

COP 3014: Fall 2018

A Guide to Using CLion

August 27, 2018

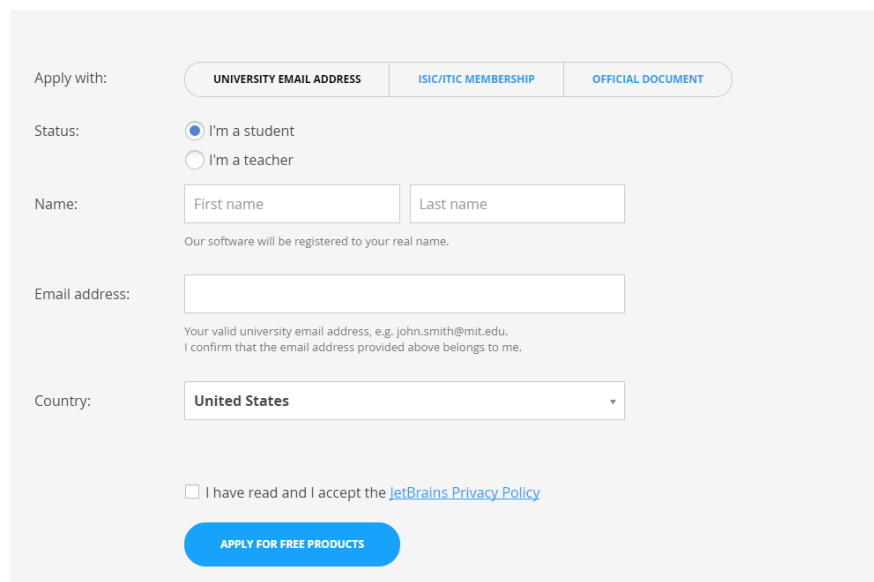
1 CLion

CLion is a comprehensive C++ Integrated Development Environment developed and maintained by JetBrains. It is very intuitive and easy to use. For this class, we will use CLion for all of our work.

1.1 Applying for a Student License

In order to use CLion, you need to apply for a license. Go to <https://www.jetbrains.com/student/>. Click on the “Apply” button. You should see the following screen:

JetBrains Products for Learning



The screenshot shows a web form titled "JetBrains Products for Learning". At the top, there are three tabs: "UNIVERSITY EMAIL ADDRESS" (selected), "ISIC/ITIC MEMBERSHIP", and "OFFICIAL DOCUMENT". Below the tabs, the form includes the following fields and options:

- Apply with:** Three tabs: "UNIVERSITY EMAIL ADDRESS" (selected), "ISIC/ITIC MEMBERSHIP", and "OFFICIAL DOCUMENT".
- Status:** Two radio buttons: "I'm a student" (selected) and "I'm a teacher".
- Name:** Two text input fields: "First name" and "Last name". Below these fields is the text: "Our software will be registered to your real name."
- Email address:** A text input field. Below it is the text: "Your valid university email address, e.g. john.smith@mit.edu. I confirm that the email address provided above belongs to me."
- Country:** A dropdown menu with "United States" selected.
- I have read and I accept the [JetBrains Privacy Policy](#).
- APPLY FOR FREE PRODUCTS** (blue button)

© 2013-2018 JetBrains | [Help](#) | [Support](#) | [Privacy Policy](#) | [Terms Of Use](#) | Build #1134

Figure 1: Applying for a Student License

Please provide an FSU email. You will get a reply from JetBrains in a few minutes, asking you to confirm the request for a student license.

Once you confirm the request, they will send you another email, asking you to create a JetBrains Student Account. This takes about 5 minutes. Once you create the account, you should see a page with your Key.

2 Installing CLion

You can now download and install CLion. Download CLion from <https://www.jetbrains.com/clion/download>. The page will automatically show you the right installer for your Operating System. Download the installer.

2.1 For Macs/Linux Machines

If you have a Mac or a Linux Machine, you already have a lot of the prerequisite programs needed for CLion. All you need to do is a couple of simple steps.

2.1.1 Mac OSX

Once you have downloaded the dmg file, use Finder to open the Applications folder. Move the file you just downloaded from the Downloads folder into the Applications folder. You're all set.

2.1.2 Linux Machines

Download the tar from the website. Unpack it and then navigate to the `bin` directory and run `CLion.sh`

2.2 For Windows Machines

Download the installer from the website and run it. It will start up the installation wizard.

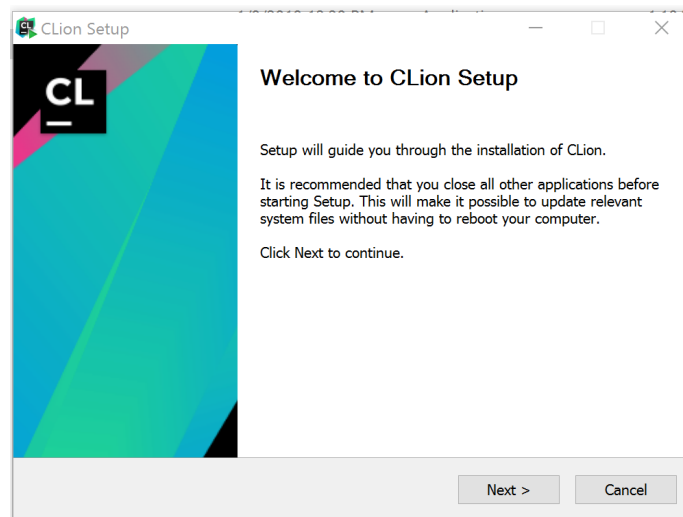


Figure 2: Installation Wizard

Click on Next. It will ask you to choose a location to install CLion. Go with the default option.

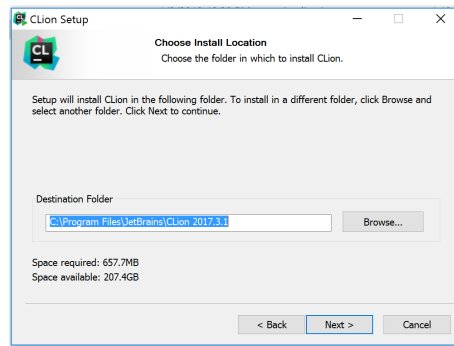


Figure 3: Installation Location

Click on Next. It will ask you for a temporary folder to put the temporary files. Go with the Default Option. Click on Next. It will ask you to choose file associations. Check all the boxes.

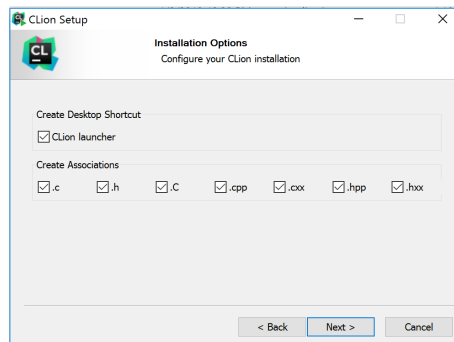


Figure 4: Choosing File Associations

Click on Next. This will complete the setup. It will ask if you want to run CLion. Say Yes.

3 Running CLion for the first time

Once you have installed CLion, you can run it. There is a bit of setup to do when we run CLion for the very first time. This is mostly the same on all platforms. For Windows, there is an extra setup to get a Unix based C++ compiler installed, but the CLion Wizard will guide you through it.

When we first run CLion, it will ask if we would like to import settings. You will see the following dialog box.

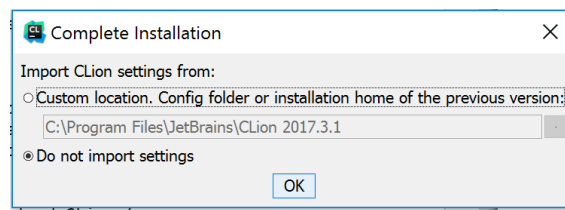


Figure 5: Importing Setting

Choose “Do Not Import Settings”. Click OK. It will now ask you to activate your CLion License. Put in the Username and Password for the JetBrains account you created. Click on Activate

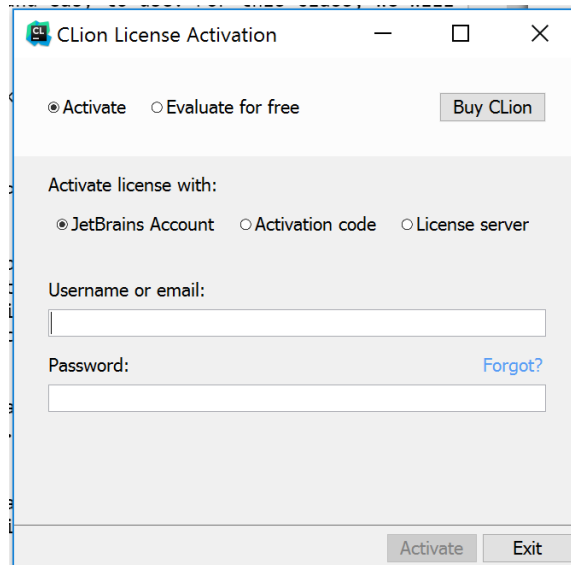


Figure 6: Activate your license

It will ask you to choose a theme. Pick one that you like.

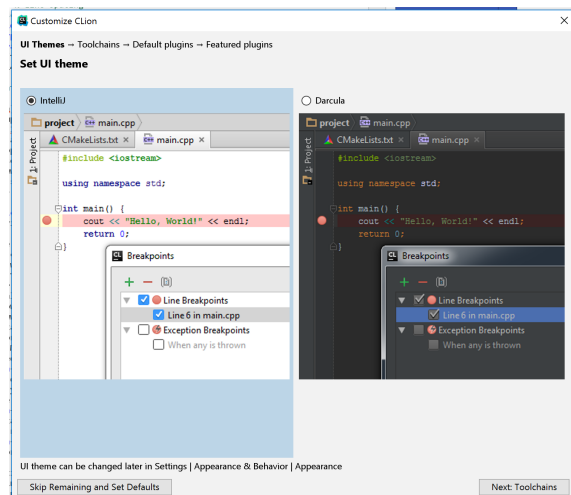


Figure 7: Choose a Theme

Click on Next: Toolchains. This is where the paths diverge for Mac and Windows machines.

3.1 For Macs / Linux Machines

Macs and Linux Machines most likely already have working installations of the C++ compiler, the make utility and a version of the GDB debugger. CLion should be able to automatically detect these. You should see a screen that looks like this:

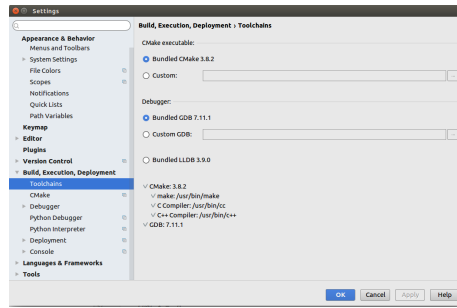


Figure 8: Toolchains

Click on OK. Your setup is now complete.

3.2 For Windows Machines

Windows machines do not come with an installation of the required programs, so you should see a screen that looks like this:

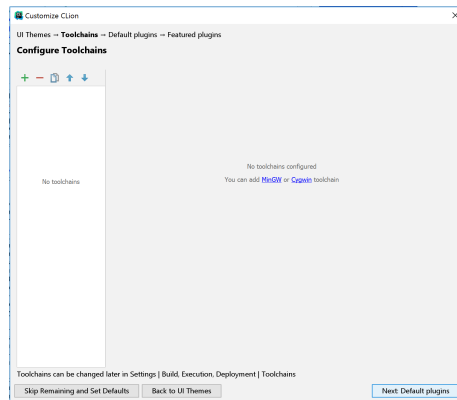


Figure 9: Toolchains

Click on the link that says Cygwin. You should now see this:

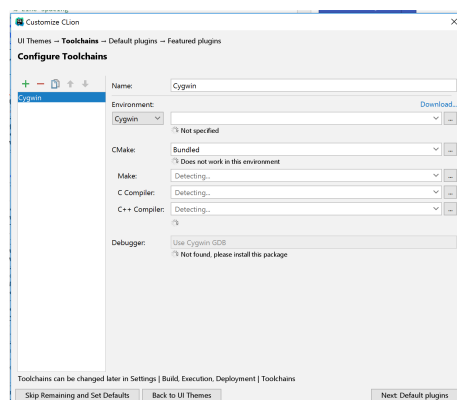


Figure 10: Toolchains

Click on Download. It will open the following screen on your web browser.



Figure 11: Installing Cygwin

Download the installer and run it. This might require Admin privileges.

- It will ask you for installation options. Choose “Install from the Internet”.
- It will ask you to choose a directory for installation. Go with the default option. It will also ask you to choose users. Choose “All Uses”.
- It will ask you where to store the Downloaded files. Go with the default option.
- On the next screen, choose “Proxy Settings”.
- The next screen will show you a list of “mirrors”. You can pick any of them. Click on Next. You should see the following screen

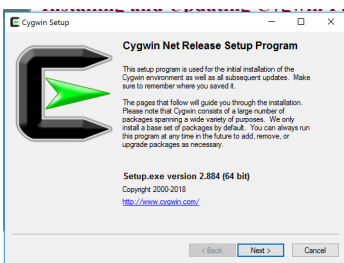


Figure 12: Installing Cygwin

Click Next. It will ask you to choose which packages to install. To keep things simple, we’re going to install the 7 required packages for C++.

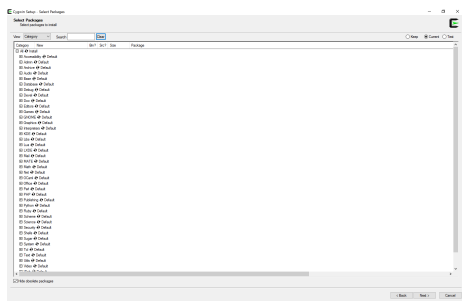


Figure 13: Installing Cygwin

- Go to the Drop-down box on the top left and choose “Full”.

- Search for `gcc-core`. You will see the following screen. Click on the circular arrow next to “Skip” on the line for `gcc-core: GNU Compiler Collection (C, OpenMP)`. Keep clicking until you see a number like 6.4. Sometimes, you might get multiple version numbers. Pick the latest (largest) version number.

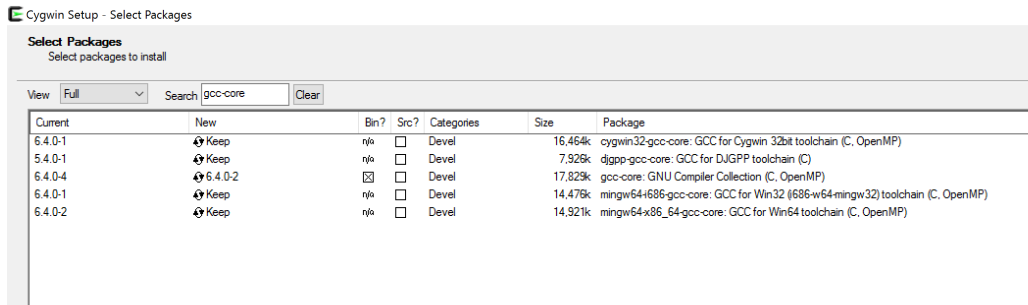


Figure 14: Get gcc-core

- Clear the search bar. Search for `gcc-g++`. You will see the following screen. Click on the circular arrow next to “Skip” on the line for `gcc-g++: GNU Compiler Collection (C++)`. Keep clicking until you see a number like 6.4. Sometimes, you might get multiple version numbers. Pick the latest (largest) version number.

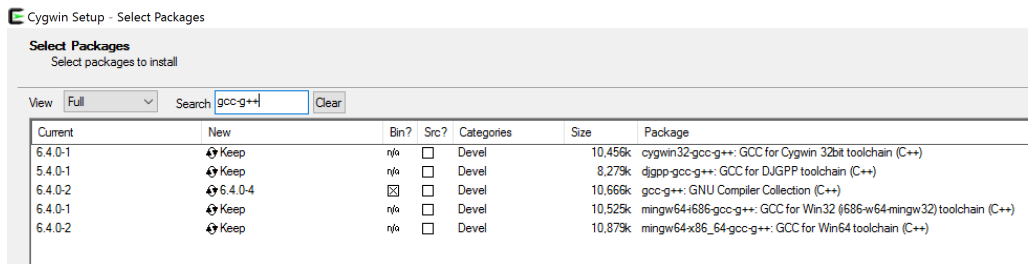


Figure 15: Get gcc-g++

- Clear the search bar. Search for `libgcc`. You will see the following screen. Click on the circular arrow next to “Skip” on the line for `libgcc1: GCC C runtime library`. Keep clicking until you see a number like 6.4. Sometimes, you might get multiple version numbers. Pick the latest (largest) version number.

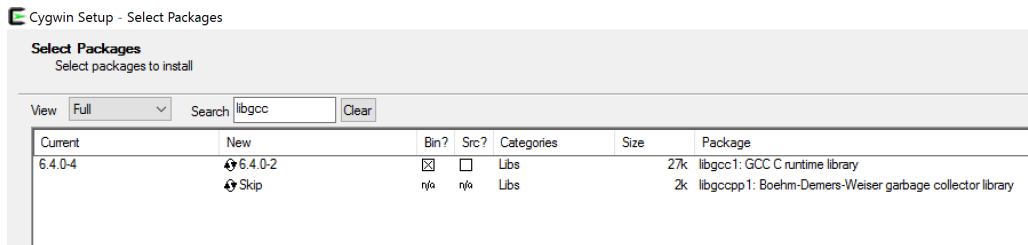


Figure 16: Get libgcc

- Clear the search bar. Search for `make`. You will see the following screen. Click on the circular arrow next to “Skip” on the line for `cmake: Cross Platform makefile generation system`. Keep clicking until you see a number like 3.3.

- Scroll down a little bit until you see make: The GNU version of the make utility. Click on the circular arrow next to “Skip” on this line until you see a number like 4.1.

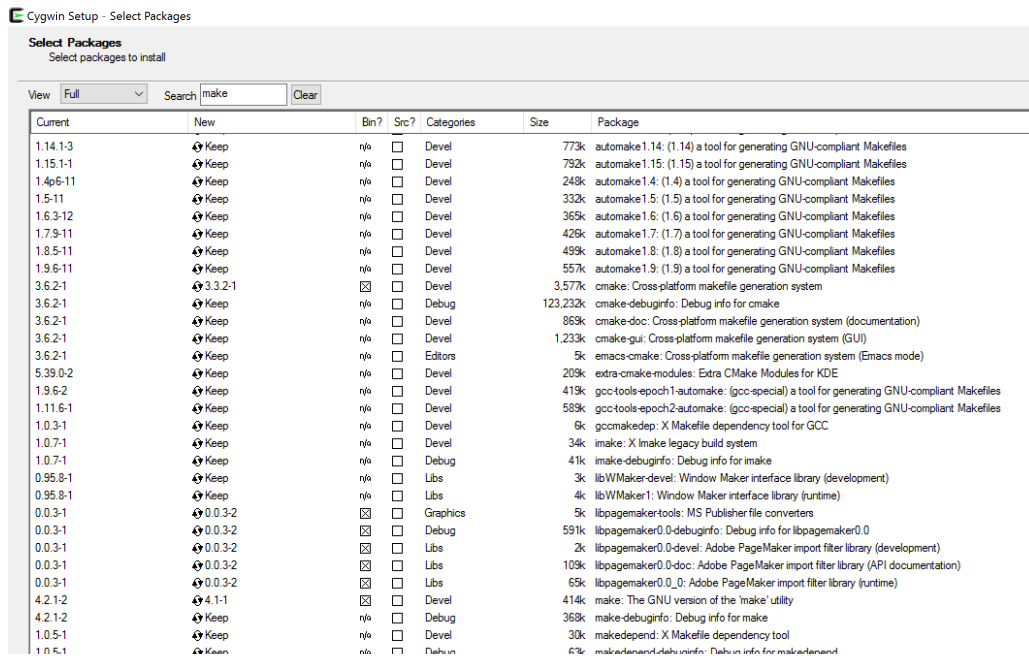


Figure 17: Get cmake and make

- Clear the search bar. Search for gdb. You will see the following screen. Click on the circular arrow next to “Skip” on the line for gdb: The GNU debugger). Keep clicking until you see a number like 7.10.

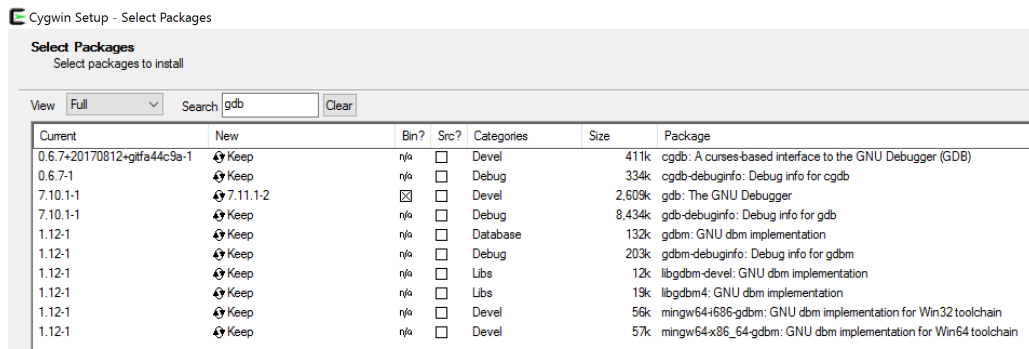


Figure 18: Get gdb

- Clear the search bar. Search for libx11. You will see the following screen. Click on the circular arrow next to “Skip” on the line for libx11-devel: XOrg X11 core library (development). Keep clicking until you see a number like 1.6.

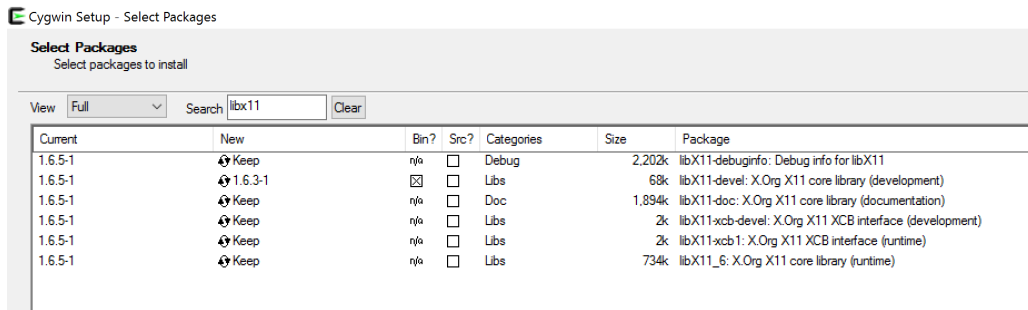


Figure 19: Get libx11

- Clear the search bar. Search for **coreutils**. You will see the following screen. Click on the circular arrow next to “Skip” on the line for coreutils: GNU core utilities (includes fileutils, sh-utils and textutils). Keep clicking until you see a number like 8.25.

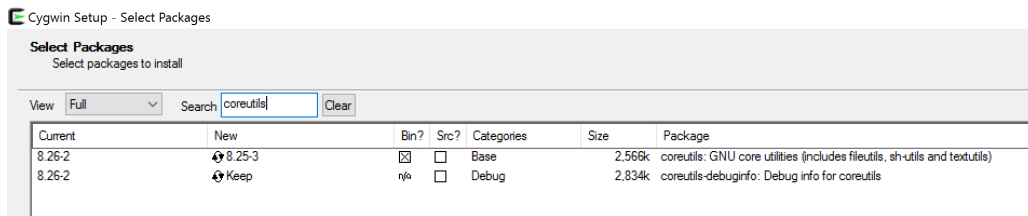


Figure 20: Get coreutils

- Clear the search bar. Then click Next. You will see the following screen where CygWin will show you a list of dependencies. Just click Next. CygWin will now install the new packages we have specified. This should not take too long, only about 10 minutes. On the final screen of the CygWin installation wizard, click Finish.

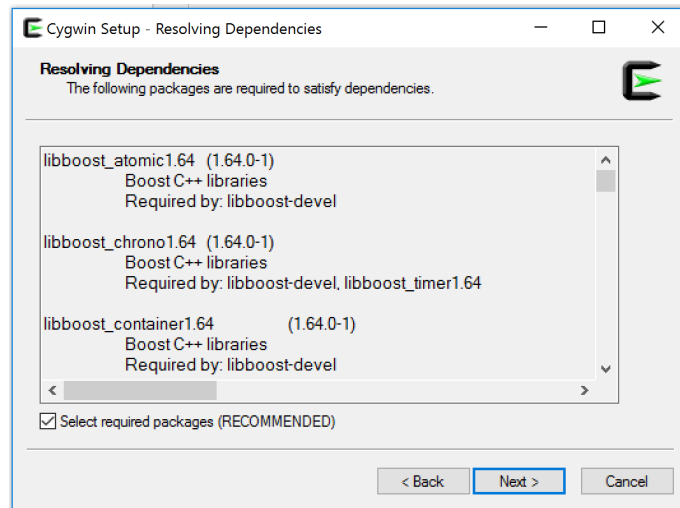


Figure 21: Dependencies

- Back in CLion, the software will now try and detect the new libraries. After about a minutes, it should have automatically detected all the libraries. Click OK.

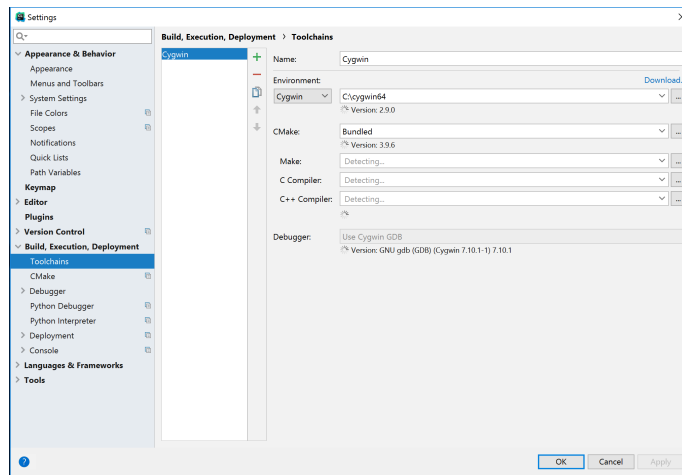


Figure 22: CLion detecting CygWin Libraries

4 Creating a Project

In order to write and run a C++ program on CLion, you should first create a project. Open CLion and then choose File ->New Project.

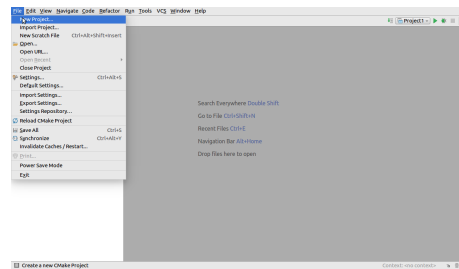


Figure 23: Opening a New Project

You will now see a screen where it asks you to choose where to save the project. Also, the project is called “untitled” by default. Change it to a name of your preference.

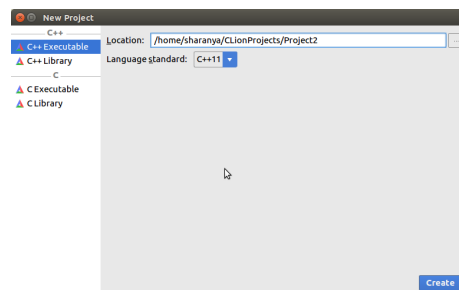


Figure 24: New Project Name and Location

Click on Create. This will open a new project with a default program stub. This program is called “main.cpp”. We will usually require to to give specific names to your programs. This requires

you to refactor the project. Right Click on “main.cpp” in the Project Explorer Window on the left and choose Refactor ->Rename.

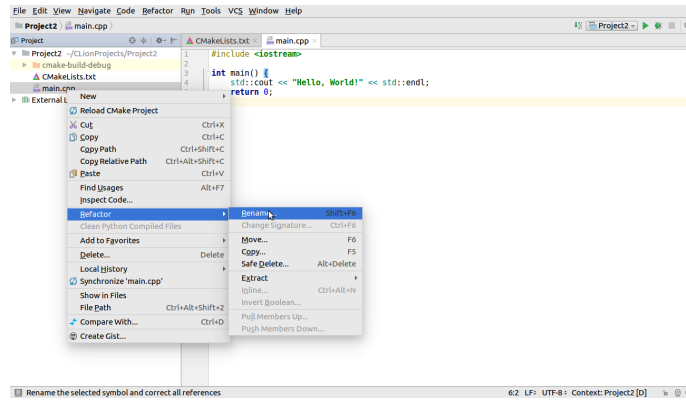


Figure 25: Renaming the file

In the following window, give the file its required name.

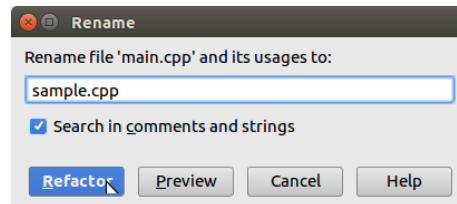


Figure 26: Renaming a file

Click on Refactor. You are now ready to start writing your program.

5 Running your program

Once you have completed writing your program, save your work by choosing File ->Save. Before running your program, you have to Build your project. This compiles your program and if your program is error free, it gets your program ready for execution. Choose Run ->Build. You should see something like the following screen:

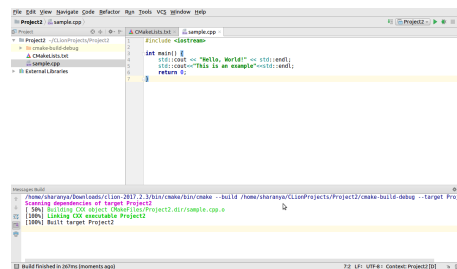


Figure 27: Building the Project

The console window on the bottom of your screen will show you the Build results. If your program has errors, they will be listed there. If the program compiles fine, it will say “100% Built

Target” followed by the Project name.

Once your project builds successfully, you can run it. Choose the project on the drop down list in the top right corner and then click on the green arrow button next to it. Your output will be displayed on the bottom of the screen in the console window.

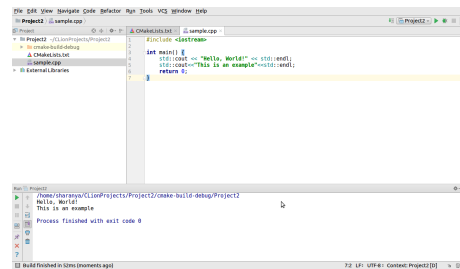


Figure 28: Running the Project

6 Opening an Existing Project

To open an existing project choose File >Open. In the dialog box that pops up, navigate to the location where you saved the project. Click on the project name and then click OK.

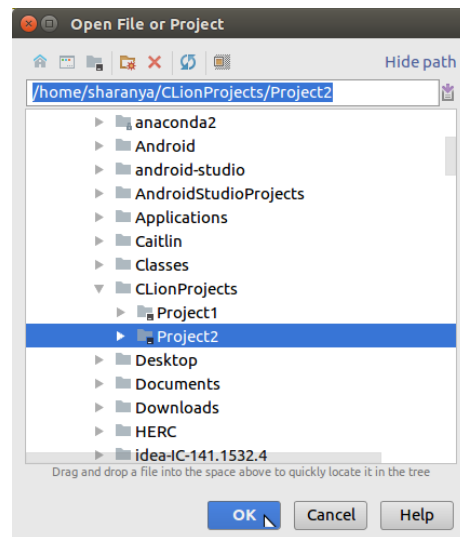


Figure 29: Running the Project

If you already have a project open, CLion will ask you if you would like to open this project in the same window or in a new window. You can choose either option.

7 Turning in your assignment

CLion would have created a folder where it places your projects by default. Alternatively, you might have specified a folder to put your project in when you created it. Navigate to the folder on Finder (Mac) or Folders (PC) and you will see a Folder for your Project. Opening that will show you all the source files under that project. The .cpp file is the file you need to turn in through Canvas

8 Troubleshooting

This section can be used for troubleshooting the most common errors you might encounter while installing CLion. If your problem persists after trying this, please contact your instructor/ TA for help.

8.1 On Windows

The most common issue on Windows is a glitch in the CygWin installation when a few critical dependency files are not downloaded. Please go through the following steps:

1. Open CLion. Go to File ->Settings. Choose **Build, Execution, Deployment.** and then choose **Toolchains.**
2. You will see “Download” link on the right. Click on the link.
3. Go through the steps for the CygWin installation again (the list below Figure 11), until you see the packages screen. However, this time, choose a different mirror.
4. Go to the Drop-down box on the top left, that says “Pending” and choose “Full”.
5. Go through the process of installing the 7 libraries again, but this time, make sure you get version numbers that are higher than the ones you have installed. Otherwise, go back and choose a different mirror. Keep doing this until you see higher version numbers, and install the latest versions.
6. Restart CLion. The old project would have been opened under the old settings. Open a new project. You should now be able to run it without issues. The very first time you build a project with CLion, it will take a little time to index all the libraries. Please wait until the little arrow on the top right goes green before running.

8.2 On a Mac

CLion usually installs without trouble on a Mac. Some older versions of OSX might have issues due to the different versions of the C++ compiler. OSX does not allow you to directly update the Clang tools needed for this, but you can re-install the XCode developer tools, which will get all the updates you need indirectly. Please go through the following steps.

1. Open the Terminal.
2. Re-install XCode developer tools by typing
`xcode-select --install`
and press Enter.
3. Accept the XCode user license. On the terminal, type
`sudo xcodebuild -license`
Enter the admin password for the computer when prompted. Please note that you will not see the password being typed in. This is ok. Type in the password and press Enter.
4. The license will then be presented on screen. Hit the space bar to scroll through the text. When you get to the end of the text, you’ll be prompted to enter “agree”. Type this in and hit Enter.
5. Restart CLion, click on File ->Reload CMake Project and everything should work.