

Using Eclipse for C Programming

George Ferguson

`ferguson@cs.rochester.edu`

September 2018
Updated Summer 2020

Abstract

Students are used to using the Eclipse IDE for programming in Java. This document describes how to install and use Eclipse for programming in C (and C++). Amazingly, it isn't as easy as it should be.

A word of warning to Computer Science students: At some point in your career you are going to need to be able to use the terminal/shell/command-line on your computer (the name varies across platforms). Even if you want to use Eclipse or another IDE for writing code, I **strongly** suggest that you learn how to build (compile) and run your programs from the terminal.

1 Obtaining Eclipse

- Visit eclipse.org.
- Click the big “Download” button.
- Don't click the “Download” button on the next page. Instead click the “Download Packages” link.
- Scroll down to “Eclipse IDE for C/C++ Developers”.
- Click the download link if that is the correct platform (that is, if you are downloading for the platform that you are using to do the download).
- Otherwise click the heading and it will take you to a page with links to download that version of Eclipse for all platforms.

2 Installing Eclipse

You've downloaded Eclipse. Congratulations! Time to install it.

2.1 Installing Eclipse on a Mac

- On a Mac, the download will be a `.dmg` disk image. It will probably be in your Downloads folder.
- Double-click it to “mount” the image.
- A Finder window should open displaying the Eclipse application and a shortcut to the Applications folder.
- To install Eclipse in the Applications folder, just drag the application onto the shortcut. You may need administrator privileges for this.
- But Eclipse can be installed anywhere, such as your Desktop or any other folder. Just drag it from the mounted disk image to where you want it.
- To unmount the disk image, click the eject button in a Finder window sidebar (if it is shown there), or drag the disk icon that is on your desktop to the Trash.
- You can delete the `.dmg` file once you've put Eclipse where you want it.

2.2 Installing Eclipse on Windows

- On a Windows machine, the download will be a `.zip` archive. It will probably be in your Downloads folder.
- Double-click it to extract (unpack) the archive.
- You can run Eclipse right from the resulting folder, or move it somewhere else.
- You can delete the `.zip` file once you've unpacked it.

3 Installing Java (yes, even for C/C++ Development)

Eclipse is itself written in Java, so you need Java on the machine to run Eclipse. If you have previously been using Eclipse for Java development on this machine, you will already have Java installed. If not, you will need to download it.

Note: As of April 2019, Oracle changed the licensing terms for Java. The version from Oracle is free for “personal use and development use.” I have added instructions for obtaining the OpenJDK version below.

Note: Mac users running macOS 10.15 Catalina or later should probably read the following document from Oracle: [How do I install Oracle Java on macOS Catalina \(10.15\)?](#).

- Download Java from java.com (Oracle)
- If you only intend to use Java to run Eclipse (but not develop in Java), you may use the “Java Download” button. This will download the current (latest) version of the Java Runtime Environment (JRE) for your platform.
- If you are planning to also do Java development, you need the Java Development Kit (JDK), which includes the JRE:
 - Click “Java Download.”
 - Scroll down to and click “See all Java downloads.”
 - In the left sidebar, click “Looking for the JDK?”
 - Then “JDK Downloads”. As of this writing, that takes you to [Java SE Downloads](#).
 - I used to recommend that students install “Java SE 8” (version 8 of the Java Standard Edition). The current version as of this writing is “Java SE 14.” I don’t see any reason not to use that and it works fine for me with the current version of Eclipse.
 - Click “JDK Download” and click the download link for your platform.
 - Accept the Oracle license agreement in the resulting popup window or see below about OpenJDK.
- Either way, run the installer that you just downloaded (`jdk-14.0.1_windows-x64_bin.exe` or whatever it is called).

You should now be able to run Eclipse by double-clicking it.

Note: The less restrictively-licensed version of Java is OpenJDK from jdk.java.net. However that is a very barebones website. You can download the current version of Java (SE 14 as

of this writing), early access versions of future releases, and legacy versions. The downloads are not installers, so you need to know what you're doing to make them work.

An alternative is the [AdoptOpenJDK](#) site. Read the the "About" page. If you're comfortable with them, download and use their installers. Good luck!

4 Installing a C Compiler

You might reasonably expect that once you installed something named "Eclipse for C/C++ Development" you would be able to develop C (or C++) programs with it. Unfortunately that isn't the case. In fact, it isn't the case for Java development either. For Java development, you need to install Java. For C development you need to install a C toolchain. Here's my method.

4.1 Installing Developer Tools on a Mac

- Download the `xCode` IDE from the Apple App Store. This is a huge package that supports development for macOS and iOS, but it also includes all the programs that you need for C development using Eclipse.
- There are other ways to get what you need (for example, the [HomeBrew](#) package manager), but they are all more involved. If you don't have 5GB of free disk space for XCode or don't want to install it, you'll have to look around.
- And Computer Science students take note: using XCode is not the same as learning how to do things from the terminal. However once you've installed the XCode tools, you can use the C compiler from the Terminal application.

4.2 Installing MinGW on Windows

- Download and install MinGW ("Minimal GNU for Windows") from mingw.org.
- The instructions are at [Documentation](#) then [Getting Started](#).
- The short version is to download `mingw-get-setup.exe` from this link: <https://sourceforge.net/projects/mingw/files/latest/download>.
- Double-click the downloaded file (probably in your Downloads folder) to start the installer.

- Accept all the default settings and click “Continue” to do the installation.
 - A student reported that the default settings didn’t seem to work for them. Here’s another guide: [MinGW C++ Download and Installation](#)

“Ensure on the left that **Basic Setup** is highlighted. Click the three boxes indicated below: **mingw32-base**, **mingw32-gcc=g++**, **msys-base**. After clicking each, select **Mark for selection**.”

The guide goes on to describe what to do next, namely closing the MinGW Installation Manager (which is a weird thing to do) and dealing with the resulting dialogs: **Review Changes**, **Apply**. Check it out. I hope it helps.
- It told me that MinGW was previously installed (which it wasn’t), but telling it to “Reinstall” worked fine.
- You may delete `mingw-get-setup.exe` after the installation is complete.

5 Using Eclipse for C Programming

Finally!

First create a new C/C++ project:

- From the menubar: New > C/C++ Project
- Select “C Managed Build”
- Enter project name and select “Executable > Empty Project”
- For “Toolchain,” on a Mac select “Cross GCC” and on Windows select “MinGW GCC.”
- For “Select Configurations,” either accept the default (two configurations, Debug and Release) or uncheck Release
- Ignore the “Cross-compiler Settings” (leave them blank).
- Click “Finish.”

Customize the project settings (your mileage may differ, but these are my recommendations):

- From the menubar: Project > Properties
- Sidebar: C/C++ Build > Settings

- For Debug configuration:
 - Tool Settings tab:
 - * GCC C Compiler item (or other depending on platform)
 - Dialect: Language Standard: select C99
 - Warnings: check `-Werror` as well as `-Wall`
 - Build Artifact tab:
 - * Artifact name: Replace `${ProjName}` with whatever name you want if you don't want your executable to be named after your project (which probably has spaces in the name, etc.).
- Project > Build Automatically should be checked

There must be a way to save these settings for use with future projects. . .

Now add a new file to the project:

- From the menubar: New > Source File
- Enter name, *e.g.*, `test.c`
- Note that this does not use a `src` folder as you may be used to seeing with Java development using Eclipse. Simply adding a folder and using it doesn't Just Work, but I'm sure it could be made to work.

Write the code. I suggest doing Hello World for your first program. Eclipse will flag your errors like in Java every time you pause and/or save.

Try to run your program with the green play button like in Java. You will get an error something like "Launch Error: Binary not found." Awesome.

You need to create a "Run Configuration:"

- From the menubar: Run > Run Configurations
- Select "C/C++ Application"
- Then click the "New" icon (the page with a plus sign) to create a new run configuration.
- Name: Debug C Application (or whatever you want to call it)
- Project: leave blank or erase
- C/C++ Application:
 - On a Mac: `${workspace_loc}/${project_name}/Debug/${project_name}`

– On Windows: `${workspace_loc}/${project_name}\Debug\${project_name}.exe`

- Apply, Close

Now click the green play button again and you should see the program output in the Eclipse console.

That is all. You now know as much as I do about this.