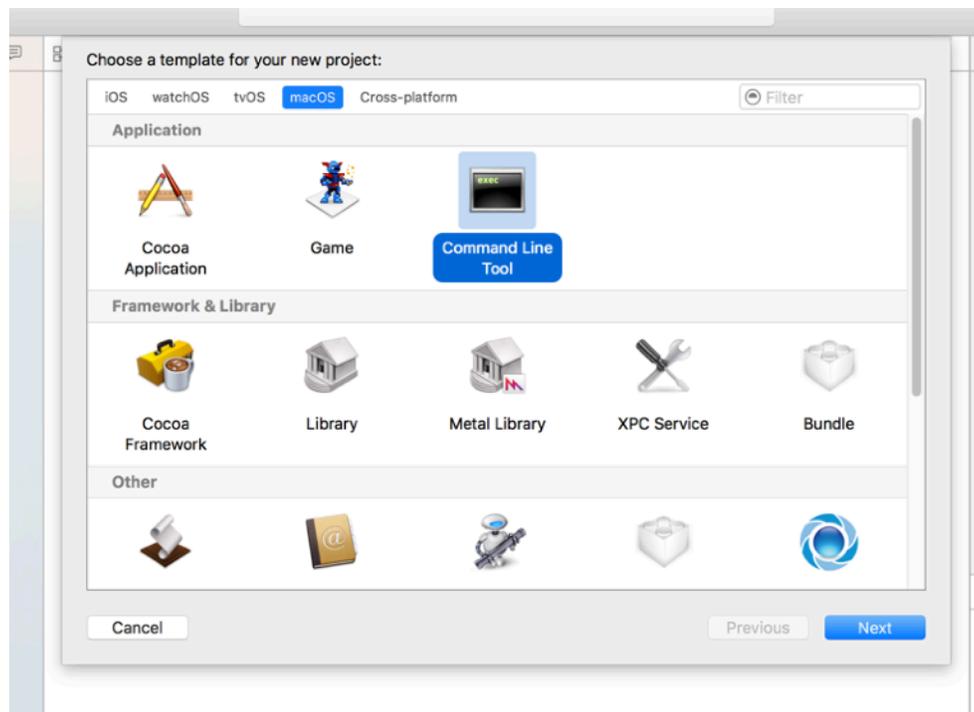


Installing and getting started with Xcode for Mac OS.

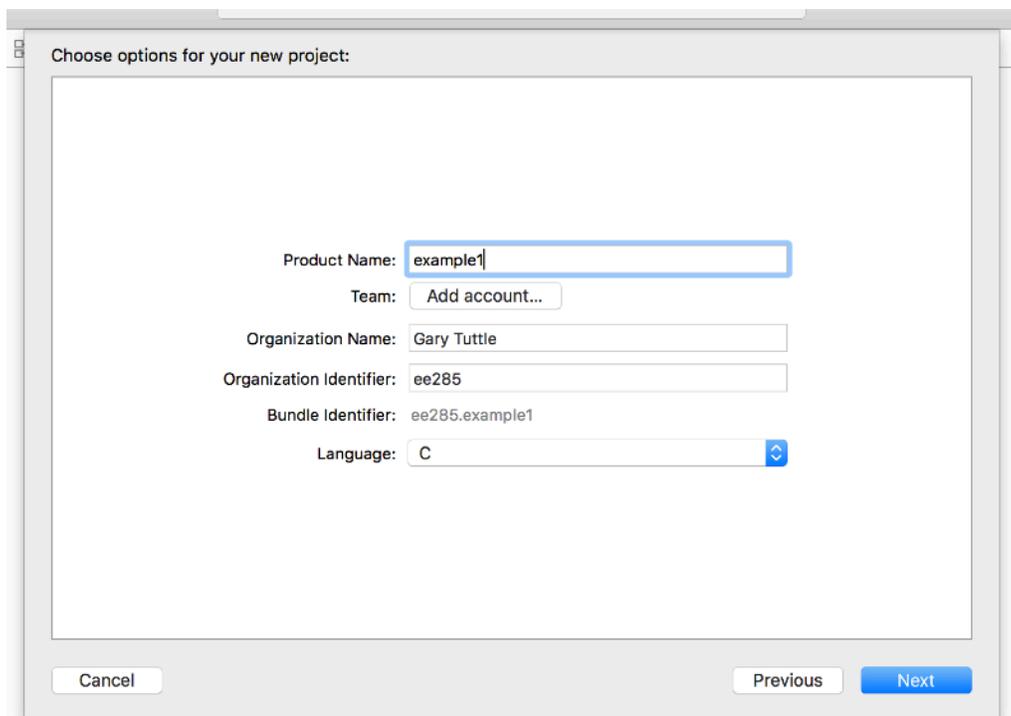
1. Go to the Mac App store. Do a search for Xcode. Then download and install it. (It's free.) Give it some time — it may take a while. (A recent update took almost an hour to download and install for me.)
2. Once it is installed, launch it.
(Tip: Use Spotlight, which can be invoke by hitting Command-Space Bar, type in “Xcode” and hit return to launch it. This saves the hassle of rummaging around in the applications folder, and you don't have to keep the icon in the dock.)
3. You will be presented with the welcome screen. The area on the right-hand side of the window is blank now. Once you start making and saving some programs, recent projects will be listed here. Click on “Create a new Xcode project”. (You can also choose that option from the File menu.)



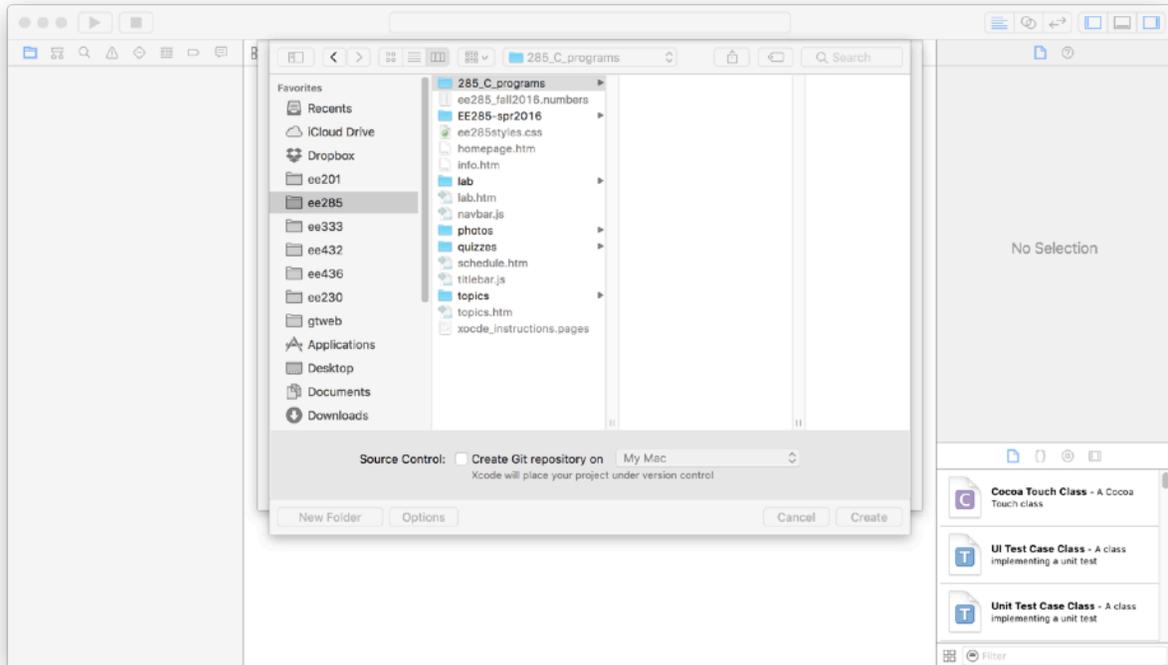
4. A window will open with a variety of options for project types. Someday, if you become an macOS or iOS developer, you might choose some of the other stuff. For EE 285, we need to stick with C, so select “Command Line Tool”. Then click “Next”.



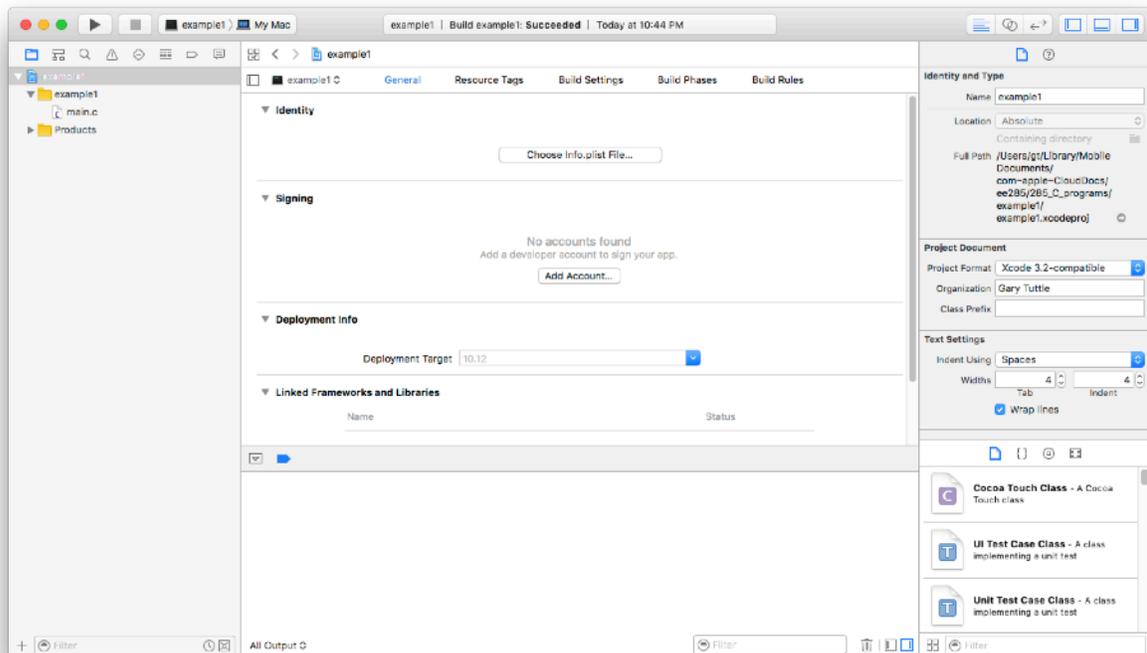
5. You can enter some information about your project. What you enter here is not terribly important, but Xcode will put some of this info in the template for your program. Click “Next”.



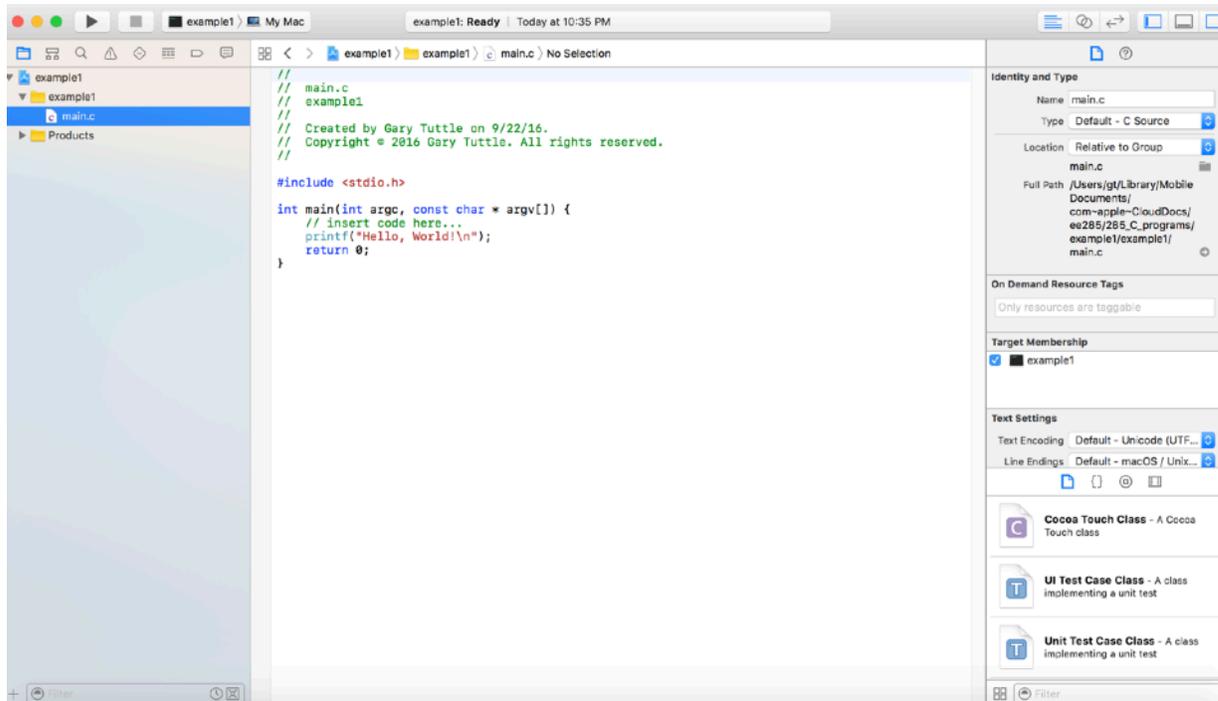
- You will be asked to find a place to keep the files for this project. You can store your programs wherever you want on your hard drive. For instance, I have made a program folder within the folder where I keep all my EE 285 materials.



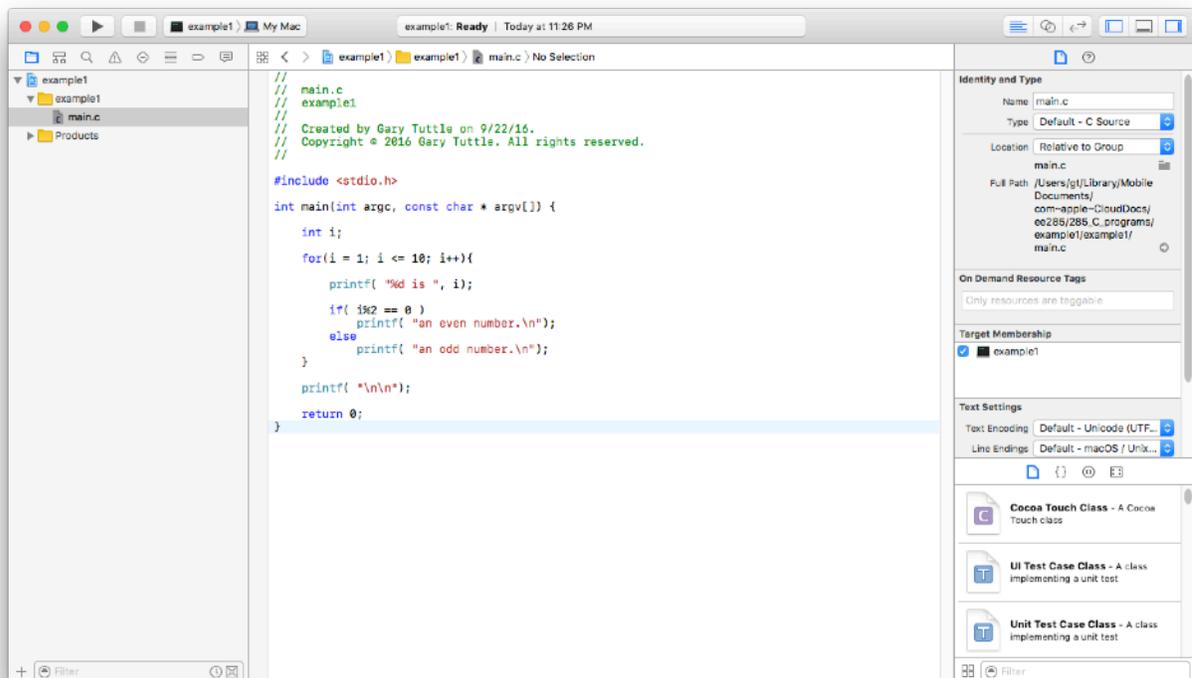
- Xcode will set up all the files for your project and then present you with the rather nondescript window shown below. None of the stuff here is important now.



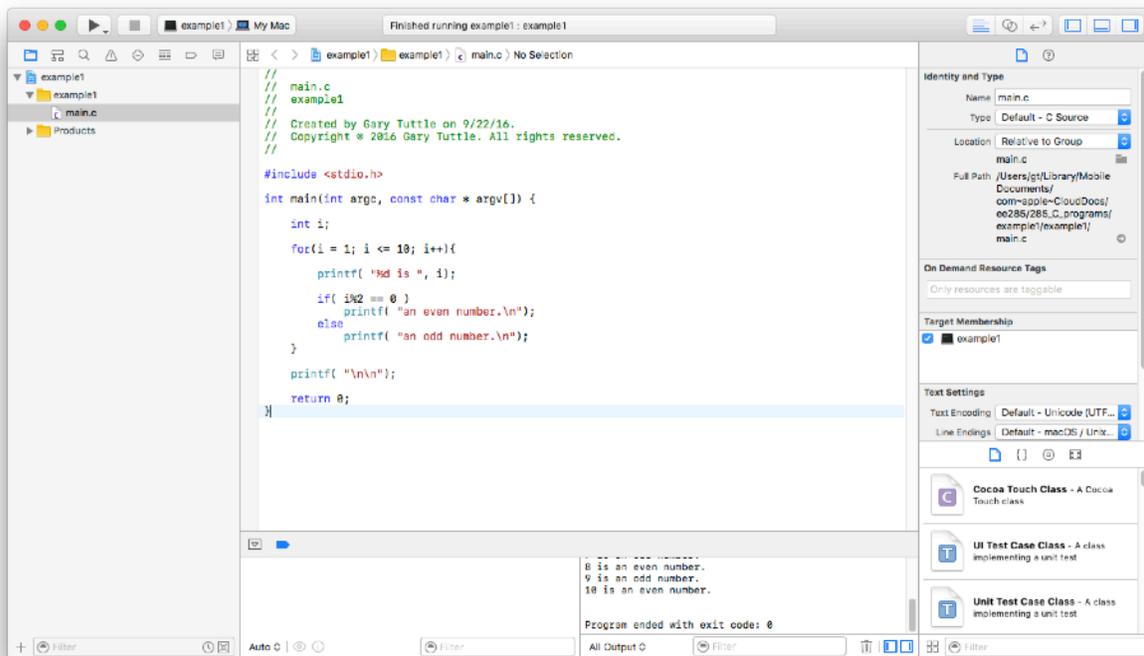
- To get started with what is important, click on the “main.c” file icon on the left side of the window. The text file that will hold your program code opens. Xcode gives you a “Hello, world” template as a starting point.



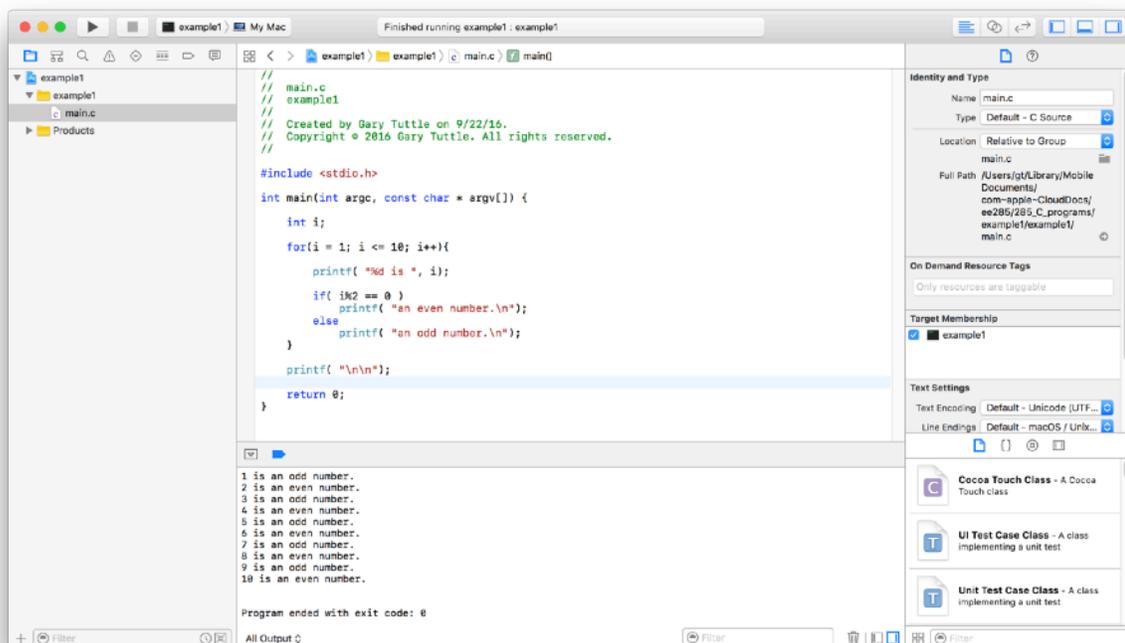
- You can remove the text that you don’t want from the template and enter the code that you do want. Below, I have entered yet another silly little EE 285 practice program.



10. When you are ready, you can run the program by clicking the “run” arrow in the upper left corner of the window. (Or select the “Product -> Run” menu item or use the “Command-R key combination.) The program runs and the output is printed into the to window area in the lower right. (The lower left is the debugging window area — we will not worry about that now.)



11. The window areas can be re-sized by dragging the dividers. Unless I want to use the debugger, I will usually minimize that area (essentially to zero) and make the output area much bigger.



12. When you are finished, choose the “File -> Save” menu item. (Or use the “Command-S” key combination.) Actually, Xcode is continually saving the files as you work, so even if you forget to save explicitly, your files are probably still safe. If you are all done for the day, you can quit Xcode. The next time you start up, you will have the option of continuing on with the project or starting a new one.

Using the Xcode “IDE” (integrated development environment) has some advantages. One of the main things is the automatic formatting and colored keywords in the text file. (All aspects of the formatting can be changed through the preferences (“Xcode -> Preferences” menu item.) For now, you should probably use the default formatting. Once you have written enough code of your own to become fussy about how it looks, you can change the preferences

Also, the code is checked as you type. Basically, you can’t run the code unless all variables are declared, all command syntax is correct, all opening braces have matching closed braces, etc. And when there are errors, Xcode will give messages as to what is wrong.

As noted above, there is an integrated debugger, which can help you find the errors in the operation of your code. As your programs get bigger and more complex, they may not run the way you think they should, even though all of the syntax is correct. The debugger helps you sort out problems by allowing you to stop and look at all of the variables at any point in time. This can be extremely useful, although we probably will not make much use of it in 285.

There are other advantages to using an IDE, most of which are beyond our EE 285 capabilities. If you continue to program beyond this class (and you will), you may have opportunity to make use of other aspects of the IDE.