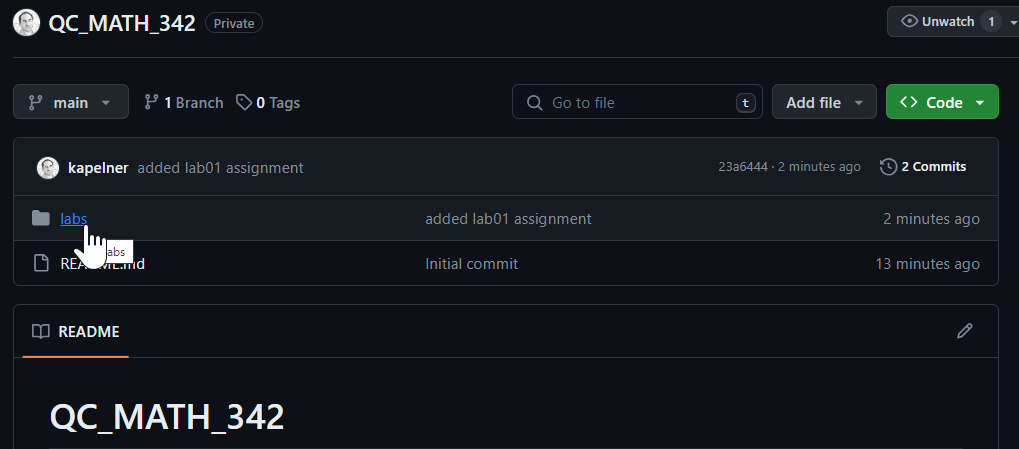
Now we’re ready to push these commits to github via

git push

If this worked successfully you should see



Now if you refresh your github page you’ll see the new directory:



In the future, you’ll repeat this

git add ., git commit -am “lab x or whatever”, git push

sequence for each new assignment (lab, homework or modeling\_paper or final\_project). For the theory homeworks, you’ll add the PDF of your scanned homework to the homeworks directory and name it “hw0x.pdf”. Ditto for the modeling\_paper or final\_project in their appropriate directory.

Make sure they’re in the appropriate places otherwise they will receive a zero.