

# Computer Science 241

## “Lab 0” - Working on CSCI 241 Projects Remotely

Due: Nothing to hand in, but please complete this by Sunday, 9/27.

## Introduction

Getting set up to work on projects for CSCI 241 remotely requires a few steps. This mini-“lab” walks you through those steps so you’re ready to go and comfortable with the tools before I start piling on other new concepts.

Assignments will be written in Java, built using Gradle, tested using JUnit, and distributed and submitted using Git and Github Classroom. In lab 1 you’ll learn how to use all these tools, so don’t worry if you aren’t familiar. The purpose of this document is to get you set up with the tools you need to develop, test, and run your code remotely in the CS lab environment.

## Github

If you don’t have one already, you will need to create a free Github account. You can do this by visiting [www.github.com](http://www.github.com) and clicking “Sign Up” in the top right corner. You can use whatever account name and email address you like (or use your existing Github account if you have one and skip this step).

## Remote access to the lab environment

The CS lab systems have everything you need already installed. “All” you need to do is log in remotely and use them. There are various approaches to this with various levels of involvement in setup and use. This section walks you through the recommended approach: it requires some set-up ahead of time, but will give you a pretty seamless experience working on the projects almost as if they were on your local computer.

Here’s my recommended approach:

1. Verify that you can SSH into a CS lab system that is booted into Linux. Documentation on how to do this from Windows, Mac, or Linux are available at this CS Support Wiki page: [https://gitlab.cs.wvu.edu/cs-support/public/-/wikis/home/survival\\_guide/day\\_to\\_day/Remotely\\_Accessing\\_Resources](https://gitlab.cs.wvu.edu/cs-support/public/-/wikis/home/survival_guide/day_to_day/Remotely_Accessing_Resources) I recommend starting by logging into a least-used lab system ([labs.cs.wvu.edu](http://labs.cs.wvu.edu)).
2. Back on your own computer, install Visual Studio Code (<https://code.visualstudio.com/> if you don’t already have it). Also install the Remote - SSH plugin, available from <https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh>.
3. Follow the **Linux/macOS Setup** or **Windows 10 Setup** instructions on this CS Support wiki page: [https://gitlab.cs.wvu.edu/cs-support/public/-/wikis/home/survival\\_](https://gitlab.cs.wvu.edu/cs-support/public/-/wikis/home/survival_)

`guide/tools/VSCode_Jump`. You should now be able to log into a lab machine through VS Code and edit files in your CS home directory.

4. You likely got prompted to enter your password (possibly twice) in order to log in. If you would like to avoid having to do this every time you connect, you can set up **SSH Keys** and use **SSH Agent** to securely provide login credentials. Follow the instructions under the **SSH Keys** section and the **Utilizing the SSH Agent** section of this CS Support wiki page: [https://gitlab.cs.wvu.edu/cs-support/public/-/wikis/home/survival\\_guide/tools/SSH](https://gitlab.cs.wvu.edu/cs-support/public/-/wikis/home/survival_guide/tools/SSH).

If you've completed all of the above, you should be able to easily connect to your choice of CS lab computer, edit files, and run shell commands all inside VS code.

## Alternatives

The above is my recommended approach. There are other possible ways to work on the coding assignments for this class:

- You may set up your local computer with all the necessary tools (git, java, gradle, junit), but this is not an officially supported approach due the wide variety of hardware and operating system setups you may be working with. I will do my best to help you if you're having problems setting things up, but the officially recommended and supported approach is to work on the lab environment.
- If you know how (or wish to learn) to use a command-line text editor such as **nano**, **vim**, or **emacs**, you can simply ssh into a lab computer and start editing your files. I recommend using a second terminal to run commands so you can build/test frequently as you edit your files.

## Getting Help

If you have any trouble getting set up or find anything unclear, please get in touch with me or a TA, or contact CS Support at [cs.support@wvu.edu](mailto:cs.support@wvu.edu).