

Installing the Eclipse C/C++ Development Environment

Philip Gust

2017-09-06

We will use the Eclipse Integrated Development Environment (IDE) to write, test, and debug applications for the class. Eclipse is open source software produced and maintained by the Eclipse Foundation. The Eclipse IDE that is used by most software professionals, so gaining experience using it will be valuable to you in future employment. Eclipse supports development in a number of language. For this class, we will be using C and perhaps some C++.

Requirements

To install Eclipse, you will need to download and install the following:

- Eclipse C Development Tools (CDT)
- Java Runtime Environment or Java Development Environment (required to run Eclipse)
- C / C++ Compiler

Download Eclipse installation package

The current version of Eclipse is Oxygen. If you have already installed a recent version of Eclipse (e.g. Eclipse for Java Developers), you can simply add the C/C++ Development Tools (CDT) later on. Otherwise, Download Eclipse at

<http://www.eclipse.org/downloads/eclipse-packages/>

You'll notice that there are versions preconfigured for many programming languages and types of development. We will choose the "Eclipse IDE for C/C++ Developers" configuration. In later courses you may also use Eclipse for development in Java, PHP, Javascript and for various kinds of development projects such as the Web, Android, or enterprise applications. You can always add these modules later directly from Eclipse, once it is installed.

Eclipse runs on Windows, MacOS, and Linux. The download page detects your operating system and offers the appropriate packages. MacOS is always a 64-bit operating system.



On Windows and Linux, it offers both the 32-bit and the 64-bit versions.



> Download the installer for the version of CDE that corresponds to your 64-bit or 32-bit operating system now.

Download Java

You will need Java to run Eclipse, even though you won't be programming in Java for this course. The current version of Java is Java 8. If you have not already installed it, you can download it here:

<http://www.oracle.com/technetwork/java/javase/downloads/>

Java Platform, Standard Edition

Java SE 8u111 / 8u112
Java SE 8u111 includes important security fixes. Oracle strongly recommends that all Java SE 8 users upgrade to this release. Java SE 8u112 is a patch-set update, including all of 8u111 plus additional features (described in the release notes).
[Learn more](#) ▶

Important planned change for MD5-signed JARs
Starting with the April Critical Patch Update releases, planned for April 18 2017, all JRE versions will treat JARs signed with MD5 as unsigned. [Learn more and view testing instructions](#).
For more information on cryptographic algorithm support, please check the [JRE](#) and [JDK Crypto Roadmap](#).

- Installation Instructions
- Release Notes
- Oracle License
- Java SE Products
- Third Party Licenses
- Certified System Configurations
- Readme Files
 - JDK ReadMe
 - JRE ReadMe

JDK
DOWNLOAD ▶

Server JRE
DOWNLOAD ▶

JRE
DOWNLOAD ▶

Which Java package do I need?

- Software Developers: JDK** (Java SE Development Kit). For Java Developers. Includes a complete JRE plus tools for developing, debugging, and monitoring Java applications.
- Administrators running applications on a server: Server JRE** (Server Java Runtime Environment) For deploying Java applications on servers. Includes tools for JVM monitoring and tools commonly required for server applications, but does not include browser integration (the Java plug-in), auto-update, nor an installer. [Learn more](#) ▶
- End user running Java on a desktop: JRE:** (Java Runtime Environment). Covers most end-users needs. Contains everything required to run Java applications on your system.

There are two installation configurations: Java Runtime Environment (JRE) and the Java Development Kit (JDK). You only need the JRE to run Eclipse. However, many developers opt to install the full JDK because they plan to develop in Java in a later course or in a future job. The JDK includes the JRE plus all the tools and libraries required to develop Java applications. Here is the JDK download screen.

Java SE Development Kit 8u112		
You must accept the Oracle Binary Code License Agreement for Java SE to download this software.		
<input type="radio"/> Accept License Agreement <input checked="" type="radio"/> Decline License Agreement		
Product / File Description	File Size	Download
Linux x86	162.42 MB	jdk-8u112-linux-i586.rpm
Linux x86	177.12 MB	jdk-8u112-linux-i586.tar.gz
Linux x64	159.97 MB	jdk-8u112-linux-x64.rpm
Linux x64	174.73 MB	jdk-8u112-linux-x64.tar.gz
Mac OS X	223.15 MB	jdk-8u112-macosx-x64.dmg
Solaris SPARC 64-bit	139.78 MB	jdk-8u112-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	99.06 MB	jdk-8u112-solaris-sparcv9.tar.gz
Solaris x64	140.46 MB	jdk-8u112-solaris-x64.tar.Z
Solaris x64	96.86 MB	jdk-8u112-solaris-x64.tar.gz
Windows x86	188.99 MB	jdk-8u112-windows-i586.exe
Windows x64	195.13 MB	jdk-8u112-windows-x64.exe

> Download the JRE or JDK that corresponds to your 64-bit or 32-bit operating system now.

Download a C/C++ compiler (MacOS and Windows only)

The Eclipse requires a compiler for whatever language you are developing in to be installed. Linux come with the Gnu C/C++ compiler and tools (GCC) already installed. On Windows and MacOS you will need to install a C/C++ compiler.

Windows

MinGW is a smaller package that includes only the compilers and a few additional tools. Most developers who use an IDE choose this option because it is smaller and simpler. You can download it here:

<http://www.equation.com/servlet/equation.cmd?fa=fortran>

You should choose the self-installer that corresponds to your 64-bit or 32-bit operating system.

Releases	Self-Extracting File		Note
	32-bit	64-bit	
6.2.0	gcc-4.8.2-32.exe	gcc-4.8.2-64.exe	Including <i>gdb</i> ; <i>make</i> ; OpenMP; <i>neuLoop</i> ; manuals

> If you choose Fortran, C, and C++ for Windows, download the installer for the version that corresponds to your 64-bit or 32-bit operating system now.

MacOS

A compiler and other development tools is available with the Xcode command line tools available from Apple. Before downloading them, try running 'gcc -v'. If you are prompted to download the command line tools at that point, follow the instructions. If the response is 'gcc: command not found' then go to:

<https://developer.apple.com/download/more/>

You will need a free Apple ID to login: create an Apple ID if you do not already have one. Now select the latest non-beta version of the Xcode command line tools to install. An installation package (.dmg) will be downloaded. Double click on the downloaded package to begin installation. Verify your installation by re-executing the command “gcc -v” and observing the displayed information.

Installation

Several steps are platform dependent, while the rest apply to all platforms

Windows

Run the GCC installer You may need to run the GCC installer as administrator (Run as administrator from popup menu). Several installation instructions recommend installing either GCC package in the root of your C: drive (C:\) rather than the default location so that Eclipse can find the compiler. For example, install MinGW in C:\MinGW. I have not independently verified that this is necessary.

MacOS

The GCC compiler comes with the Xcode tools, but the debugger program GDB does not, so you need to install it from the Homebrew project. Here is a link to an article that explains how to do this.

<https://medium.com/@royalstream/how-to-install-and-codesign-gdb-on-os-x-el-capitan-aab3d1172e95#.yf2ip0ux7>

Run the following command to install a gdb configuration file in your home directory as “.gdbinit”:

```
echo set startup-with-shell off > ~/.gdbinit
```

All Platforms

Run the Java installer so that is in place for the Eclipse installer to detect and configure. On Windows, the 64-bit version is installed in C:\Program Files\Java or the 32-bit version is installed in C:\Program Files (X86)\Java.

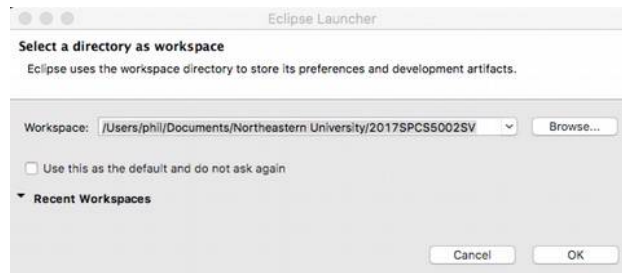
Now run the Eclipse installer to install the Eclipse CDE. If you are on Windows, copy the *eclipse* directory to C:\Program Files for the 64-bit version or C:\Program Files (X86) for the 32-bit version. You may need to restart your computer to ensure that everything is properly configured. You can add Eclipse to your Windows, MacOS or Linux system dock or toolbar for convenience.

Once all components are installed, start the Eclipse CDE. You will see the splash screen while it starts.



You will be asked to specify a directory to use as a workspace. This is where your projects will reside. It is convenient to create a new project for each class assignment, so I use a separate project directory

per class. I have a “Northeastern University” subdirectory where I keep all my Northeastern University projects and workspace subdirectories for classes, such as “2017FACS5001SV” with projects for the CS 5001 Fall 2017 class in Silicon Valley.



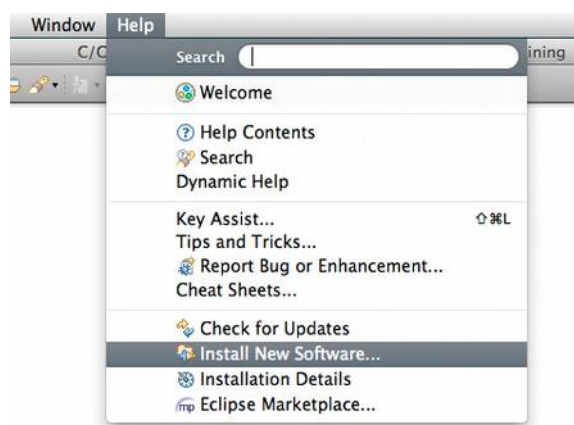
Once you specify the workspace directory, you will see a welcome page like this if you have just installed Eclipse.



If you previously installed Eclipse for Java Developers

If you previously installed a Java version of Eclipse, you can install the C/C++ Development Tools (CDT) now.

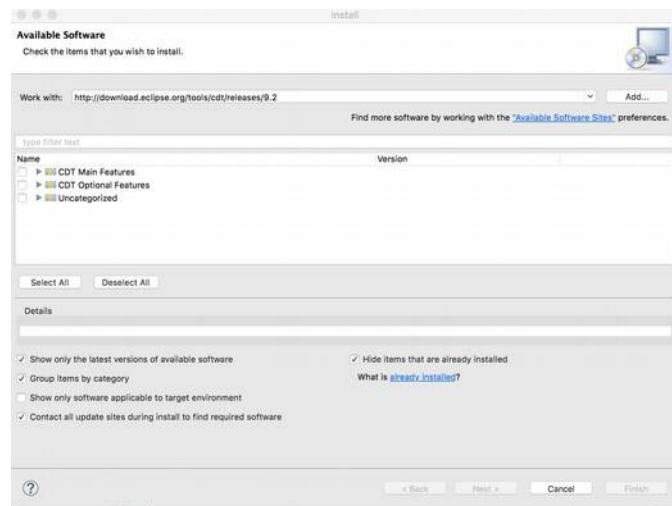
To install the Eclipse CDT, navigate to *Help > Install New Software*.



In the window that appears, paste this link to the textfield next to “Work with:”

Eclipse Neon: <http://download.eclipse.org/tools/cdt/releases/9.2>

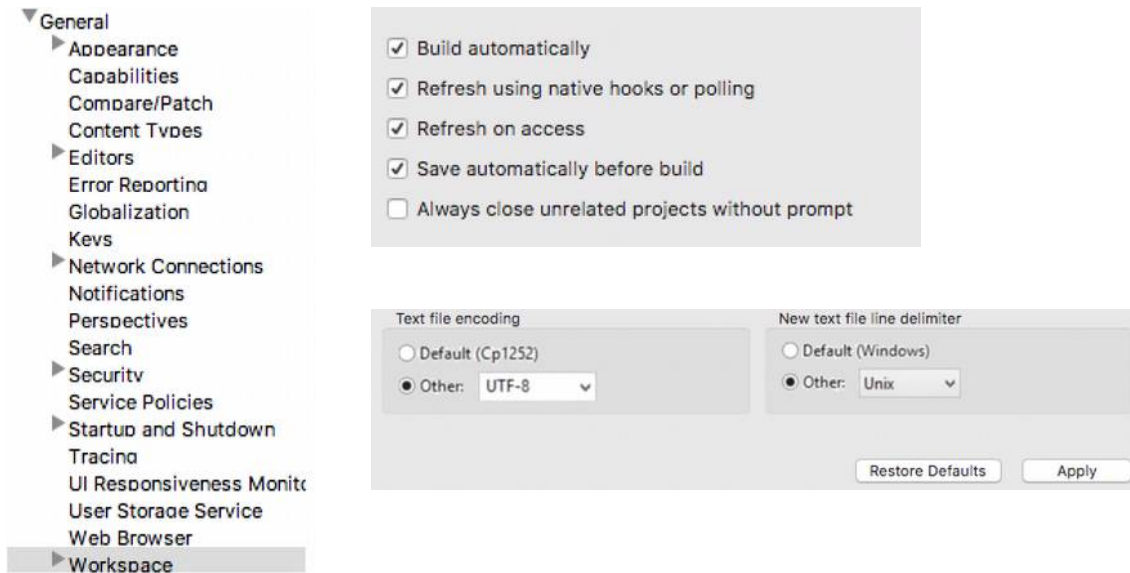
Eclipse Oxygen: <http://download.eclipse.org/tools/cdt/releases/9.3>



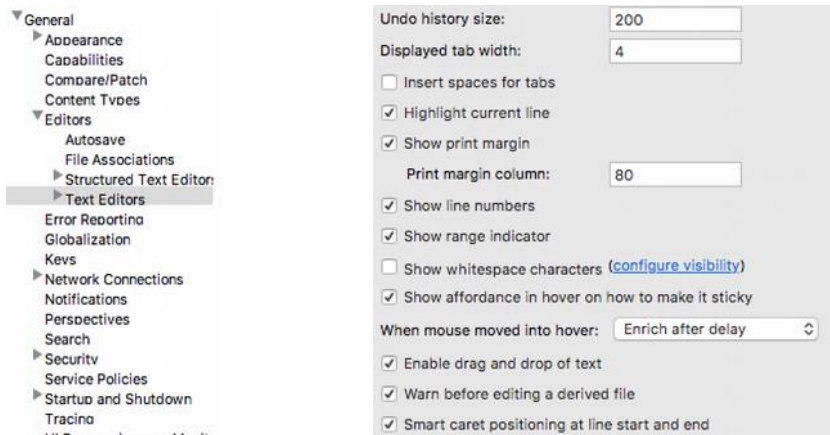
Select “Select All” to select all the installation packages, and then select “Finish.”

Configuration

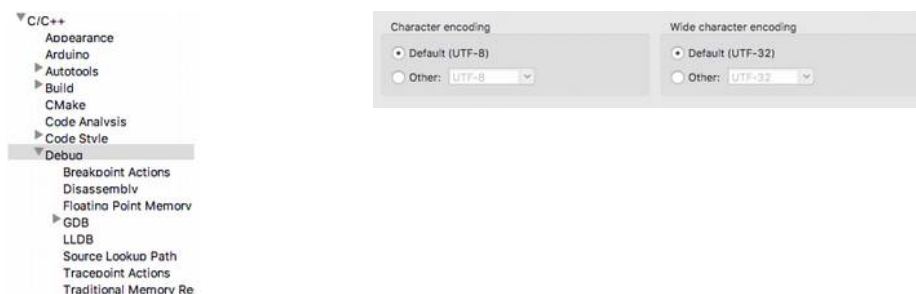
We’ll do some configuration before creating a project, so select the “Workbench” icon in the upper right to go directly the workbench. Open the “Preferences” dialog. On Windows and Linux, it is on the “Windows” menu. On MacOS, it is on the “Eclipse” menu. Select “General”->”Workspaces” in the tree at left. Make sure that the top four boxes are checked. Also make sure that “Text encoding” is UTF-8 and “New text file delimiter” is set to “Unix.” This ensure files can be shared across operating systems. If you made changes, select “Apply.”



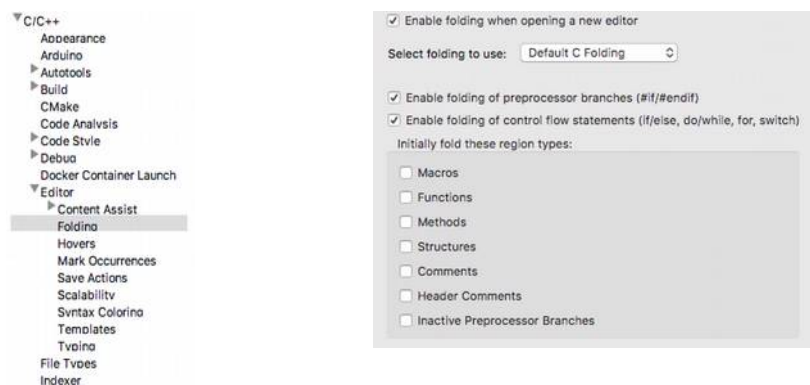
Under “General”->”Editors”->”Text Editors” in the tree at left, ensure every box is checked except “Insert spaces for tabs” and “Show whitespace characters.” If you made changes, select “Apply.”



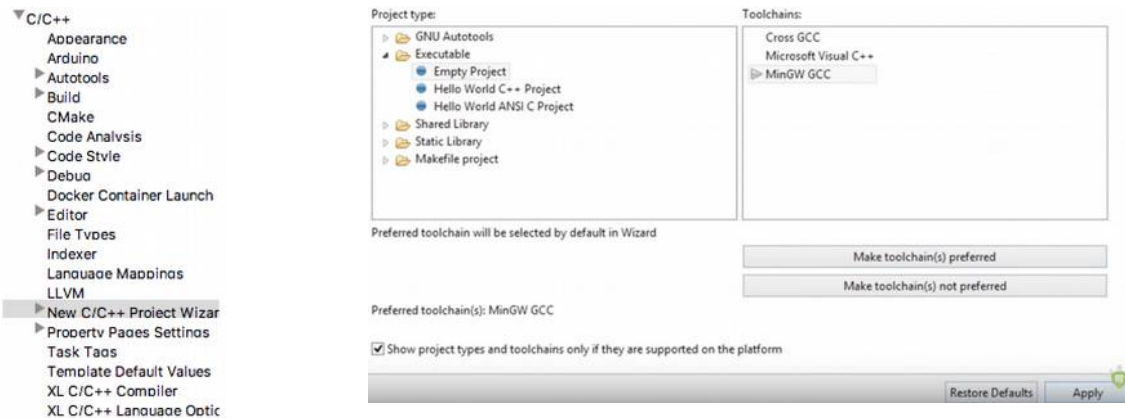
Under “C/C++” ->“Debug” in the tree at left, at left.make sure that “Character encoding” is UTF-8, and “Wide character encoding” is UTF-16. If you made changes, select “Apply.”




Under “C/C++”->“Editor”->“Folding” in the tree at left, ensure that only “Enable folding when opening a new editor,” “Enable folding of preprocessor branches,” and “Enable folding of control flow statements” are checked. If you made changes, select “Apply.”

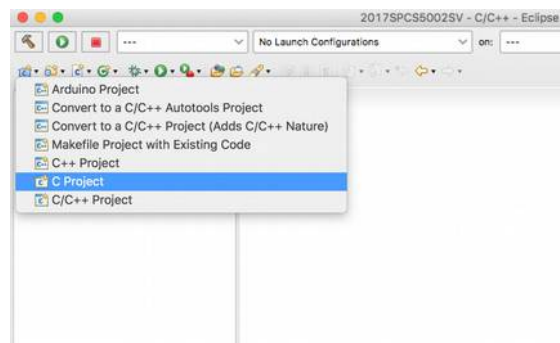


Under “C/C++”->“New C/C++ Preferences” in the tree at left, ensure that all the entries in the “Executable” category use the appropriate compiler. On MacOS, select “MacOSX GCC” for each of the four entries. On Windows, select whichever GCC compiler you installed (e.g. “MinGW GCC”). Then select “Make toolchain(s) preferred”. If you made changes, select “Apply.”

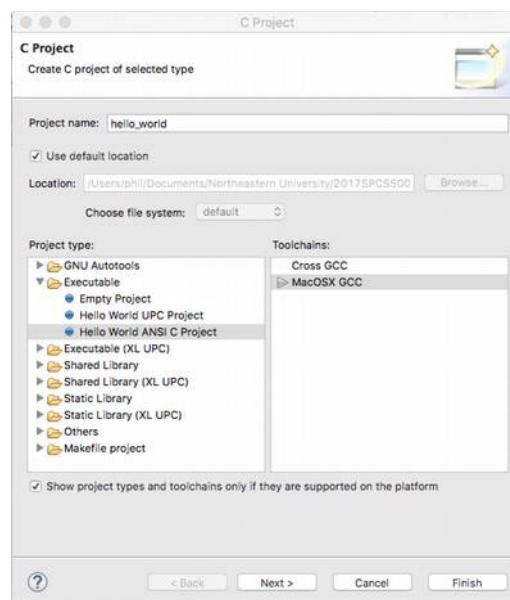


Testing

We'll test our Eclipse CDT installation by creating a project. You can either pull down the menu for the first icon in the toolbar () and select "C Project"




or use "New Project"->"C Project" from the "File" menu. In the "C Project" dialog, select "Hello World ANSI C Project," give the project the name "hello_world."

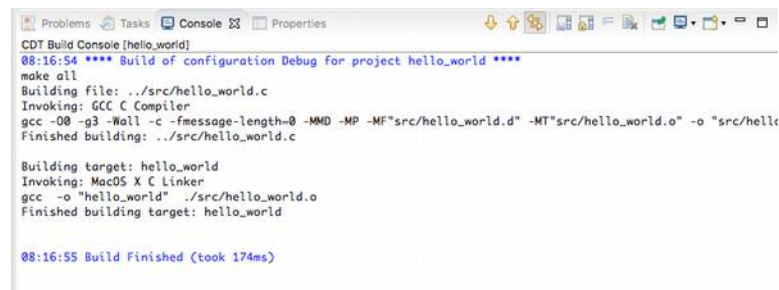



Now select “Finish.” We will examine other options another time. This creates a C project with a sample program that prints “Hello World!” to the console. Notice that the Author field is filled in and that the program will print. There are two resources for this project: “includes” and “src.” Expand the project tree to show C program source file.

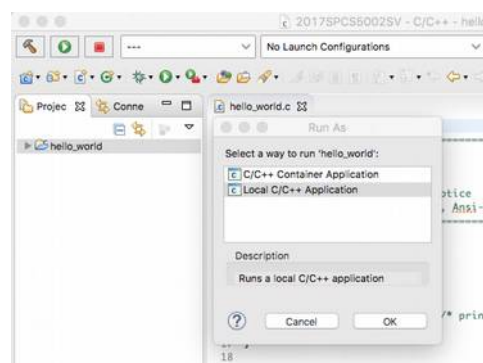


We’ll look at this first C program another time, but for now, let’s run it. Before you can run the program for the first time, you need to build it. Select the  (Build) button in the tool bar. This compiles the C program into an executable form. We’ve configured Eclipse to save unsaved files before rebuilding and incrementally rebuild before running or debugging.

You’ll now notice two new resources: “Binaries” and “Debug,” corresponding to the standard and debugging versions of the application. The GCC compiler output is shown in the console tab in the bottom right of the window.



Select the “hello_world” project in the left pane, then select  (Run) in the tool bar and select the “Local C/C++ Application” option.




The output of the program is displayed in the console pane.

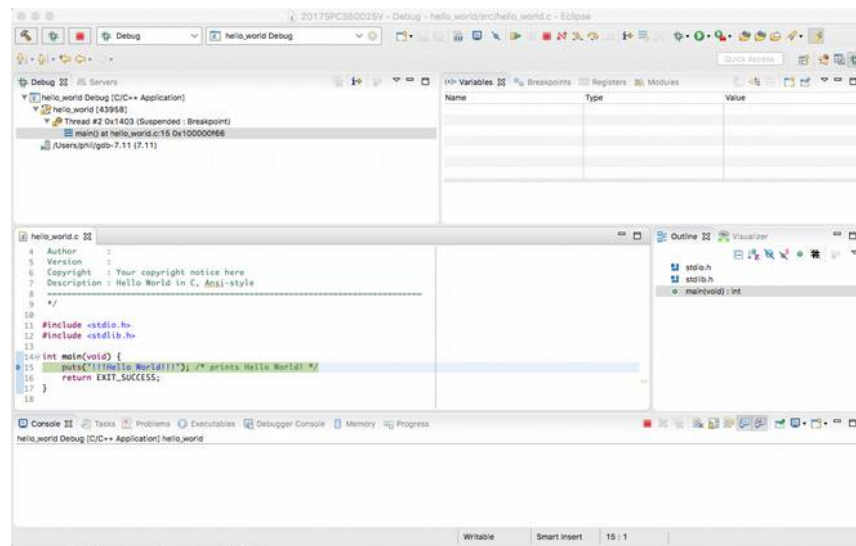



Now let's make sure debugging works.

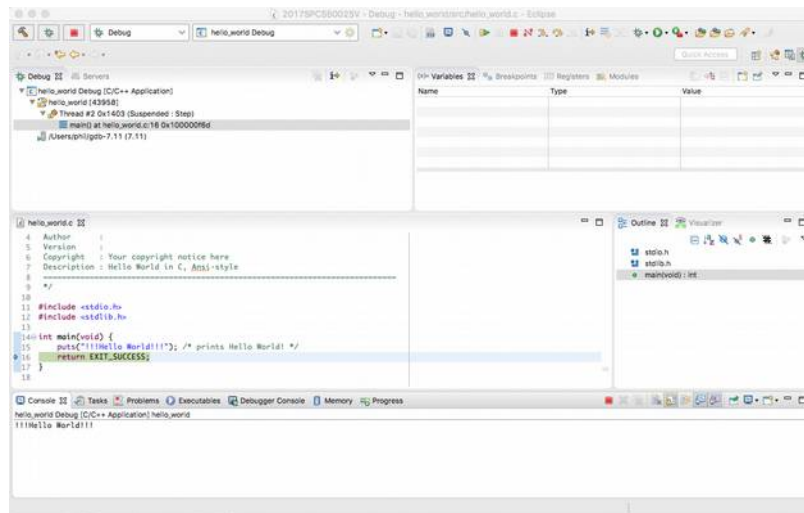
MacOS Only: Before using the debugger, you will need to configure the location of 'gdb' and the '.gdbinit' file that you created. From the "Run" menu, select "Debug Configurations..." and then select the "Debugger" tab of the configuration dialog. Modify the *GDB debugger* text as "/usr/local/bin/gdb" and modify the *GDB command file* as "~/gdbinit". Then select *Apply* and then *Close*.




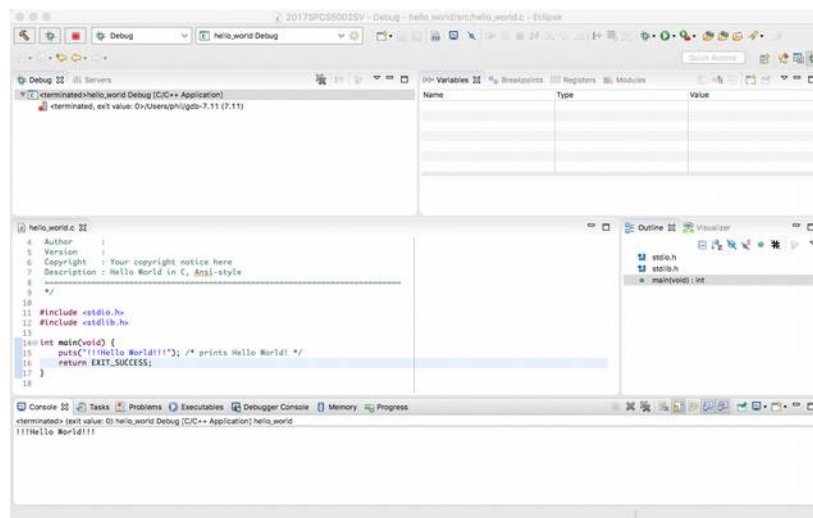
Now select  (Debug) in the tool bar and select the "Local C/C++ Application" option. Eclipse automatically changes from the "C/C++" perspective to the "Debugger" perspective. By default, the debugger automatically stops at the first line of the program.



Select  (Step Over) to execute the print statement and move to the next statement.



Select  (Resume) to resume running the program.



Once everything tests out correctly, you're ready to use Eclipse CDT for your class assignments.