



Workspace Repository Setup

Before beginning these instructions you should have first updated your operating system and then installed the required software (Docker Desktop and Visual Studio Code).

O. Find Your Canvas UNC E-mail Address

Canvas UNC e-mail addresses typically end in @live.unc.edu or @email.unc.edu!

Find your Canvas e-mail address by logging into Canvas, clicking on "Account" in the top right corner, selecting "Settings", and looking under the Email field on the right.

Use your Canvas e-mail address when you login to Gradescope and register for GitHub.

1. Register for GitHub

GitHub is for backing up your course Workspace Repository and for downloading course materials.

You can think of GitHub as a social network where people and organizations share and collaborate on code with one another. Organizations like NASA, NOAA, Peace Corps, Washington Post, New York Times, and so on, host projects publicly (and privately) on GitHub. It's a valuable service for software engineers, data scientists, research teams, and more.

Your coursework will have its own private "git repository". The technology we use for backing up versions of our projects and transmitting them to or from the internet, hosted on GitHub, is called *git*. Visual Studio Code has git support built into it. Git is a professional *version control* tool we use in limited ways in COMP110, but if you continue on in a field that involves programming and data you'll likely encounter it again.

1. Register for a GitHub account with your primary UNC address: <https://github.com/>
 - You are encouraged to choose your ONYEN as your username, if it is available (not required)
2. After registering, edit your profile from this page: <https://github.com/settings/profile>
 - Be sure your name and email match what is in Canvas
 - Fill out the rest of your GitHub profile with a high quality photo of yourself, a brief bio, UNC Chapel Hill as your "company", and Chapel Hill (or home) as your location if you are going to continue in a field that involves programming (computer science, data science, information science, and so on). Think of this as a secondary LinkedIn when it comes to future opportunities, because that's often how employers and graduate schools look at it.

2. Login to Github in VSCode

1. Open the VSCode application on your computer.
2. In the bottom left corner, look for a user/profile icon above a settings gear icon. Click it.
3. Click "Sign in" or "Sign in to sync settings"
4. This will open a browser window. If you are not already logged into GitHub, you will be prompted to login. If you are logged in, you may get redirected back to VSCode. Follow the prompts until you are taken back to VSCode.
5. Confirm you are logged in by clicking the user/profile icon in the bottom left corner of VSCode. You should see your GitHub username followed by (GitHub).

3. Installing VSCode Dev Container Extension

VSCode can be customized with extensions specific to the types of work you use it for. We need a specific extension to enable a feature called "DevContainers" short for Development Containers. This feature of VSCode, combined with Docker, is what will ensure you and all of your peers in the course have exactly the same software setup.

1. From the View menu, select Extensions
2. In the search box, type: "Dev Containers".
3. Click the top result that is verified as published by Microsoft.
4. Click the Install button to install the extension.
5. If prompted to reload the VSCode window, accept.

4. Setup your Workspace in Visual Studio Code

2. In VSCode, from the *View* menu and select *Command Palette*

Overview

- 0. Find Your Canvas UNC E-mail Address
- 1. Register for GitHub
- 2. Login to Github in VSCode
- 3. Installing VSCode Dev Container Extension
- 4. Setup your Workspace in Visual Studio Code
- 5. Setup your Backup Workspace on GitHub
- If You are Asked to Configure Your Github User Information
- 6. Confirm Everything is Good to Go
- 7. Get Started on EX00 - Hello World



3. Type in "Git: Clone" (without the quotes) and press **Enter**.
 - If you do not see this option after installing course software previously, try rebooting your computer.
4. Copy the following URL: **<https://github.com/comp110-26s/comp110-26s-workspace.git>**
 - Right Click and Paste the copied URL into the text box that appeared in VS Code after the previous step.
 - Or use the Paste keyboard shortcut: **Control+V** Windows / **Command+V** Mac
5. Press Enter and you will be asked where you want your course workspace repository to be saved on your computer. Accepting the default suggestion is OK, but if you have a folder you would prefer to keep the course work go ahead and select it. **Please don't move this workspace until after the semester has ended.**
6. Press OK when asked if you would like to open the repository after it downloads. If you get a pop-up message in the bottom right corner, asking if you would like to create a virtual environment for your Python interpreter, please *do NOT* click "Create".

5. Setup your Backup Workspace on GitHub

1. Navigate a web browser to <https://classroom.github.com/a/8AS8F-OG>
2. Click "Skip to the next step" when prompted to select your user / identification
3. Click "Accept this Assignment"
4. When the workspace is getting set up, you may need to refresh until it tells you you're ready to go!
5. Click the link to your personal backup repository that looks something like this **<https://github.com/comp110-26s/comp110-workspace-iHinks>** (except instead of **iHinks** you will see your GitHub username).
6. Look for the "Quick setup- if you've done this kind of thing before" box. You will see the same link that you clicked on to get here in step 5. Copy that text or click the copy to clipboard (overlapping squares) button.
7. In VSCode, open the *View* menu and select *Command Palette*, the shortcut for this menu is:
 - Windows: **Control+Shift+P**
 - Mac: **Command+Shift+P**
8. In VSCode, type in *Git: Add Remote* and press enter with the option selected.
9. In the blank text box that appears, paste in the URL to your backup repository that you just copied. Press enter.
10. When asked for "Remote name" type in: **backup**
11. Open a new terminal tab, from the *Command Palette* select "Terminal: Create New Terminal"
12. In the terminal that pops up, type: **git push --set-upstream backup main** then press enter (two dashes in the of **--set-upstream**). This command backs your work up to your GitHub repository.

If You are Asked to Configure Your Github User Information

You may receive an error message when you try to push to git (step 12). If so, try these steps and try again.

1. If your *Terminal* window from above is still open, great! If not, open a new *Terminal* pane in VS Code.
2. At the prompt, type the following command and press enter with your primary e-mail address inside the double quotes:

```
git config --global user.email "your@primary.email"
```

3. Then, type the following command and press enter with your first and last name inside the double quotes:

```
git config --global user.name "Your Name"
```

After doing this, when you make backup commits in **git** they'll be correctly attributed to you.

6. Confirm Everything is Good to Go

Phew, you're almost to the end of setup! Hang in there!



First, make sure Docker is open in the background. Click "Accept", and do not sign in to Docker.

Next, in VSCode, open the Command Palette once more, type "Reopen in Container", and press enter. This step will take a few minutes.

VSCode should now be running in what is called a "Dev Container." To confirm this, look in the lower left corner of VSCode, in the blue box, you should see "Dev Container: comp110-...".

Once it's done, in the Explorer pane, you should see **.devcontainer**, **README.md**, and **welcome.py** (and potentially some other files and folders). You will notice some of the other files are now hidden; they are hidden because they are not relevant to your work in the course.

Let's run the welcome program! Open the terminal by going to the Terminal menu and selecting New Terminal. You should see a terminal window open at the bottom of VSCode. In the terminal, type **python -m welcome** and press enter. You should see a welcome message printed to the terminal!

Let's be sure you are able to exit out of VSCode and find your work again. Go ahead and close VSCode. It will take a few seconds for your DevContainer to fully shut down in the background, so count to ten, take a few deep breaths, and be proud of setting up a modern, professional software development environment on your personal laptop! Then open VS Code again. If you see the same files in the left-hand sidebar, and "Dev Container: comp110-25f" in the bottom left corner: great! Keep reading, though, so you know what to do in the future if you do not see your workspace.

If you do not see your workspace files in the Explorer sidebar, go to File, select Open Recent, and look for the topmost entry that has **comp110-** in it and ends in **[Dev Container]** and select it. This is how you can get back to your workspace most easily in the future.

7. Get Started on EX00 - Hello World

You are now ready to begin the first exercise of the course, "Hello, world!". Get started here: EX00 - Hello World

Contributor(s): Kris Jordan, Izzi Hinks

© 2025 Kris Jordan - Feedback Form - Made with in Chapel Hill