HW00: Getting Started

- Assignment HW00 should be done individually.
- You should seek out help on this assignment in the evening lab or the #csc226 Slack channel.

Learning Objectives

- Learn some new terms related to computer science.
- Setup your coding environment, including your IDE and other tools.
- Practice using all of the tools we will need in this class.
- Get to know the TAs who will be supporting your learning in this course.

How to Start

1. We will use Google Drive significantly in this course. For that, you'll need a Google account. If you don't have an account already, register for an account here.
2. To begin, make a copy of this document by going to File >> Make a Copy...
    a. Check the box to "Copy comments and suggestions"
3. Change the file name of this document to ***username1* - HW00: Getting Started** (for example, **heggens - HW00: Getting Started**). To do this, click the label in the top left corner of your browser.
4. Go to the **Share** settings for this document (top right). Set them to "Anyone with the link can view". This is how we will be able to access this document to grade you:

Tools

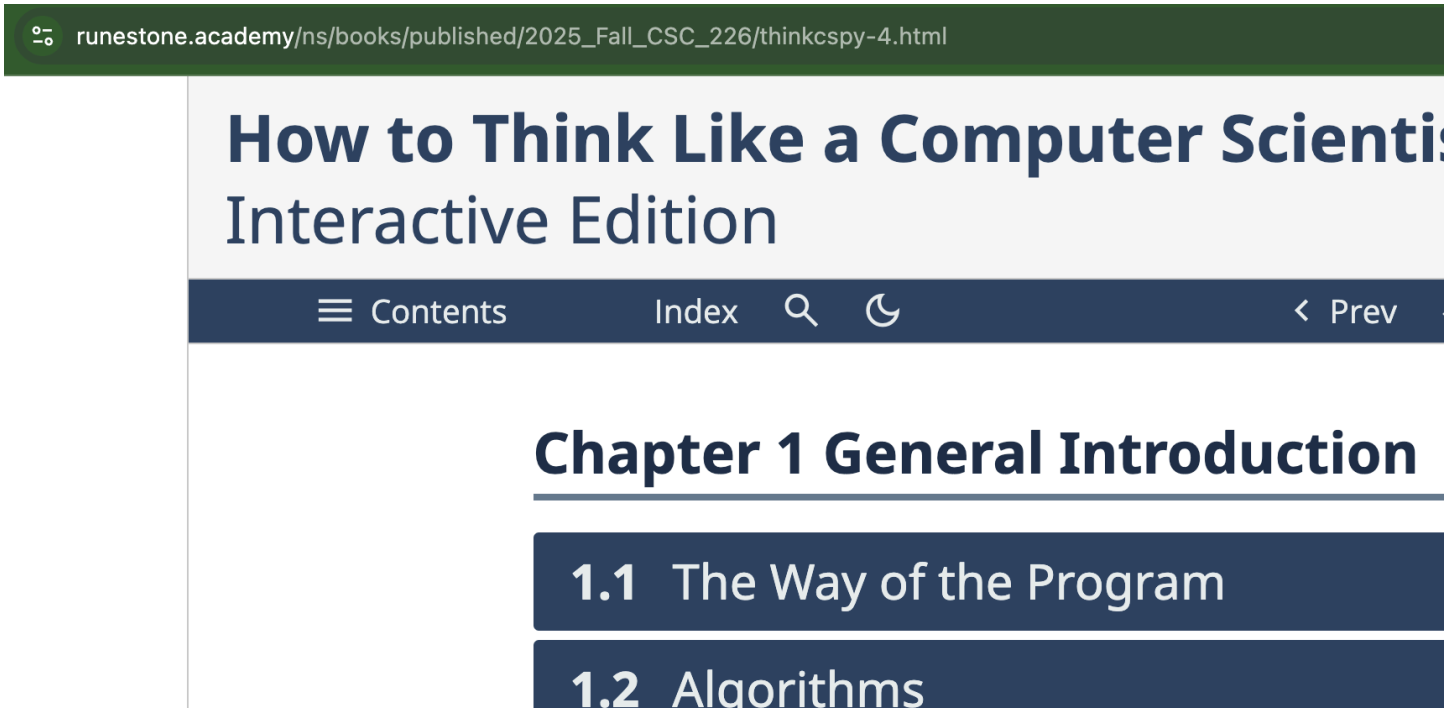
To be successful in this course, you will need to become familiar with using several primary tools. You will need to install some of these tools and register for some new accounts. This assignment is intended to carefully walk you through the steps necessary, so please **read it carefully**. But before you dive into the deep end, let's make sure you have the support you need to be successful in this assignment and in this course.

Task 1: Runestone Book and Friday's Quiz

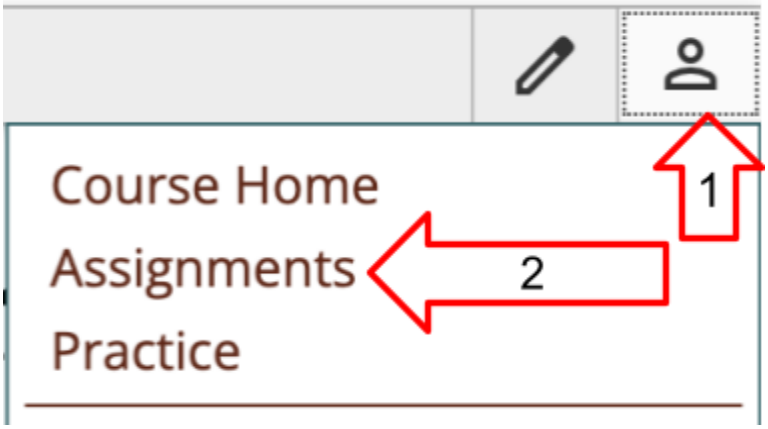
Friday's quiz will cover two topics: the course syllabus and Chapter 1 in our textbook. Let's make sure you can be prepared for that quiz:

1. Go to **Runestone Interactive**.
2. Make an account in Runestone:
    a. To register for a new account, hit the big Blue SIGNUP button at the top.
    b. On the Runestone registration page, fill it out the form:
        - USERNAME must be **your Berea username** (e.g., heggens for Dr. Heggen).
        - EMAIL must be **your Berea email address** (e.g., heggens@berea.edu for Dr. Heggen).
        - COURSE NAME field should be **2026_Spring_CSC_226**.
        - Check the box for I AGREE to the terms and conditions.
        - Do NOT check the box for CREATE COURSE. That will spawn a new book, not use my book!

c. Now when you go to https://runestone.academy/, it should take you to our textbook. You will know you are logged in when you see (Note the URL at the top includes the book name matches this term):



3. Next, go to **Assignments** in the top right.



4. You will see all your reading assignments. So far, only one reading should show in your assignments list.

According to the assignments page, what is the name of the first assigned assignment?	1.a. “Reading 01: Syllabus and Chapter 1”
When is it due (date and time)? NOTE: The time will always be the same for all readings, on the day of the quiz.	1.b. “2026-01-16 09:00:21”
Click on the first reading assignment. How many	1.c. 15 (14 excluding the exercises)

sections of chapter 1 are there to read?	
Make sure you complete this part of the assignment before Friday's class.	

Task 2: Evening Lab

As you *surely* remember from the **course syllabus** that you just finished reading in preparation for Friday's quiz, you'll know there is an evening lab that supports all of our courses in Computer Science. This part of this assignment is to visit the evening lab.

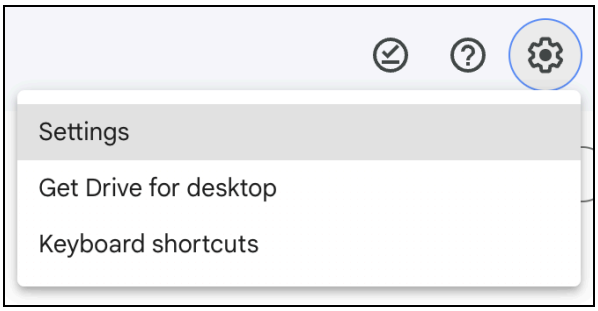
Questions and Instructions	Your Answers
According to the syllabus, when is the evening lab open (days and times)?	2.a. Sunday-Thursday; 7-9pm
Where is the evening lab located?	2.b. CMIT 361
Visit the evening lab at your earliest convenience. Speak with any one of the many teaching assistants at the lab. Most have taken CSC 226, and are happy to help you with the course. Ask them three or more questions of your choosing. Some suggestions (please don't just ask the following questions. it'll become very boring for our TA's to answer the very same questions 20+ times) -- Instead, ask them what you really want to know that is relevant to either the CSC 226 course or to the CIS major or minor! • Why did you pick CS? • What was the most challenging part of CSC 226 for you? • What do I need to do to be successful in CSC 226? • If you could go back in time and talk with your pre-CSC226 self, what advice would you give? • What do you think of the coding interviews in this class?	2.c. Question 1: How have you dealt with all of the readings related to the CS classes? TA's Answer 1: By focusing more on the practice of coding and spending only the necessary time on the readings, highlighting what's important and reviewing the highlights at the end. Question 2: What has CSC 226 helped you improve on? TA's Answer 2: As the classes is object-oriented, problem solving has been the greeted improvement of the TA's as they practiced step-by-step solutions and learn the process of utilizing pseudo-code to help plan their work and take on problems piece-by-piece. Question 3: What advice would you give me in preparation for the future of my CS major? TA's Answer 3: To immediately put the code you learn into practice. This was suggested to be done by setting aside a GitHub repository and story code learned throughout the textbook that you practice: display personally-created functions or practice exercises, etc.
The reason you are visiting the evening lab is two-fold: 1. To enlighten you to a valuable resource for seeking out help in this course. 2. To prepare you for your Tri-Weekly Check-ins 3. So you can get help with the installation processes, which are coming in 3, 2, 1…	

Task 3: Google Drive

You've made it this far, so you're off to a good start!

One of the best features of Google Drive is the ability to allow multiple people to write in the same document at the same time. Most homework and teamwork assignments will be accompanied with a Google Document just like this one. Since we will be exploring Google Drive in our first teamwork, we will leave much of that exploration for later. However, some things you should explore that will help you later in this course (and life, in general):

Go to Google Drive's website. Explore the interface. Note the "Get Drive for Desktop" link in the top right. You'll want to consider installing this application, and putting ***all*** of your files (from now until the end of time) in the **Google Drive** directory it creates on your computer inside the **My Documents** folder. It can serve as your backup solution until you graduate, ensuring you ***never*** lose any of your work, even if your computer crashes. Ever!



For your reference, here's a helpful 4 minute video on why and how to do the installation.

Task 4: Trello Course Agenda

All course materials, including the syllabus, are posted on the **CSC 226 course agenda.**

This is the course agenda. You will find all of your due dates and expectations here!

Some notes about the agenda:

- **I will update the course agenda regularly.** You should check it regularly. If a due date gets extended, it will be announced there. Although we will make regular announcements in class and via Slack, you are responsible for knowing what is due each day.
- The course agenda contains the **official due date** of all course materials.

Visit the website site, and answer the following questions:

At what *date and time* is **HW00: Getting Started** due?	3.a. Jan 14 - Jan 21, 11:55 PM
At what *date and time* is **RQ01: Syllabus & Chapter 1** due?	3.b. Jan 16, 9:00 AM
Read then paraphrase the listed rules about using outside sources:	3.c. CITE. Done. Essentially, if we are to use something like a Reddit Forum or ChatGPT for *anything* that will impact our work, we should cite it.

Task 5: PyCharm IDE

To write code, typically you'll want to use an **Integrated Development Environment (IDE)**. For this course, we will rely on the **JetBrains PyCharm IDE (Community Edition)**. You'll need to:

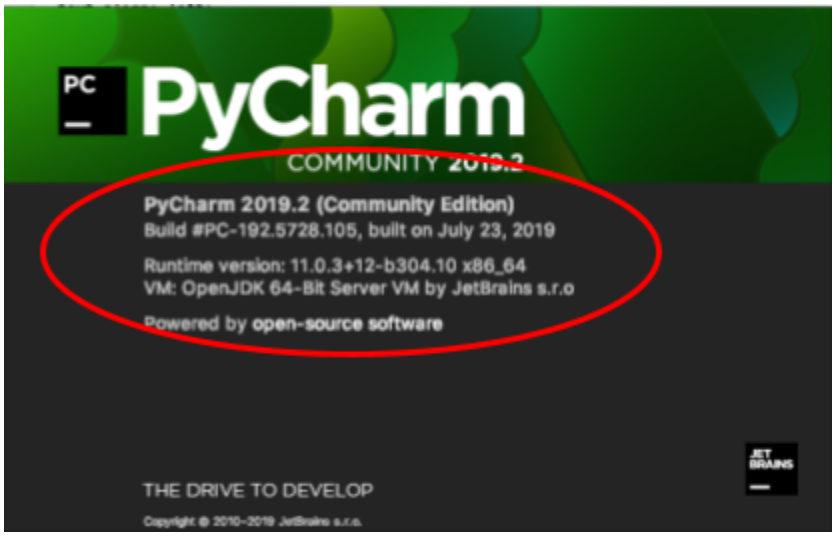
1. Register for a Jetbrains account.
2. Request an educational license, which will give you free access to all of the JetBrains features (including other IDEs for other programming languages). Scroll down the webpage until you find the APPLY NOW button.
3. Download and install the PyCharm IDE. If asked, select the COMMUNITY version.

Once you've installed PyCharm, open the application and sign in using the Log In to JetBrains Account... button. Play around in the interface to become more familiar with it. However, we will be exploring the interface in our first teamwork assignment on Friday, so don't fret if it seems overwhelming right now.

Task 6: Git and GitHub

As part of this course, we will be introducing you to a tool often used in software engineering known as **git**.

Git is a **version control system** for software. Ever notice how some software has a version, such as PyCharm?



One way software engineers control this versioning is through git. Git also has another extremely powerful use for software engineers; preventing two (or more) programmers from deleting each other's work.

Imagine you're building a program such as the software you're looking at right now: Google Drive. Certainly you realize that more than one developer created all this. To ensure multiple developers don't clobber each other's hard work, they use git.

So, each developer **clones** all of the code, makes some changes to that code, then requests the code be incorporated in the final product. Git monitors all of the files, and when it notices two developers write code in the same spot, it flags those two sets of code as having a **merge conflict**. Then, the developers can discuss whose code should be kept, or modify the code so that both can be kept. By developing good communication within an organization about who is working on what features, coupled with git as a failsafe, developers are able to produce very large programs quickly, with less duplication of labor, and with less errors. More on this later...

Describe two or more reasons git will be useful in this course, based on what you know now.	4. a. We plan to work in teams, thus being

HINT: no answer is wrong here, except not trying.	able to work on the same project at once without having to wait on our partner to finish will allow for better efficiency. b. We will have more backups of our work, which is always a plus.

First you will need to register for a Github account. Use your **Berea Email address** and a password you'll remember. Github is a website that uses git plus a host of additional tools for managing code, which we'll explore later as well. It will also hold all of your assignment submissions.

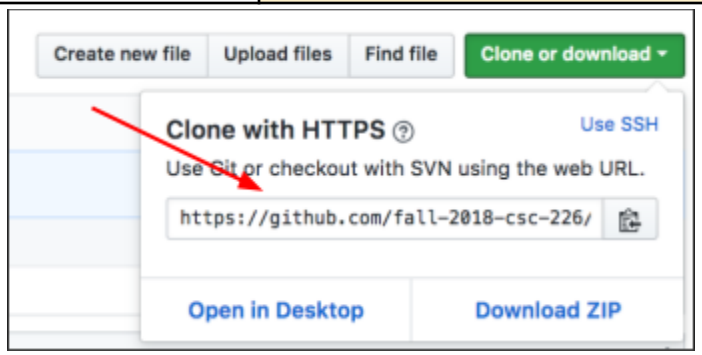
Up to this point, you've created accounts for Google Drive, Jetbrains, Runestone, and Github. You will need to remember the passwords for these systems. I recommend a password manager, like LastPass, if you do not want to have to remember them all (with LastPass, you only need to remember the master password to get to all your other passwords).

Task 7: Set up PyCharm Settings

Github

Next, join our **Github classroom for HW00**. Select your email from the list. Follow the remaining prompts until you arrive at the main page of the repo (you'll see three files: a **LICENSE**, a **README.md** file, and a **.gitignore** file).

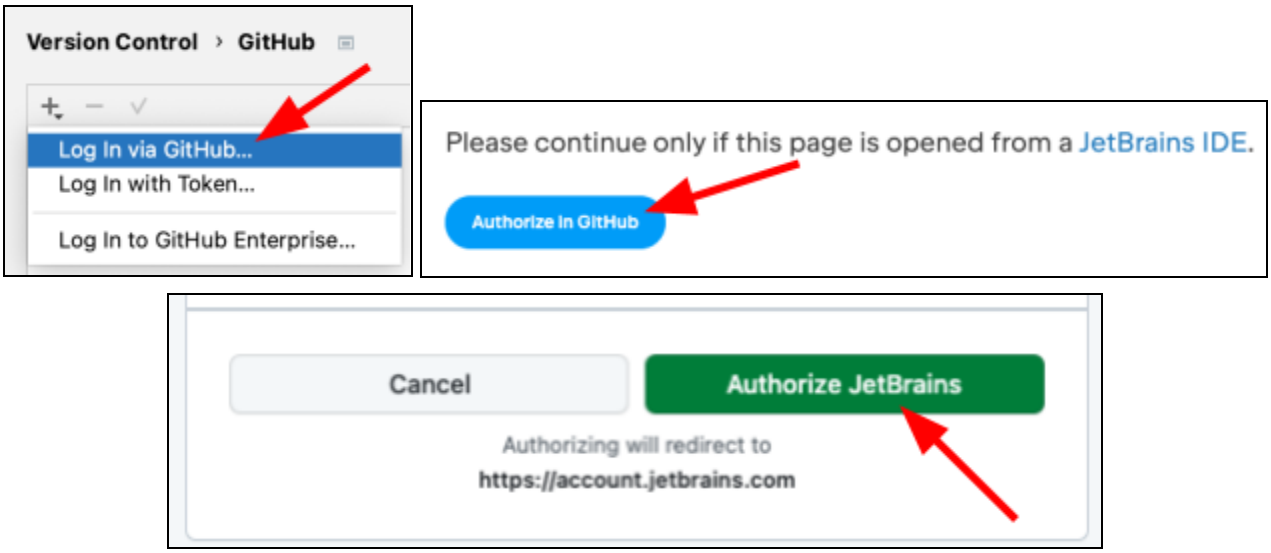
Copy your repository link, indicated by the big red arrow below, and paste the link here:	5. https://github.com/Skylar-the-IDE/CSC-226.git



We'll come back to Github later. For now, head back to the **PyCharm IDE**.

Next, we need to change some PyCharm settings to allow Git integration:

1. Go to your PyCharm settings (under File), and go to the **Version Control** section. Under that section is **Github**. Click into it.
2. Configure PyCharm so it knows about your Github account that you just created. Hit the + symbol, then select "Log in via Github...". Follow the prompts, accepting them when asked:



If you've done this correctly, your username should appear on the PyCharm settings page after you select "Log in". Save the changes and you're done with Github setup! If you run into issues, let the instructor or TA know via Slack or the evening lab, and we will help you debug the issue. Many students get stuck here, so do feel encouraged to ask for help!

Git

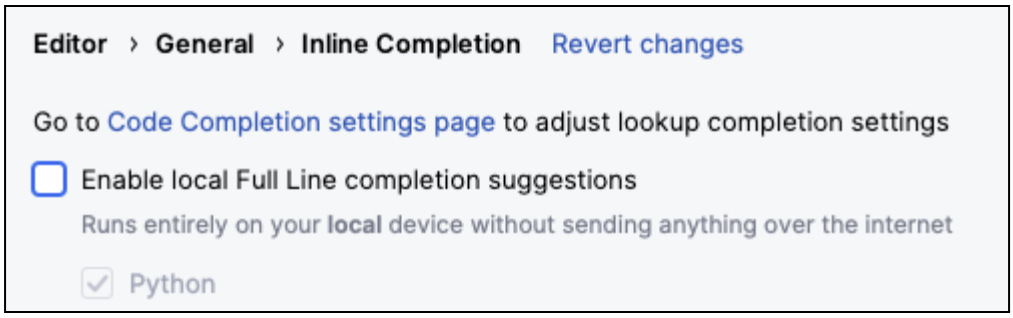
While we are in the settings, PyCharm will likely also want you to install Git. Go to the settings page for Git (the one above Github on the left). At the top, click the "Install Git" link and follow the instructions.

Now you'll be able to use Git and Github to manage all your code, work collaboratively with other students in the class, and never lose your code because it's saved in Github!

Autocomplete

The last setting we need to change is Autocomplete. Later in this class, you will be allowed to use Autocomplete. For now though, autocomplete ***has been found to be one of the most detrimental tools to student learning***. Autocomplete is an excellent tool when you understand what the IDE is suggesting. When you are new to a concept, however, it gives students a false sense of learning. They feel like they understand the code they are "writing", when in reality, the IDE took away the opportunity for you to understand that concept by coming up with the solution yourself. I have found that students who have autocomplete turned on prior to the first coding interview *tend to do the worst on that first interview*.

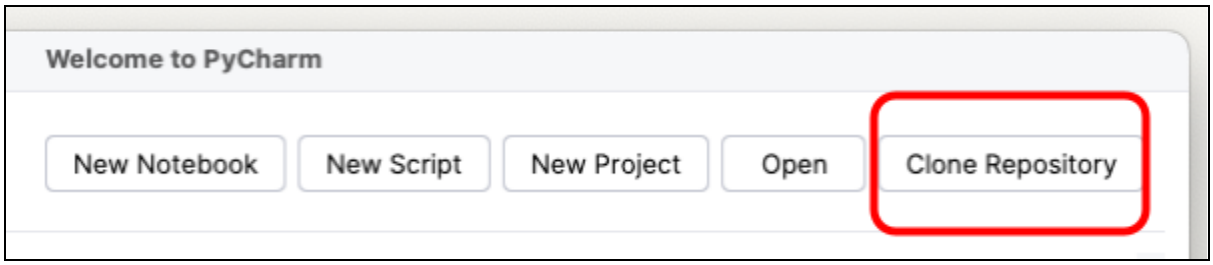
To turn off autocomplete, go to the Editor >> General >> Inline Completion setting in PyCharm. Uncheck the box for "Enable local Full Line completion suggestions":



Task 8: Using Git in PyCharm

Now, let's use our first repository in PyCharm. You've already created the repository (i.e., repo) in Github, and the link to that repo is above in Question 5. Copy the link.

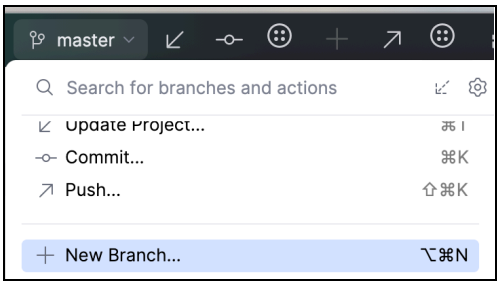
If you are not at the "Welcome to PyCharm" screen, click **File >> Close Project** which should take you back to the welcome screen. On the welcome screen, click the "**Clone Repository"** or **"Get from VCS"** option, shown below:



Paste the link from Question 5 in the URL box. The Directory is where the files will go on your local machine. You can leave the last one the same as the default, or change them. Your choice; just know this is where your files are located. **Click Clone**. Your PyCharm project and your Github repo are now in sync!

Next, let's edit the README.md file so we can learn how to make changes using git.

1. Create a new branch. At the top of PyCharm will be the word "main". Click it, then click "New Branch":

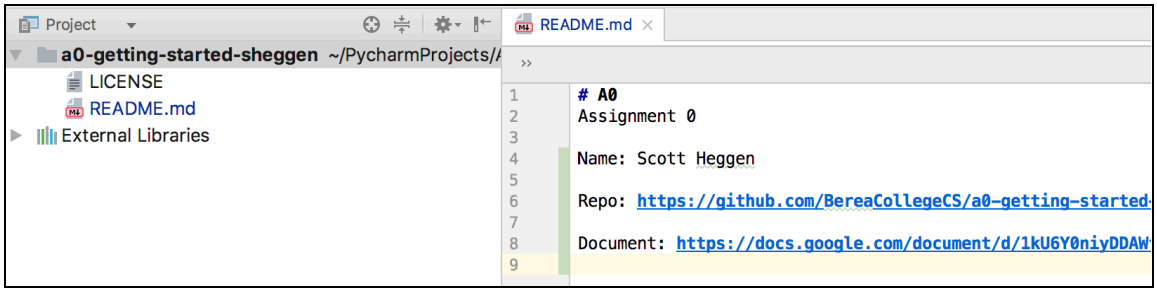


2. Name your branch ***HW00_username***, replacing username with your Berea username.

3. Open the file by double-clicking README.md in the left pane. You may need to switch to **Editor** mode to make changes.

Make the following changes to the README.md file:

- Add your name
- Add your Github Repository link (the URL from Question 5)
- Add a link to **this Google Doc**. To get that link, click the "Share" button in the top right of this document, and then click "Get shareable link".

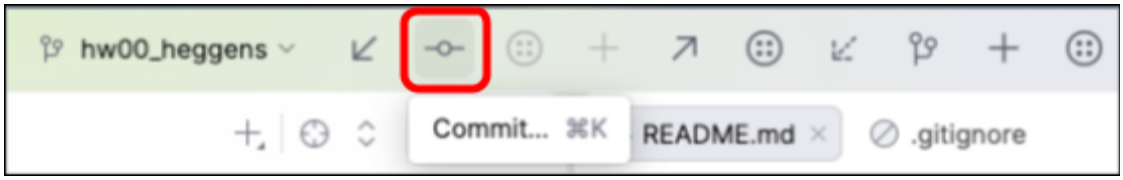


Changes you make in PyCharm are saved automatically in your **local repository**. They are not submitted to the **remote repository** (Github) until you've **committed** and **pushed** them. Committing code is a way of marking important points when coding. Often, commits are made when a small piece of code is working, and you want to mark that point so you can get back to it later. Commits always come with a commit message; the better the message, the more valuable the commit is later. Commit messages like "Edited file" are usually not helpful; a better message would be something like "Added name, repo link, and document link to README".

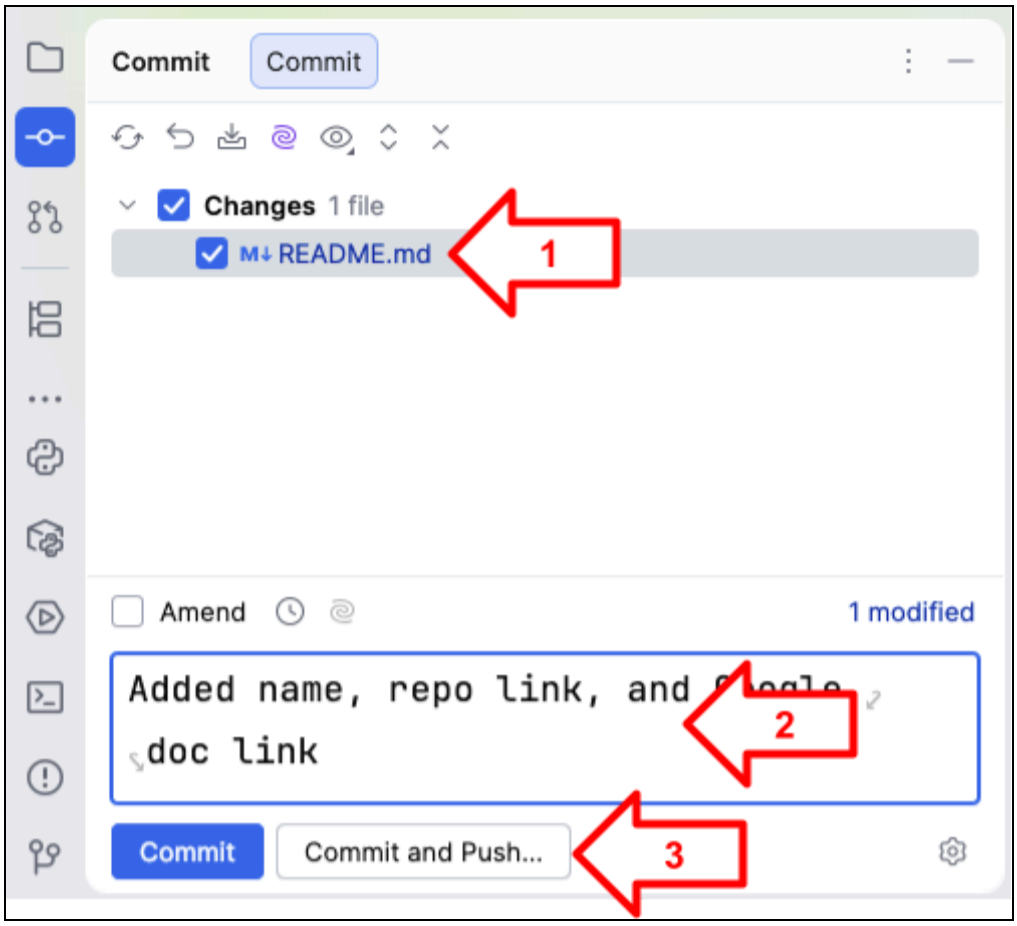
After you've made a commit (or many commits), and you want to make your code a part of the remote repository (Github), you're ready to **push**. Pushing often happens when a feature is complete, such as an entire assignment. We are ready to both commit and push your changes to the README.md file.

Let's practice this process for the first time:

1. In the top toolbar, there is a Git menu. Click the icon representing "Commit":



2. Add a meaningful commit message, then click "Commit and Push" at the bottom:



3. Click the “Push…” button on the screen that follows.

In a web browser, open the link you copied in Question 5. At the top is a button labelled main. Switch that to the branch name you created in PyCharm (**hw00_username**). Do you see your changes in Github? If so, this means you were successful in creating a change on your computer and using git to push (i.e., submit) those changes to Github. Congrats! If not, come get some help. There are many reasons you could have hit a snag at this point, and not all of them are your fault! We are happy to help you get unsnagged!	6. Yes, I see the “HW00_mcdaniels” branch within the website form of Git.

You'll be repeating Task 8 at the end of this assignment to complete this homework and make the final submission, and every time you submit teamworks and homeworks. Needless to say, this process is important and we'll be practicing it a lot moving forward!

Task 9: Slack Channel

Lastly, go to the **course Slack channel**. Post a one paragraph introduction with your name and a large picture of yourself with your face clearly visible. Tell us about yourself and what you enjoy! Over the next week, read what others posted and react to some of them!
My Intro:
Hello, guys, I'm Skylar–I'm from a small town in Missouri and I'm here to study CompSci for the sake of creative development. I really enjoy creating, whether it be games, paintings, spritesheets, mods, or whatever else I've done. Hope to learn a lot with you guys!"

Recap

A lot was done in assignment HW00. Let's make sure you completed it all. For each item below, answer the following: 1. Did you get the task complete? 2. What was the most challenging or unclear part of the task?
Task 1: Runestone and Friday's Quiz 1. Yes 2. The only unclear part was what would be on the quiz/ how much we really needed to study
Task 2: Evening Lab 1. Yes 2. It was, at first, unclear to me as to what I should do in the lab–but the people quickly helped me
Task 3: Google Drive 1. Yes (for years now, haha) 2. Nothing–perfect in all ways
Task 4: Trello Course Agenda 1. Yes 2. No issues navigating
Task 5: PyCharm IDE 1. Yes 2. It inherently has unclear parts as all complex applications and software-developing programs do, but I figured out the basic navigation (any issues were how the push/pull works between it and Git and what updates/ how they update)
Task 6: Git and GitHub 1. Yes (i think…) 2. The struggles were the same as the PyCharm ones as co-updates are confusing but with practice I'll figure it out
Task 7: Set up PyCharm Settings 1. Yes 2. No issues
Task 8: Using Git in PyCharm 1. Yes 2. The biggest problem between the two is knowing which one I need to edit for specific things (I can

edit the README on both, but which SHOULD I edit it on) and which buttons I need to specifically use to update between them (like PyCharm has several “update” or “commits” but which do I use and when)
Task 9: Slack Channel 1. Yes 2. No issues

If you were unsuccessful in completing any of the tasks above, reach out for help from the instructors or the TAs. Get help early so we can ensure you are able to use the tools needed to be successful in this course, even if it’s after the due date for this assignment!

Submission Instructions

Follow the submission instructions to submit your work by next Wednesday at 11:55PM.