CS221 – Lab14

Paul Haskell

INTRODUCTION

This lab will have you install "Eclipse for C/C++" on your computer and then use it to create, run, and debug a simple program. Finally, you will use Eclipse for C/C++ to push your program to GitHub.

Eclipse installation

Eclipse for C/C++ looks a lot like Eclipse for Java and it works similarly. But it is a separate program. You can have both installed on your computer at the same time, but you have to use the correct one for whichever language you are using.

To install Eclipse for C/C++ (I will just call it "Eclipse" from now on), browse to https://www.eclipse.org/downloads/packages/

and look for the "Eclipse IDE for C/C++" project. Download the appropriate package for your computer. The download will be a ZIP file. When the download is complete, right-click the ZIP file and extract it to someplace useful on your computer.

- On Windows I do not have luck putting eclipse inside the "**Program Files**" directory, because of file permission problems. I usually put Eclipse inside my **c:\Users\phaskell\Eclipse** directory
- Please don't forget where you put the program—you will need to run it!

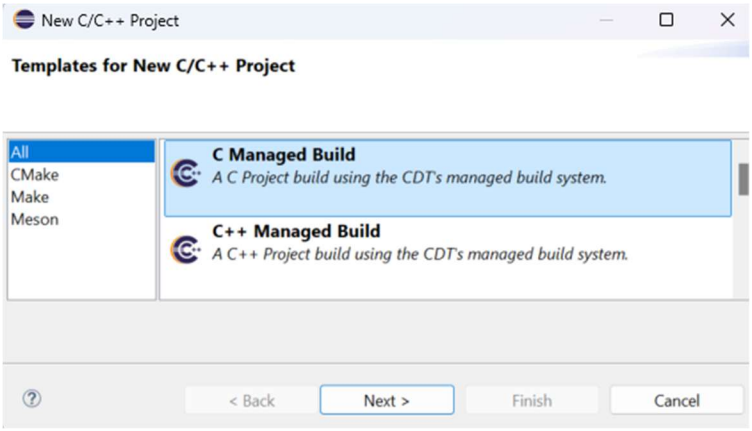
Browse to your Eclipse installation and run the Eclipse application. When Eclipse asks you to set your workspace, browse to your **CS221/MyWork** directory and select that. Check the "Use this as the default" checkbox. Then go ahead and launch Eclipse.

If you desire, you can import your existing C projects into Eclipse by selecting the **File->Open Projects from File System** menu and then pressing the **Directory** button to browse to each of your projects. You must import each project one-by-one, unfortunately.

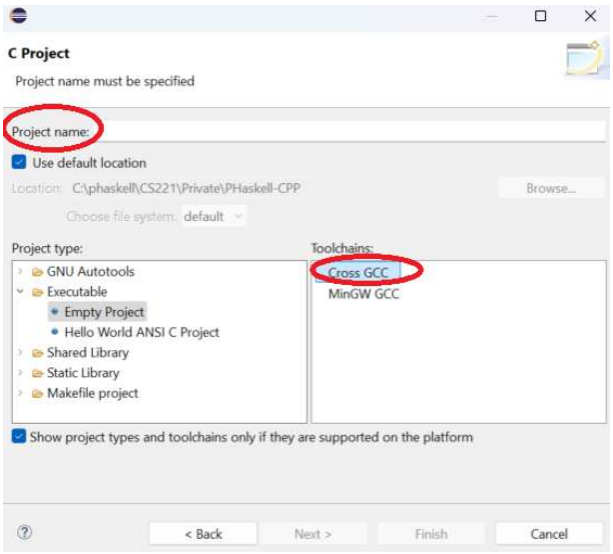
Today's Lab – a simple program

Using Eclipse, please create a new C/C++ project called **Lab14** .

- The type of the project should be "C Managed Build"



- On Windows, the "Toolchain" should be "MinGW GCC", not Cross GCC.



- On Mac, the "Toolchain" should be "Linux GCC", not Cross GCC
- Fill out the name of your project ("Lab14") and select **Finish** .

Now you will add a C file to your project. Right-click the Lab14 project entry and select **New->Source File** . Give your file the name **simple.c** .

In the file editor, have **simple.c** do the following:

- initialize a short variable to 1 and print it out
- left-shift it by 1 and print the result
- Keep repeating the previous step until the value stops changing. Print out that value.
-
- Now initialize a `char` variable to -128 and print it out
- right-shift it by 1 and print the result
- Keep repeating that step until the value stops changing. Print out that value.

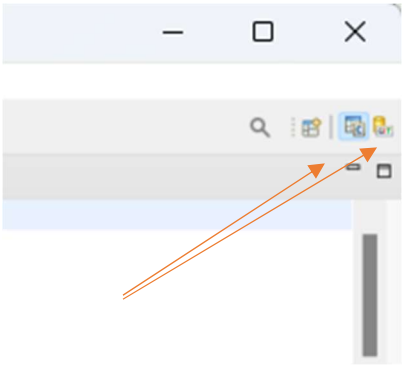
That's it!

Eclipse for C/C++ makes it easy for you to create, edit, compile, run, and debug C programs…but maybe not quite as easy as with Eclipse for Java. Here are a few hints:

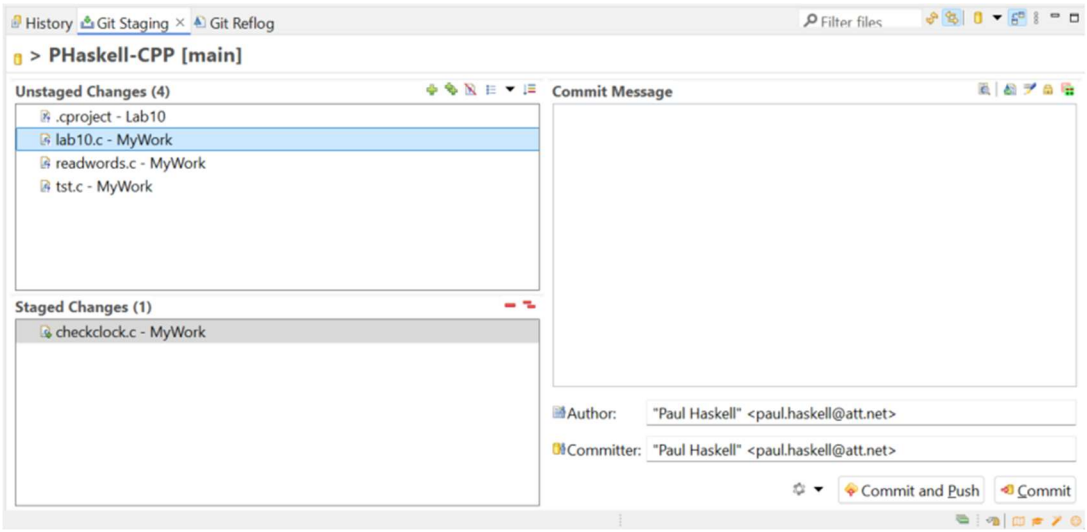
- To compile a brand-new project, right-click the project name in the Project Explorer and select **Build Project**. Hopefully building will happen automatically after a project's first build.
- To run a brand-new project, right-click the project name in the Project Explorer and select **Run As->Local C/C++ Application** . After the first run, you should be able simply to press the "Play" button (green circle with white triangle)
- Eclipse for C/C++ has very similar debugging abilities as Eclipse for Java. Please try them out!

GitHub

Your work isn't done until you push your program to GitHub, and we can do that inside Eclipse. First, please open the "Git Perspective" in Eclipse: open the **Window->Perspective->Open Perspective->Other** menu, and select "Git". From now on, little buttons let you jump to the Git Perspective or C Perspective whenever you want.



In the Git Perspective, if you select the "Git Staging" panel, you should see your **simple.c** listed as an "Unstaged change". Simply drag that file to "Staged Changes", enter a "Commit Message" in the proper window, and press the "Commit and Push" button.



Please use a browser to go to your github.com account and verify that **simple.c** actually was uploaded.

CONCLUSION:

The purpose of this lab is to get you set up to use Eclipse for the rest of the semester. If you are not familiar with Eclipse from past use, please tell the instructor, and we will find a time to do some review.

Please push the program to GitHub before the lab deadline. The deadline for completion of Lab15 is the end of class.

Task	Score, points
simple.c compiles, runs, and gives the correct output	15