

CSCI 262 Data Structures

Spring 2019

Lab 1: Compile

[\(Back to Labs\)](#)

Goals

- Set up your development environment
- Compile a simple multi-file program

Files

- [Starter code](#)

Overview

Welcome to your first data structures lab! Labs in this class will be guided exercises designed to strengthen your abilities as a programmer and your understanding of the course material. This particular lab requires you to set up your personal development environment, so you will each need a computer to work with. For most future labs, we will ask that you work with a partner or a team. For today, just start by introducing yourself to the people on your left and right. You are welcome to check your work today with your neighbors.

Note: while we intend for the work in these labs to be completed in class, if you are unable to complete the work during the class period, you may complete the lab later in your own time.

For this lab you will work with some simple code we provide, making sure you can compile and run code in the CLion development environment. You will then answer a couple of simple questions in Canvas (most labs will have a *bit* more depth than this one). Note that **each** person on a team must answer the questions in Canvas to get credit for the lab! (This will be the case for all labs.)

Instructions

CLion

JetBrains CLion is an online integrated development environment (IDE) that is cross-platform (Windows, OS X, linux) and free for students and faculty. It can use different compilers on the "back end" to actually build your program; for this course, we will be using g++. CLion and g++ are already installed on all lab machines on campus.

To get started with CLion, you must first create a free student account with JetBrains at <https://www.jetbrains.com/student>. If you created a student account previously (e.g., in CSCI 261), it should still work.

Optional: you can also set up CLion on your personal laptop or desktop computer - you can do this whenever you want. You will need to start by installing the g++ compiler, if not already on your computer. The steps to do this depend on your operating system:

- Windows: Download **MinGW**. Open the zip file and drag the `mingw64` folder to the root of your C: drive.
- Mac OS X: Install XCode from the Mac App Store.
- Linux: Depending on your distribution, you may already have g++ installed. (You can find out by trying the command 'g++' at a command prompt.) If not, use your package manager to install the latest g++ package available on your distribution.

After installing g++, you can proceed to install **CLion**. (If you previously installed CLion, you can keep using the version you installed previously, or you can update to the latest. Either is fine.)

Note: if you really want to, you can use an IDE of your preference instead of CLion. However, you will be largely on your own if you run into trouble.

Configure CLion

For the most part, everything should be set up for you. The first time you start CLion, there are a few screens you need to move through:

1. Do not import any existing settings
2. Accept the Terms of Service
3. Login with the JetBrains account you just created
4. Choose a color theme (you can change this later)
5. For Windows, make sure that MinGW is selected (and not Visual Studio). On other platforms, g++ should be automatically detected.
6. Now click the "Skip Remaining and Set Defaults" to get on to the good stuff.

Create a Project

Depending on its complexity, a software project will have many different files associated with it, such as the source code, header files, resource files (images, sounds), data files, and configuration files. Most IDEs, including CLion, expect programmers to organize all of these files into an entity called a project.

If you've just started CLion, it will prompt you to create or open a project. For most assignments in this course, you will be provided some starter code, and this lab is no different. The starter code for this lab can be downloaded [here](#).

Download the starter code and think about how you want to organize your files. You probably want to keep your code on the Z: drive, or on a flash drive, if you are not working on your personal laptop. A good choice might be to make a directory like `Z:\csci262`, and make subdirectories for each lab and assignment as needed.

Since there is already a directory "lab01" in the downloaded zip file, just unzip or drag this directory into your csci262 directory (you can always rename things later).

Choose the Import Project from Sources option, or if you have already created a new project accidentally, go to the File menu and choose Import Project.... Then browse to the directory you created containing the source files for this lab. A page will pop up showing you the source files to import, which should be lab01.cpp, muddler.cpp, and muddler.h (leave all of these checked). Click OK and CLion will open on your project.

Note: Make sure you follow the instructions in the last paragraph carefully, especially the "Import Project" bit. If you instead just create a new project, CLion will by default create a main.cpp for you that prints a regular "Hello, World!" message. The lab 1 code should **not** print the same thing, so if you are seeing "Hello, World!", you need to go back through the steps, or talk to a TA.

Breathe...

CLion may spend a bit of time when you first create the project searching for various information in your system - things like C++ header files, libraries, and so forth. You won't be able to do any compiling until these tasks complete, so now is a good time to look at the source code.

In the left-hand pane in CLion, you should see your new project listed. You can expand or collapse this project; expand it to see what files you have. You will notice there is an extra file you did not import, named "CMakeLists.txt". You can ignore this for now. Make sure you have all three source files (lab01.cpp, muddler.cpp, and muddler.h). Spend a little time looking at the source code in these files (double-click on files to open in the editor). See if you can guess what the program does.

Compile and run your Code

Each time you use CLion on a lab machine, there is a setting you need to change. On the top menus, select Help > Find Action. Search for "registry" and hit enter. Uncheck the box that says "run.processes.with.pty". Close the window and continue on. This needs to be done every time on the lab machines, as the setting is not remembered due to the way these machines store your personal profile. (If you are using CLion on your personal Windows machine, you need to apply the same change, but it should remember it after the first time.)

To compile and run your code, simply click the green triangle button in the upper right corner of the IDE. This should result in a new pane or tab opening in the IDE to show you the results of running the program. Note the output of the program, and respond to the quiz for lab 1 in Canvas. If you are working with a partner, make sure your partner also does the quiz to get credit for this lab!

Troubleshooting

CLion has a few quirks, which you might occasionally run into. The "registry" hack mentioned in the previous section fixes one of these quirks, but there are others that occur less frequently. If you are getting odd behavior, errors, etc., be sure to check out the [CLion FAQ](#) section of the [Help page](#) of the course website to see if your issue is listed.

Play!

If you have completed all of the above successfully (and there is time left in the lab), this would be a good time to mess around with CLion and the code we just gave you. See if you can modify the code, recompile (using the same command as before), and see what your changes did. In particular, you might try using different strings in `lab01.cpp` to see if you can figure out what the muddler is doing.

Yay, lab 1 is complete. You can close CLion (any edits you made are automatically saved), log out, etc. Make sure you take your flash drive, if you used one!

Now, enjoy your weekend!