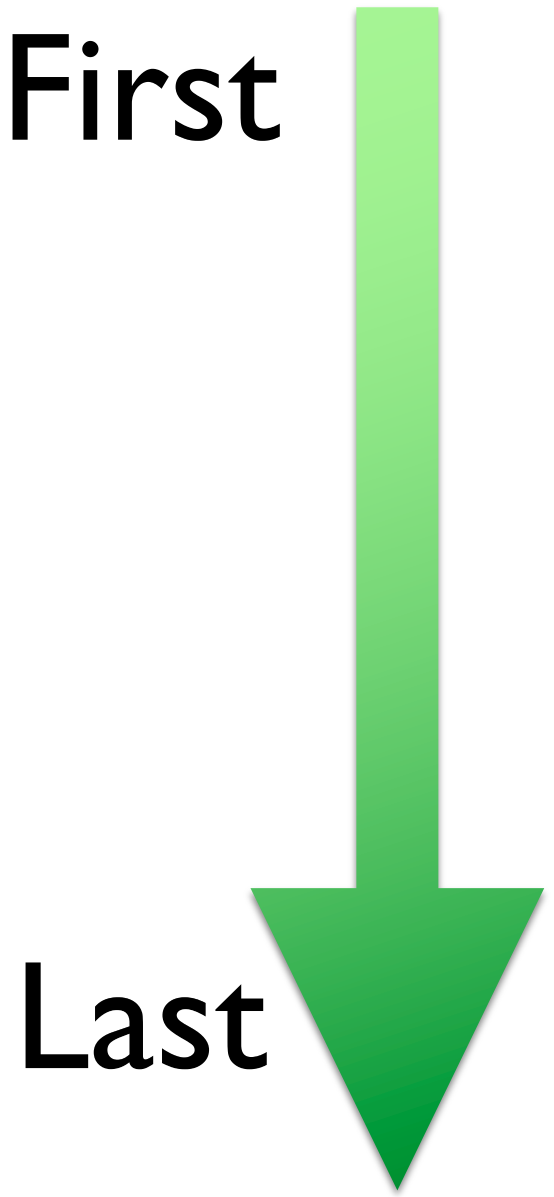
Lecture: Getting Started

ENGR:2730 Computers in Engineering

Classroom Laptops

- The laptops are not valuable outside of the classroom
  - no battery, no optical drive, no wireless
  - can’t boot from any device but hard drive
  - networking only works on classroom network
  - laptop cases are marked
- The classroom has electronic surveillance

Starting the laptop



- Pick up one laptop and one equipment bag
- Place laptop on top of computer bag
- Plug in power supply, network cable, mouse (optional), and USB flash drive (optional)
- Turn on laptop

Methods for using CLion

- VDI (Virtual Desktop Infrastructure)
  - VMware Horizon Client on your personal computer.
  - VMware Horizon Client on the classroom laptop.
  - Open web browser inside VDI and go to https://apps.engineering.uiowa.edu
  - Start the CLion application.

- CLion on the classroom laptop (not recommended)
  - Need to enter the class SVN path each time.
  - May lose all your work if lose power or you forget to commit your changes.

- Install CLion on your personal computer
  - If this does not work, use the VDI.

- To use SVN, you must enter the following URL into CLion.

https://class-svn.engineering.uiowa.edu/cie/projects/spring2024

VDI - Virtual Desktop Infrastructure

- VDI, Virtual Desktop Infrastructure, creates a secure connection between your computer and a virtual Windows machine that has the Engineering lab software load.
- Use VDI if you want to use CLion on a classroom laptop.

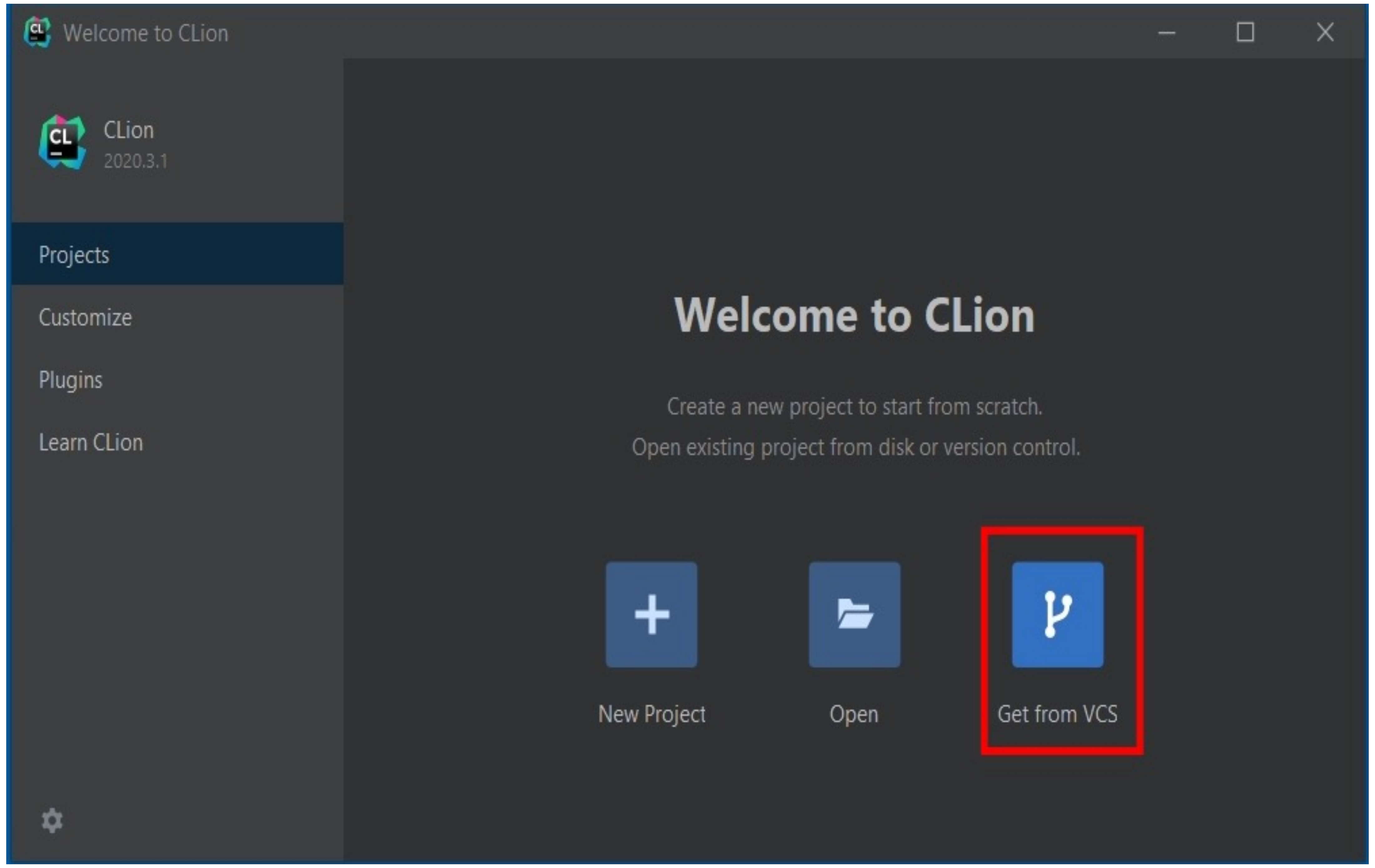
- Install VDI on your computer and login. https://engineering.uiowa.edu/etc/help-desk-computer-services/information/virtual-windows-desktop-vdi

Turning off the laptop

- Stop AppsAnyWhere Clion.
- Log off of VDI if logged in.
- Shutdown laptop.
- Once off, unplug network cable, power supply, USB mouse, and USB flash drive.
- Insert laptop with the “hinge end” first when putting it in bag.
- Place power supply and network cable in bag.
- Return laptop and equipment bag to cabinet.

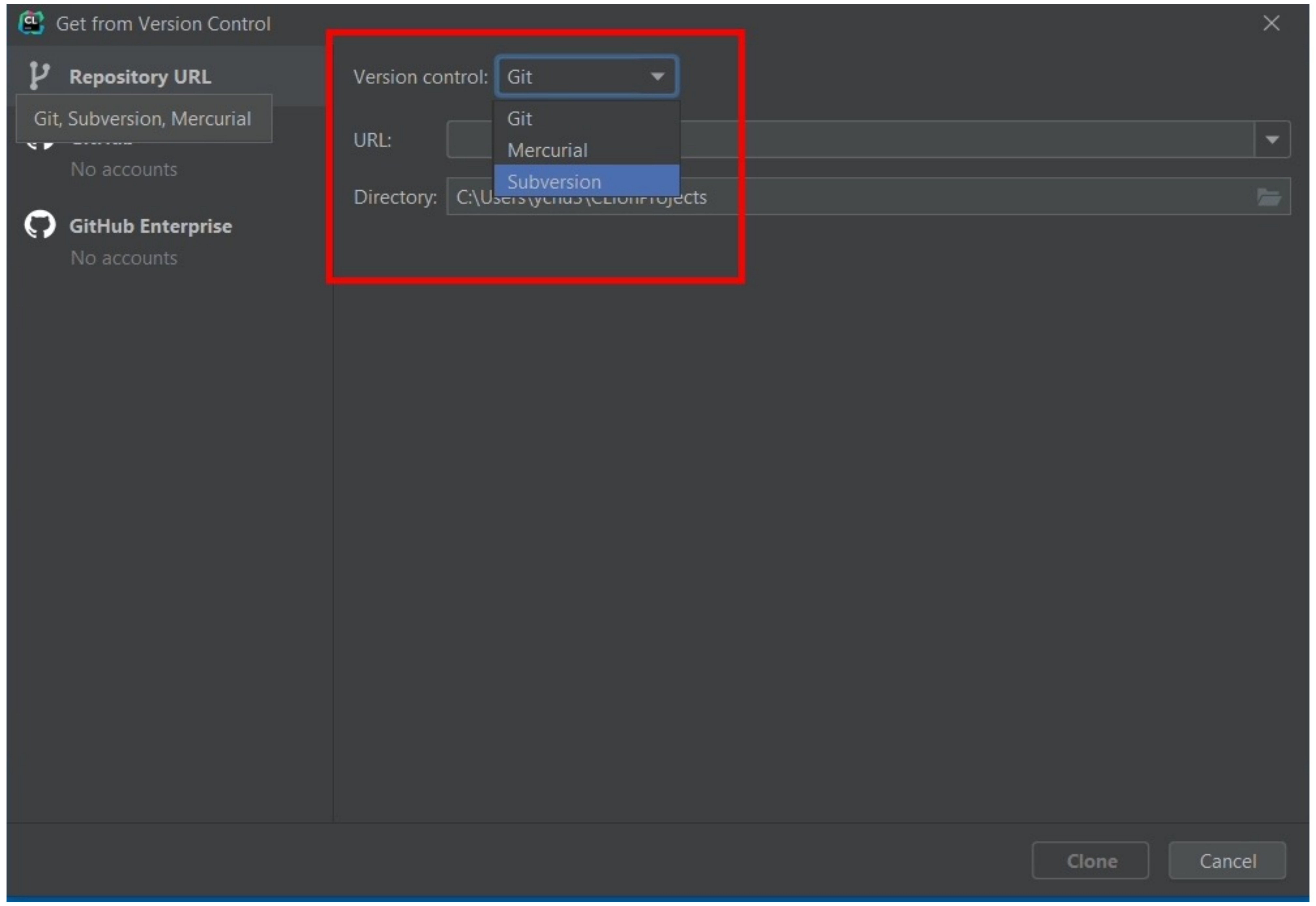
Initial CLion Setup for CIE

- Start CLion
- Click on “Get From VCS”, select “subversion”



Welcome to CLion

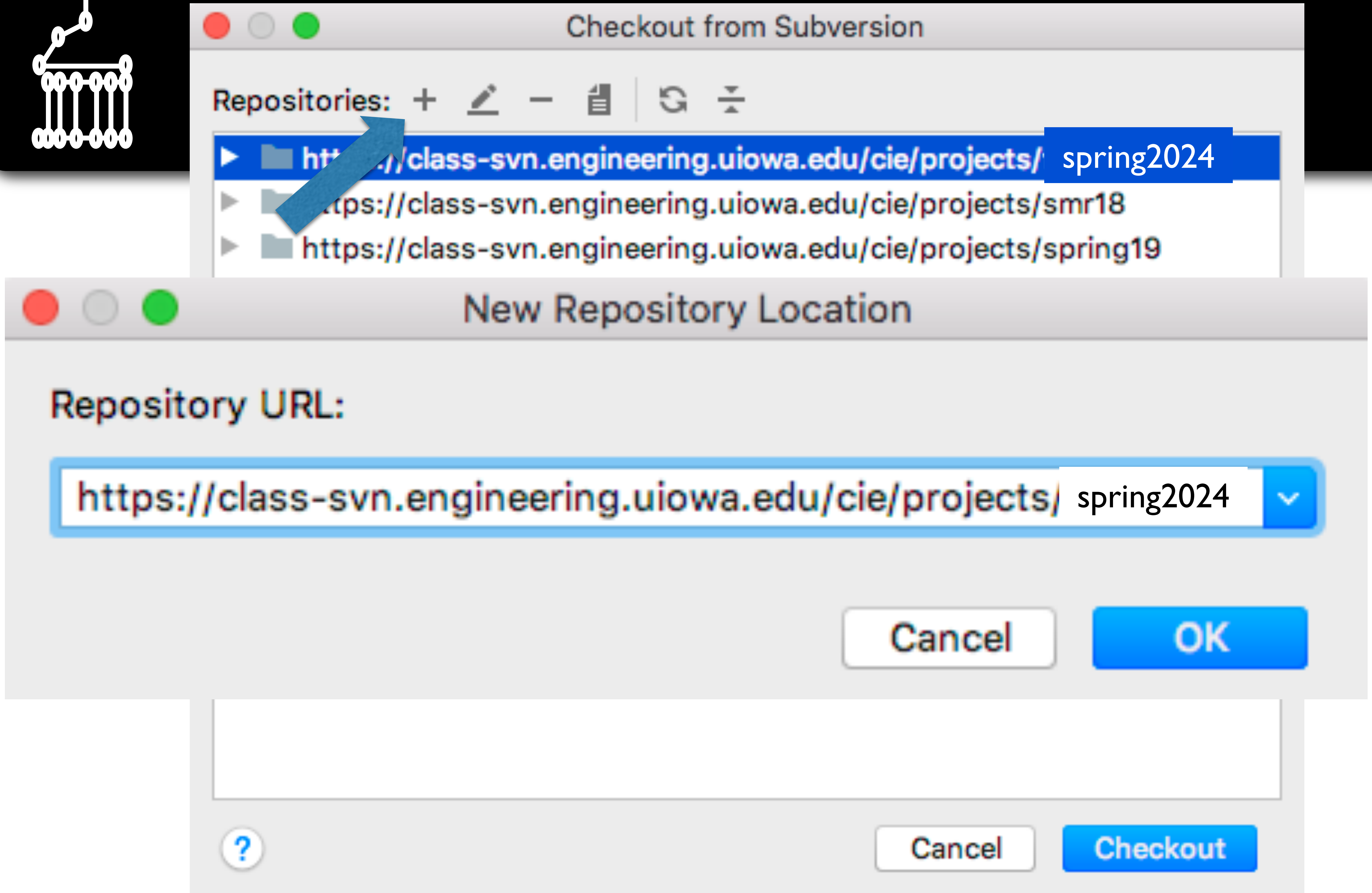
Create a new project to start from scratch.
Open existing project from disk or version control.



Version control: Git

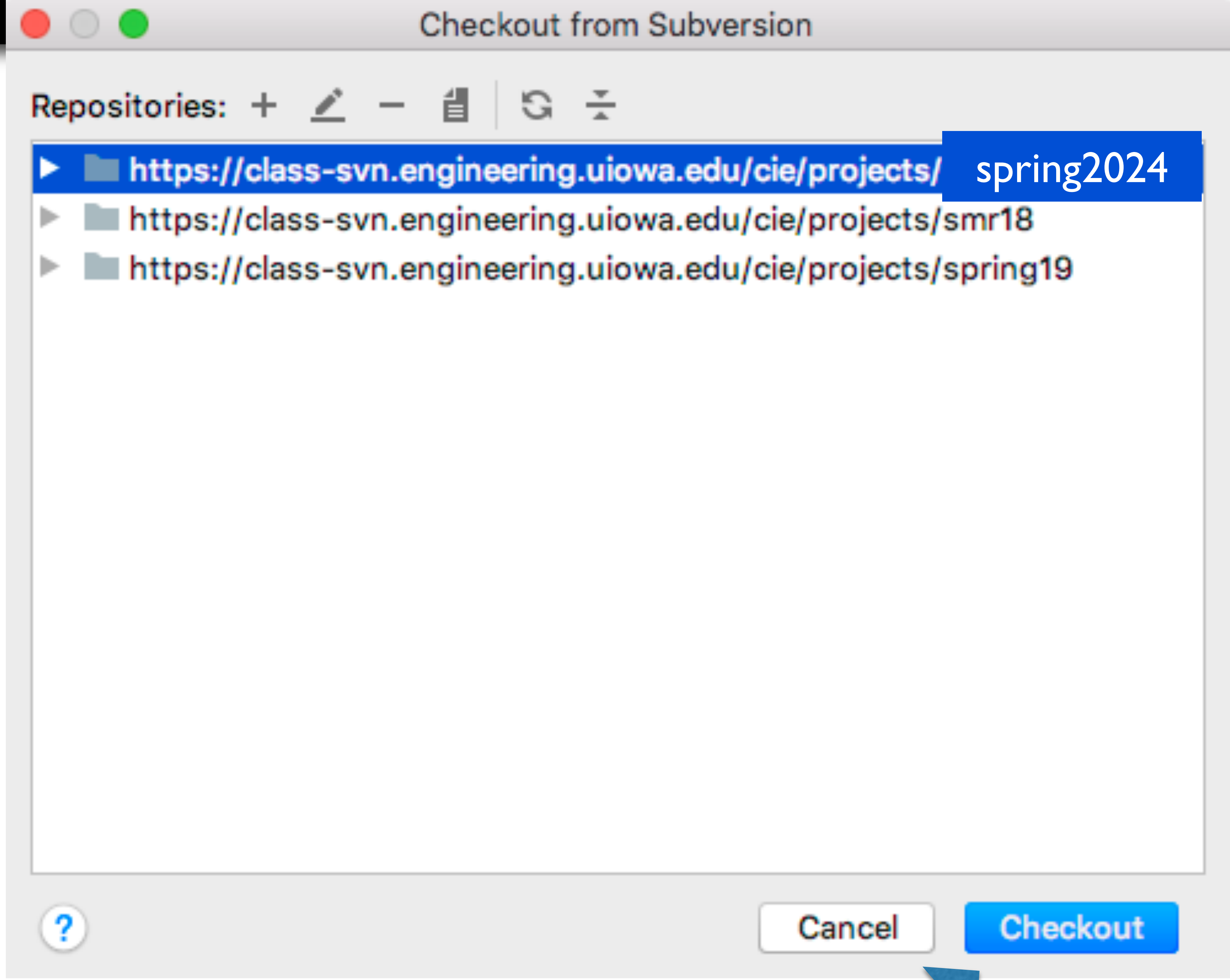
URL:

Directory: C:\Users\ychu3\CLionProjects



Repositories:
- https://class-svn.engineering.uiowa.edu/cie/projects/spring2024
- https://class-svn.engineering.uiowa.edu/cie/projects/smr18
- https://class-svn.engineering.uiowa.edu/cie/projects/spring19

Repository URL: https://class-svn.engineering.uiowa.edu/cie/projects/spring2024



Repositories:
- https://class-svn.engineering.uiowa.edu/cie/projects/spring2024
- https://class-svn.engineering.uiowa.edu/cie/projects/smr18
- https://class-svn.engineering.uiowa.edu/cie/projects/spring19

- Click the plus (+) to add a new SVN repository
- Enter the location https://class-svn.engineering.uiowa.edu/cie/projects/spring2024 and then click **OK**
- Highlight the new Repository listing and then click **Checkout**

engr2730fall19
Search
Favorites
Dropbox
iCloud Drive
Applications
Desktop
Documents
Downloads
Music
Pictures
gec
OneDriveBusiness
Devices
Remote Disc
Shared
AW_Atlas
ellum
lowa
pg
ılator
ece5480_f14
ece5550_s17
ece5820_f16
engr1300honorsf15
engr1300honorsf16
engr1300s15
engr1300s16
engr1300s17
engr1300s18
engr1300sum16
engr2730spring18
engr2730spring19
JavaScriptQuizExample
math5000
math5010
MEAN stack tutorial
mitLinearAlgebra
New Folder
Cancel
Open

engr2730spring2024

Remote Repository

Local Copy Location

SVN Checkout Options

Checkout: https://class-svn.engineering.uiowa.edu/cie/projects/ fall2022

Destination: /Users/gec /engr2730spring2024

/Users/gec/temp/engr2730fall19/cie/projects/ fall2022

/Users/gec/temp/e engr2730fall2022/fall2022

Update/Switch to revision: ● HEAD ○ Specified

Depth: infinity

☑ Include externals locations

- Go to your I: drive and create a new directory called engr2730spring2024.
- In the “SVN Checkout Options” dialog, ensure the Destination is set to your new folder
- Click **OK**

SVN Reading and Quiz

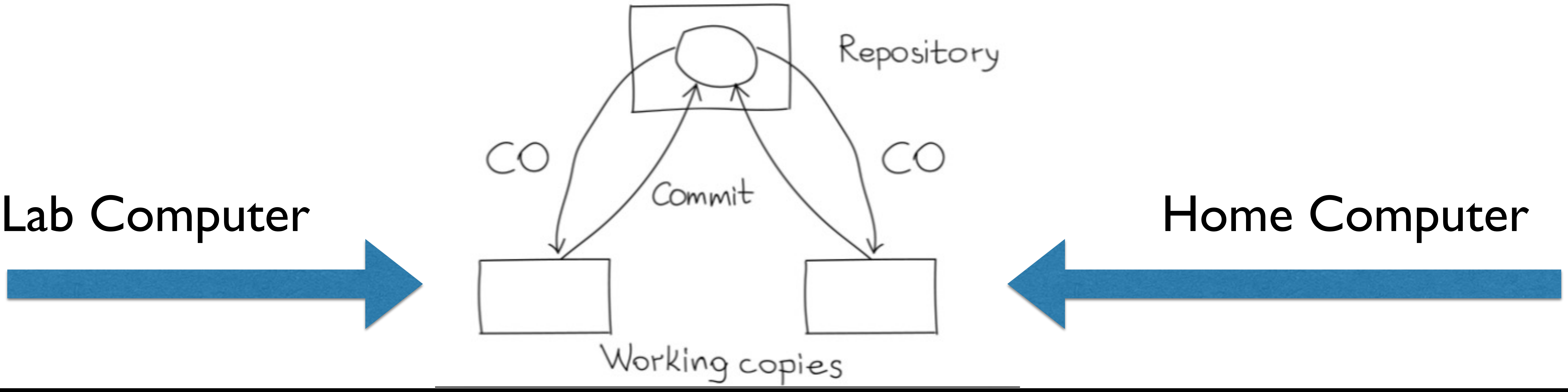
- Read the following sections of the Subversion (SVN) tutorial
  https://www.tutorialspoint.com/svn/svn_life_cycle.htm

  - SVN – Basic Concepts
  - SVN – Life Cycle
  - SVN - Checkout Process
  - SVN - Perform Changes
  - SVN - Review Changes
  - SVN - Update Process
  - SVN - Fix Mistakes
  - SVN - Resolve Conflicts
  - Watch videos posted on ICON
    - How to create a project using CLion.
    - How to add a project to SVN and ignore cmake-build-debug and .idea directories.

- **Take SVN Quiz on ICON**

New content in SVN often

- We will be adding new content to your repositories or to a Public repository before some lectures
- You will need to synchronize your local “working” directory before each use of the local repository. You will then commit your local work to the remote svn when finished.
- **Keeping everything in sync is very important.**



Lab Computer

Home Computer

Basic subversion (svn) summary

One time only

checkout

[modify]

commit

Every working session

once you have an initial checkout

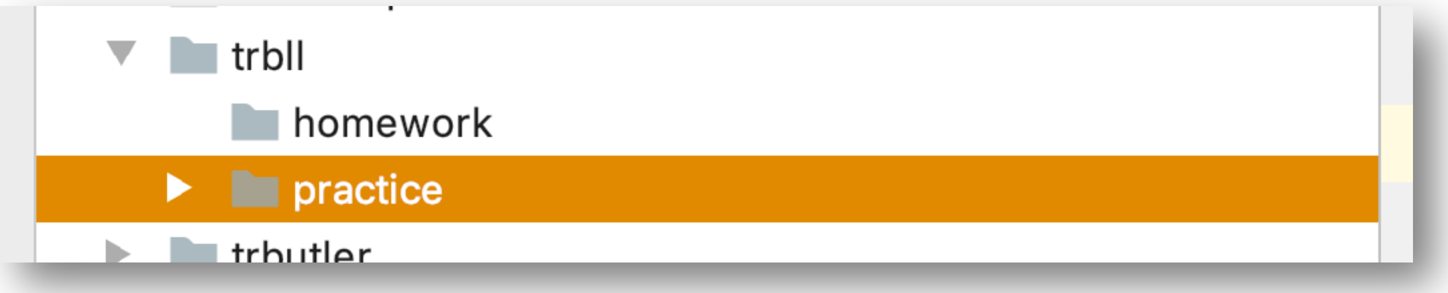
update

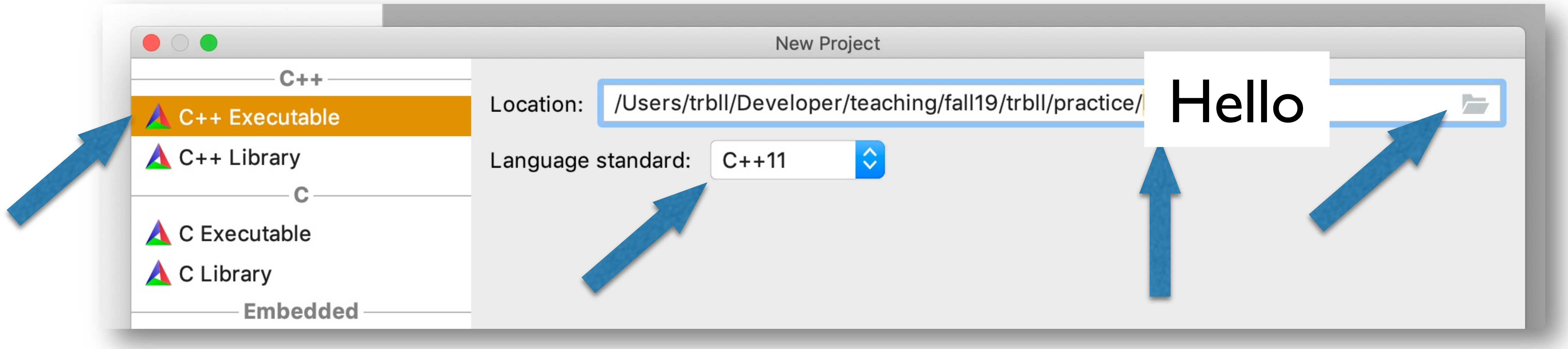
[modify]

commit

Hello World Program

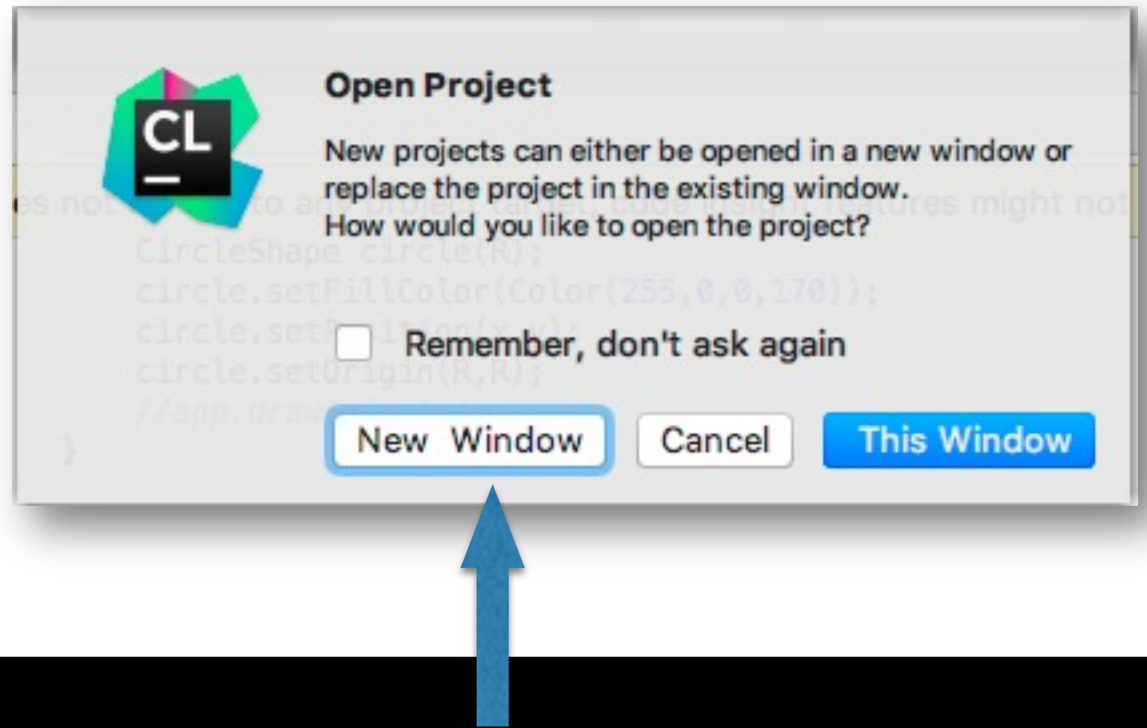
- Check out your class directory from SVN
- In left navbar, expand your hawkId directory. You should see two empty folders: **homework** and **practice**
- In top navbar, select *File > New Project…* This pops up a dialog window.
- In the left navbar, make sure “C++ Executable” is selected
- In “Location”, navigate to your practice directory and create a **Hello** directory. Click **Create**.
- Click “New Window” in the dialog that appears and then close the new window that appears.





Location: /Users/trbll/Developer/teaching/fall19/trbll/practice/ Hello

Language standard: C++11

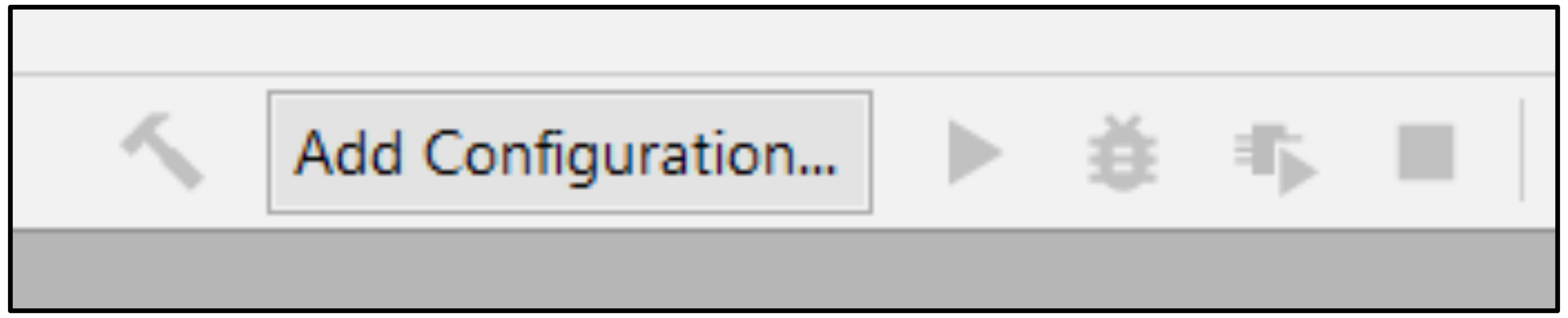

Open Project

New projects can either be opened in a new window or replace the project in the existing window. How would you like to open the project?

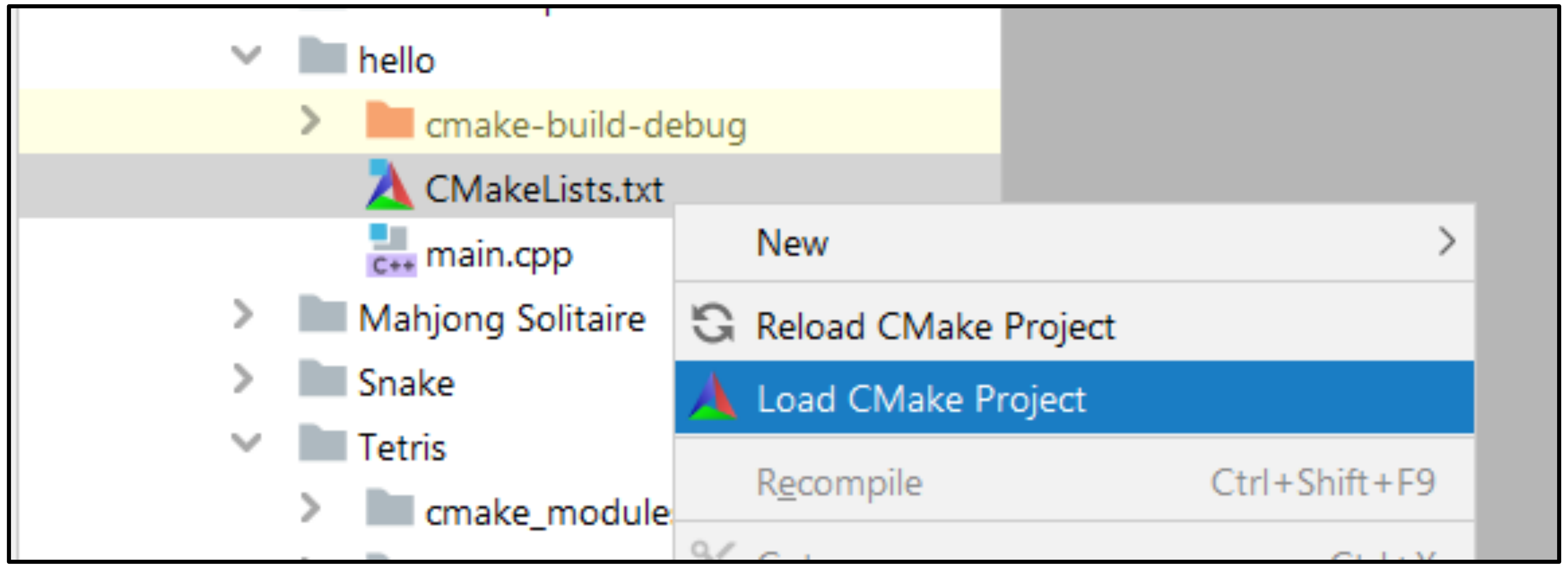
□ Remember, don't ask again

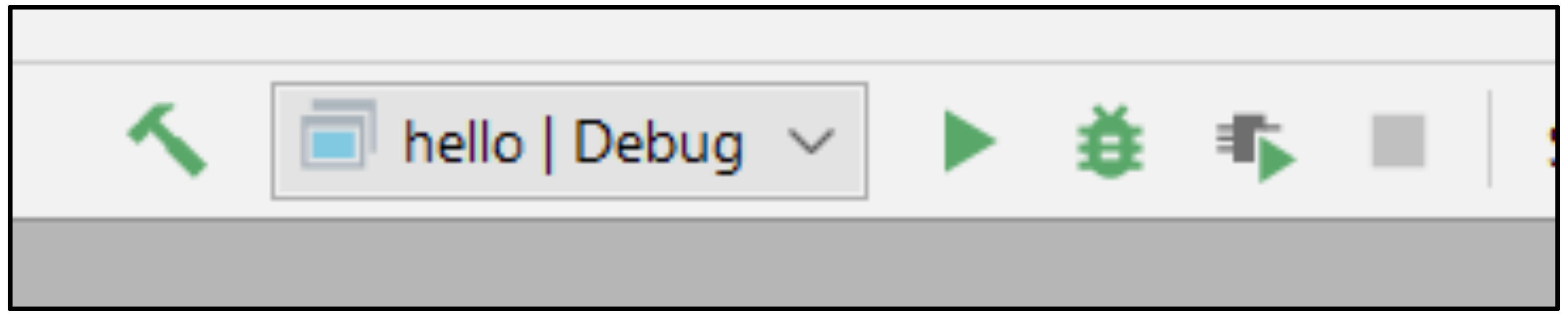
Load a Project in CLion

- If you see “Add Configuration…” there is no active project in CLion

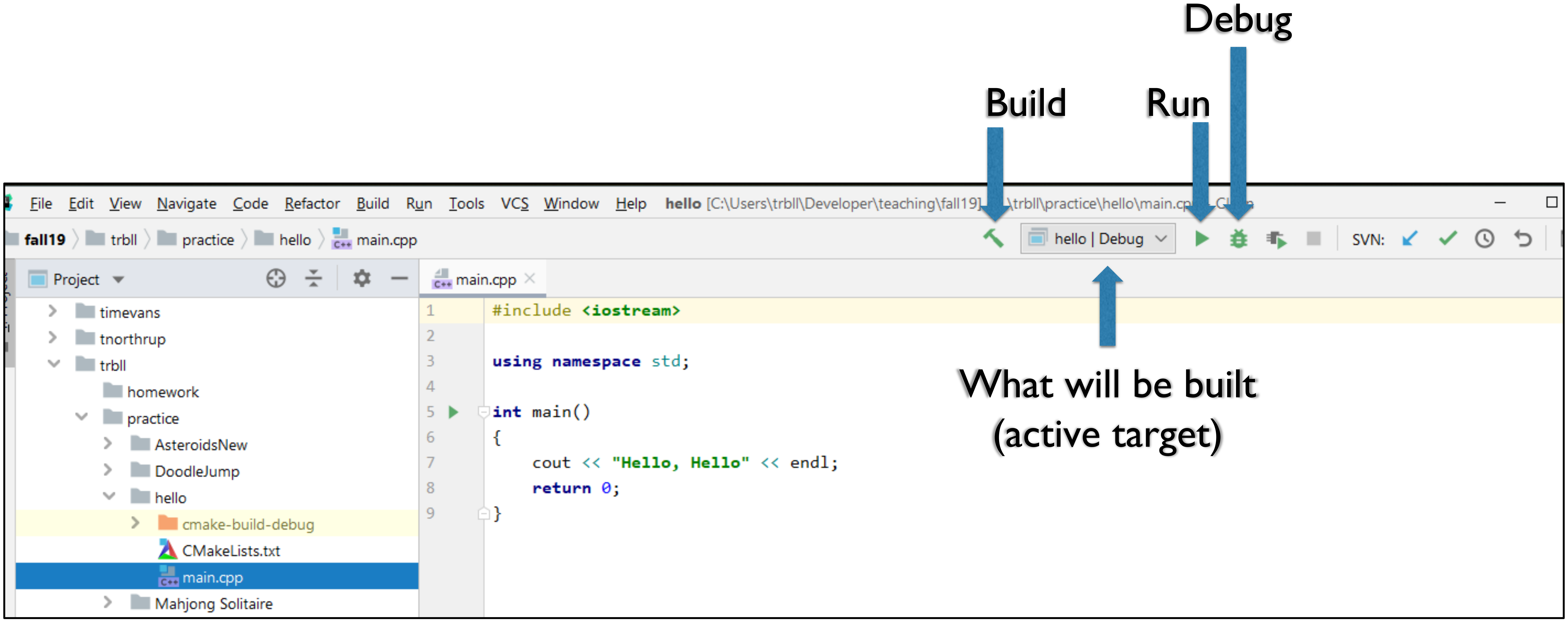


- To load a project: in the project’s folder, right click the “CMakeLists.txt” file and select “Load CMake Project”





Try building/running the starting program



Debug

Build

Run

What will be built (active target)

```cpp
#include <iostream>

using namespace std;

int main()
{
    cout << "Hello, Hello" << endl;
    return 0;
}
```

CMake

	A Simple Project CMakeLists.txt file
1	cmake_minimum_required(VERSION 3.14)
2	project(Hello)
3	
4	set(CMAKE_CXX_STANDARD 11)
5	
6	add_executable(hello main.cpp)

CMake tutorial:
http://derekmolloy.ie/hello-world-introductions-to-cmake/

- Line 1 sets the minimum version of CMake for this project
- Line 2 is the *project()* command that sets the project name.
- Line 4 is the minimum version of C++ needed to build the project.
- Line 6 is the *add_executable()* command, which requests that an executable be built using the main.cpp source file. The first argument to the *add_executable()* function is the name of the executable to be built, and the second argument is the source file from which to build the executable.

SVN: One-Time Setup per Machine

- The cmake-build-debug and .idea directories are automatically created to compile a C++ program. These directories contain the build files and should never be committed to SVN. You need to do the following for each computer that you use CLion, i.e., the VDI CLion and your personal computer:

- Windows: In CLion, click Files->Settings...->Version Control->Subversion. Check the option "Use custom configuration directory:"
  Mac: In CLion, click CLion->Preferences...->Version Control->Subversion. Check the option "Use custom configuration directory:"

- Open the file in the "Use custom configuration directory:" text box in CLion (see instructions below).

- Find the line: "# global-ignores = *.o *.lo *.la *.al .libs *.so *.so.[0-9]* *.a *.pyc *.pyo __pycache__"

- Change this line to "global-ignores = *.o *.lo *.la *.al .libs *.so *.so.[0-9]* *.a *.pyc *.pyo __pycache__ cmake-build-debug .idea .vs .vscode" and save.

- From now on, adding any project file to SVN such as the CMakeLists.txt and/or main.cpp will automatically run the custom configuration file and ignore the cmake-build-debug and .idea directory.

Editing Text Files

To edit the config file on a Mac

- In CLion: Click File->Open
- On keyboard press the command + shift + period keys to show the hidden dot files.
- Navigate to the config file and open it.
- When finished, press the command + shift + period keys to hide the hidden dot files.

To edit the config file on Windows

- Copy the full path of the file from the text box in step 2 above.
- In CLion: Click File->Open
- Paste full path to file in the file to open text box.
- Edit and save the file.

Committing your Hello World Program to SVN

- You should have a **hello** folder under your practice folder.

- Notice that everything is in red. Red means that the file is not in SVN.

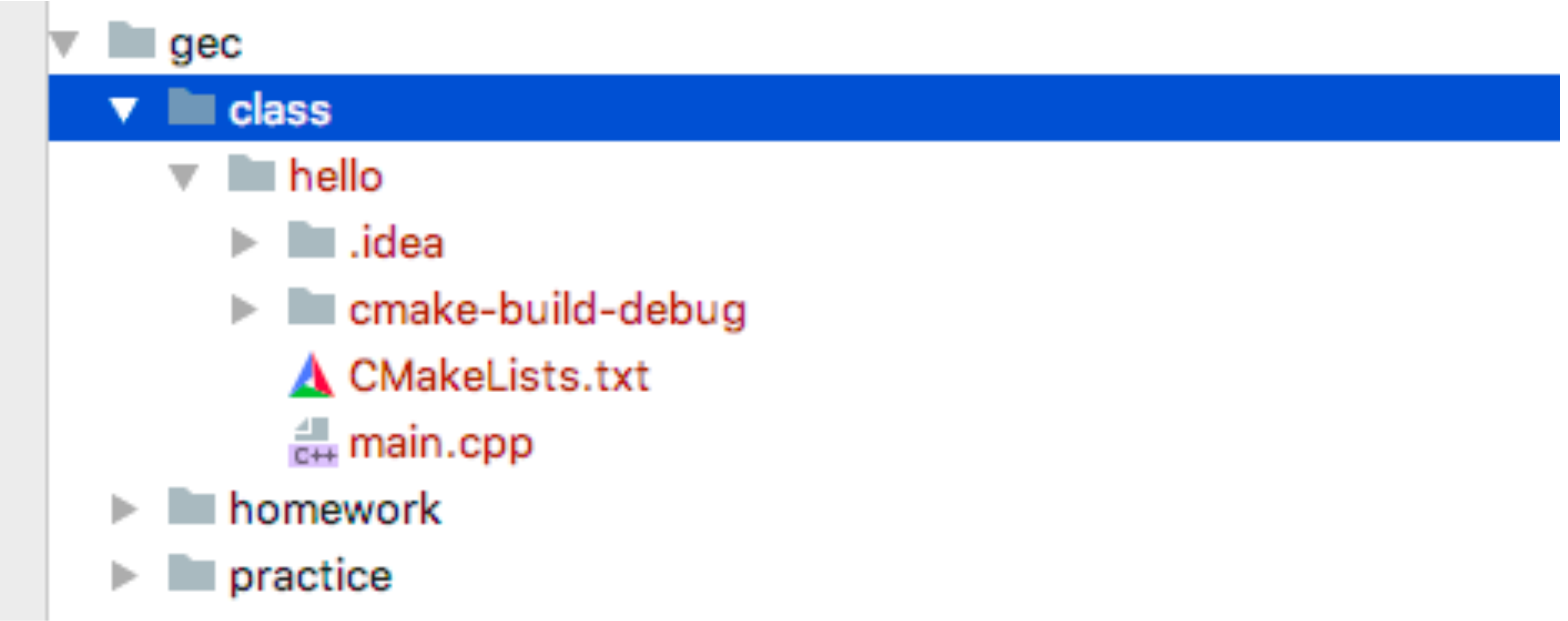
- Right click on the *main.cpp* and select *Subversion > Add to VCS.* The main.cpp and the hello directories should turn to green.

- Right click on the *CMakeLists.txt* and select *Subversion > Add to VCS.* The CMakeLists.txt file should turn to green.

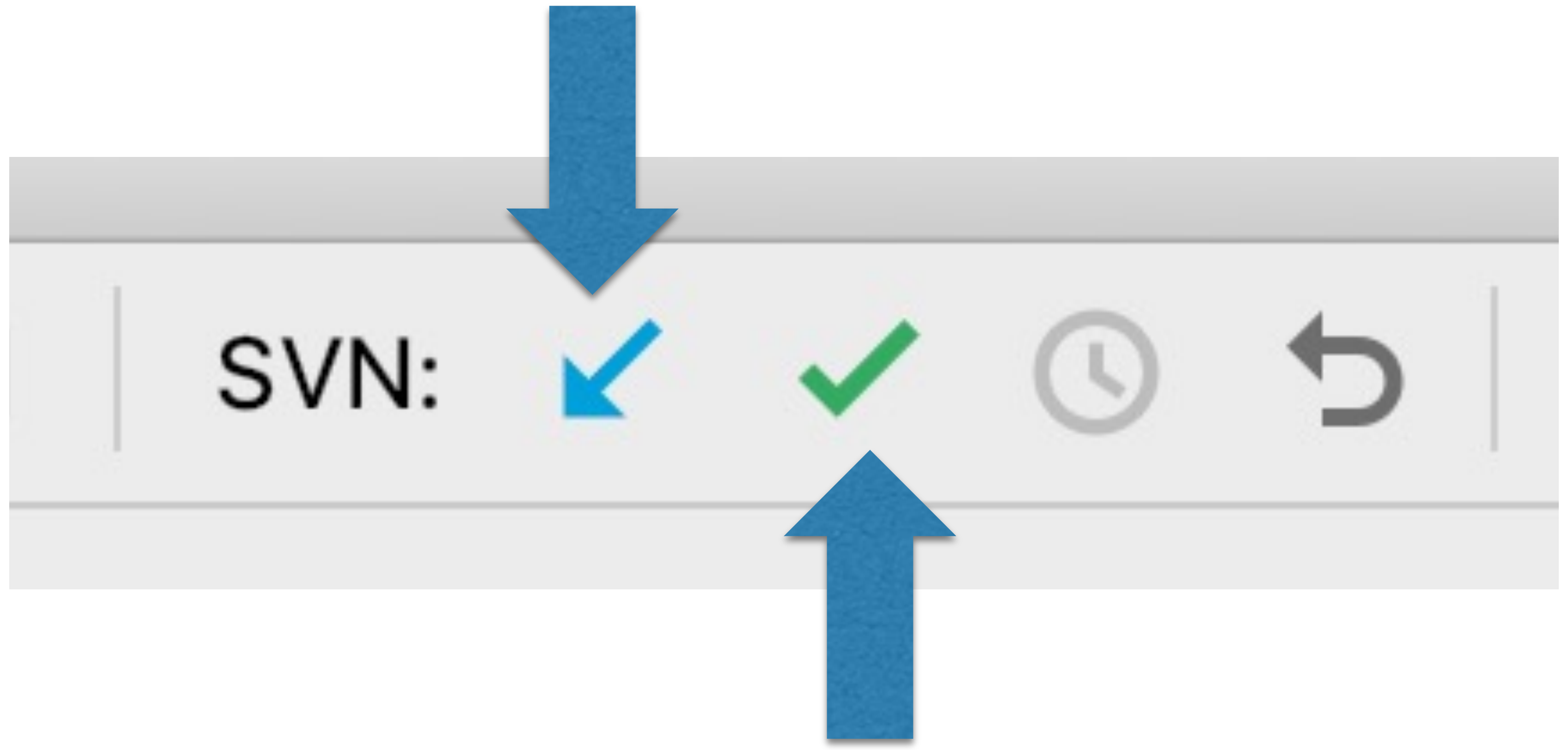
- **DO NOT ADD THE .IDEA OR CMAKE-BULID-DEBUG DIRECORTIES TO SVN. YOU WILL SCREW UP SVN FOR YOURSELF. If you properly did the one-time setup for SVN, these directories should turn tan once you add a file to be committed to SVN.**

- If you make a mistake and accidently add a file that should not be added to SVN, right click on the file and select *Subversion > Revert…*

- When you are ready to commit your added files, click on the commit button as shown in the screenshot.



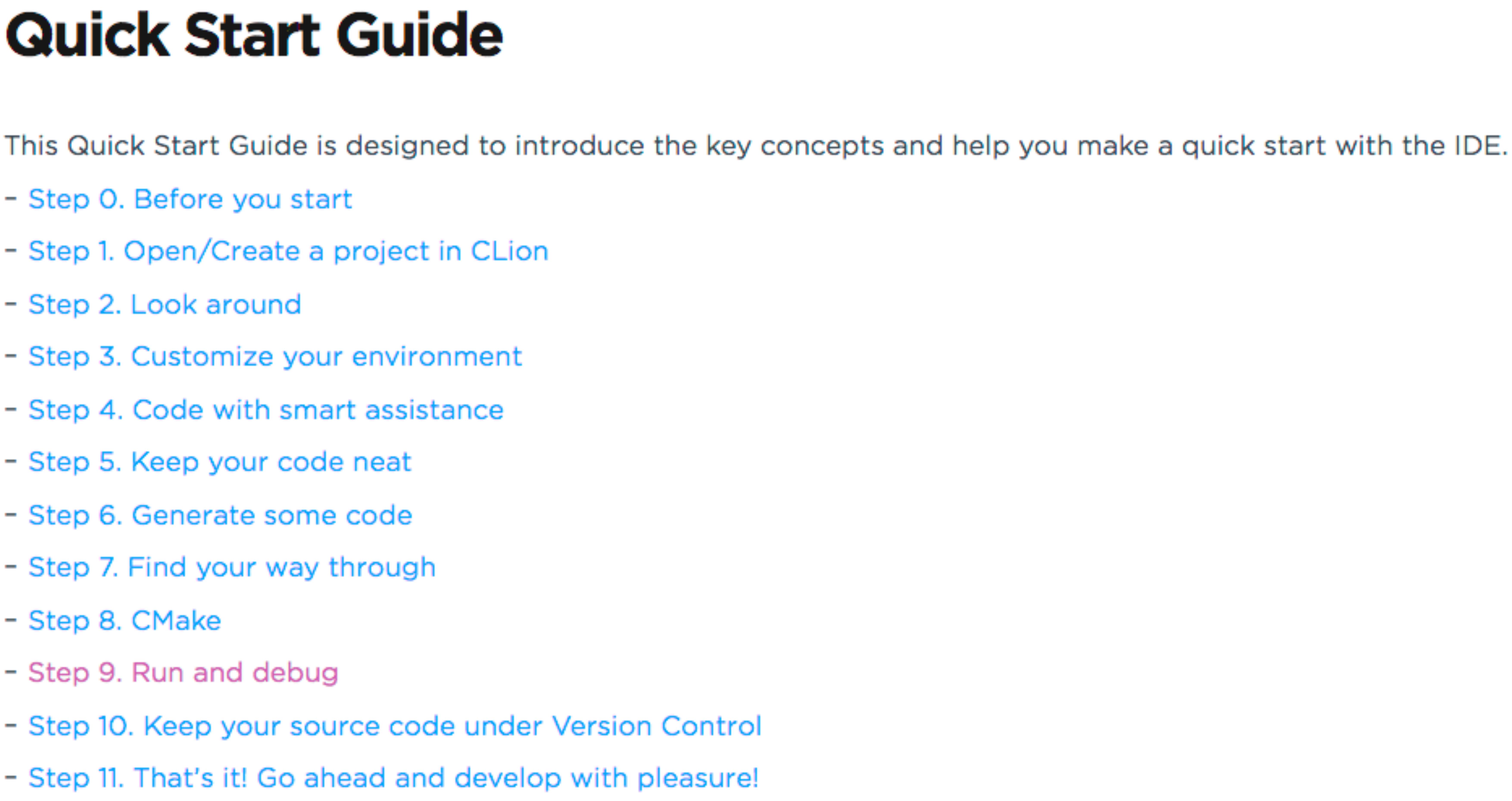
Click to update



Click to commit

Quick Start Guide for CLion IDE

- https://www.jetbrains.com/help/clion/clion-quick-start-guide.html



Quick Start Guide

This Quick Start Guide is designed to introduce the key concepts and help you make a quick start with the IDE.

- Step 0. Before you start
- Step 1. Open/Create a project in CLion
- Step 2. Look around
- Step 3. Customize your environment
- Step 4. Code with smart assistance
- Step 5. Keep your code neat
- Step 6. Generate some code
- Step 7. Find your way through
- Step 8. CMake
- Step 9. Run and debug
- Step 10. Keep your source code under Version Control
- Step 11. That's it! Go ahead and develop with pleasure!