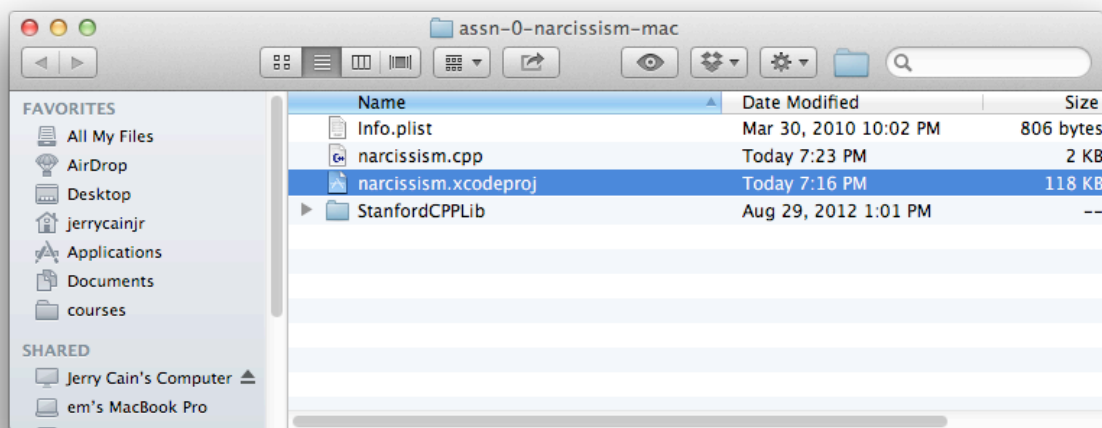


Using Xcode

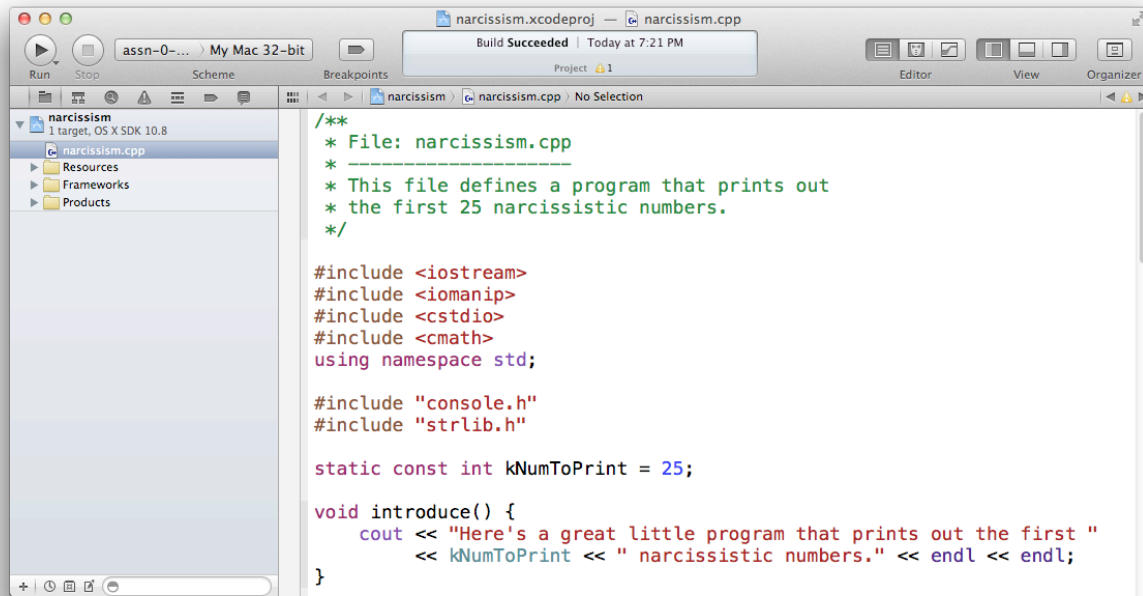
Once you have access to Xcode, you should play around with it for a while before tackling any of our major assignments. I've gone ahead and posted a starter project for Assignment 0: Narcissistic Numbers, which is a **complete, working program** that prints out the first 25 narcissistic numbers. **You don't need to actually code or submit any work for Assignment 0**—it's simply here so we have a small programming project to refer to while teaching you Xcode.

Download the version of the Assignment 0 starter files appropriate to your own operating system. Unzip the file, and open up the folder to expose the following set of files:



The **narcissism.xcodeproj** is the primary file that helps you manage all of the other files that contribute to a working C++ program. In general, you should look for file ending in **xcodeproj** whenever you download and extract assignment starter files.

When you want to code and the project isn't open, you'll typically double click the relevant **xcodeproj** file to launch Xcode and to open up the Xcode project the **xcodeproj** file represents. If you go ahead and double click **narcissism.xcodeproj**, you should be presented with the following:



In particular, you should see a Finder-like dock in the left margin, and a code editor should occupy the majority of the project window. If you don't see the editor, you can single click on the **narcissism.cpp** file in the left margin to get it, or you can double click to open the code editor in an altogether separate window.

There are aspects of this setup that should be familiar to those of you who used Eclipse in CS106A to write Java programs. The high level concepts are the same: you use the editor to write your code, and you use a combination of menu items and shortcut keys to compile and run your programs. In particular:

- if you want to compile and run your code, you should select **Run** from under the **Product** menu, or you should press ⌘R to do the same thing. Provided your code is syntactically correct and well formed, your program will execute. If there are syntax errors, the project will decorate each of the offending lines with some angry, red symbols resembling stop signs.
- if you just want to confirm that your code compiles, but you're not interested in running it just yet, you can select **Build** from the **Product** menu, or you can press ⌘B if shortcuts are your thing.
- if you want to create a new file and add it to the project, try the New \rightarrow File... option under the **File** menu, or hit ⌘N . You'll be led through a wizard that helps you name and place your new file. (You can also control-click on the project name at the very top of the left margin. Doing so will present a flyout menu with an option to add—either new or existing—files to the project.)
- If you want to add an existing file to the project, you can do so by selecting **Add File To "narcissism"** under the **File** menu. (Shortcut is option- ⌘A .)

These commands should be enough to make your way around Xcode. Of course, it helps to know what the program output should be, so try ⌘R and confirm that the code generates something like this:

Here's a great little program that prints out the first 25 narcissistic numbers.

```
1.) 1
2.) 2
3.) 3
4.) 4
5.) 5
6.) 6
7.) 7
8.) 8
9.) 9
10.) 153
11.) 370
12.) 371
13.) 407
14.) 1634
15.) 8208
16.) 9474
17.) 54748
18.) 92727
19.) 93084
20.) 548834
21.) 1741725
22.) 4210818
23.) 9800817
24.) 9926315
25.) 24678050
```