## cisc1110 fall 2010 How to get started with **Xcode**

1. Start up Xcode

From the Finder, select the **Go – Go to Folder** option:

Finder file Edit View	Beck Forward Occasing Failur	N H H	
	10 Computer 17 Honse 19 Natwork 12 Julik	0 MC 6	Finder File Edit View Car Window Help
	Applications DUITIBLE	0 KA 0 MU	Go to the folder:
	Recent Polders		/Developer/Applications
	Co to Folder Connect to Server	0 #C	(Camair) (Gamair)

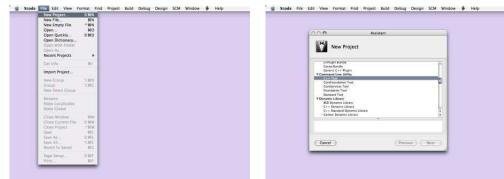
and type **/Developer/Applications** in the "Go to Folder" window that pops up. Then find the **Xcode** icon in the "Developer - Applications" window that looks like this:



and double-click on it and Xcode will start up.

# 2. Open a project

Click on File – New Project to create a new project, and then select C++ Tool, under Command Line Utility in the Assistant window that pops up after selecting "New Project". Then click on Next.



Then, in the "Assistant" window, enter the name of the project in the text field next to the **Project Name**. Note that the **Project Directory** field will fill in automatically with the project name entered above. If you want to change the directory (i.e., folder) where the project is located, then edit the **Project Directory** field. When the name of a directory begins with the tilde and slash ( $\sim$  /) characters, this means that the directory will be found under your home directory (i.e., as a subfolder inside your home folder).

New C+-	Assistant		000 New 0	Assistant	
Project Name: Project Directory: -/		• ( Choose )	Project Name Project Directory The project directory hello.xcodeproj all o	-/heila/	• Choose
Cancel	e	Previous) Next	Cancel		(Previous ) (Finish

### 3. Edit the project

When the project window opens, double-click on the line that says **main.cpp** to open the editing window.

000	🛅 hello	D	000	🛅 hello	C
Active Target Action		Q- String Matching	Active Target Action		Q+ String Matching Search
Graups & Fries.	Tie Name ■ Helio ■ Helio ■ Helio ■ main spo	A S Code O A O	Source	File Name * 4	c ¥

Next, the text editor window opens. The default template is shown on the left. On the right, the template has been edited to contain the version of the "hello world" program that we created in class.

000	a main.cpp		000	e main cpp	
Build Build and Co #include clostron #include clostron Unt nain (int argc, // insert code	akts For Breakpoints : <no selected="" symbol=""> : chor + corst grav[]) {</no>	Project Ungrouped	Build Build and Co Build And Co B	Tasks Fix Breakpoints p:5 : mainO :	Project Ungrouped
			hello exited normaily.		© Succeeded //

4. Compile the project

Find the **Build** icon located in the upper left corner of the editor window:



and click on it to compile your code.

The status bar at the bottom of the editor window will display any status messages, such as "Build succeeded" or "Build failed". If there are errors compiling, and the "Build failed" message appears, then the lines where the errors occur will be marked with red X's in the left margin of the editor window. Correct the errors and click on "Build" again.

If the build is successful, then go on to the next step.

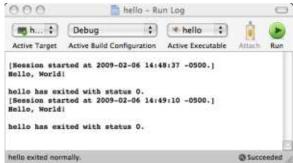
#### 5. Run the project

Find the **Build and Go** icon located in the editor window, next to the "Build" icon:



and click on it to execute your code.

A "Run Log" window will appear to display any output your program generates. The status bar at the bottom of the run log window display any status messages, such as "program exited normally."



#### 6. File structure

When you use an Integrated Development Environment (IDE) like Xcode, it is important to understand the file structure of the project which is generated automatically by the IDE. As indicated earlier (see step 2), opening a new project creates a directory which, by default, has the same name as the project. In our example, this directory is called  $\sim$ /hello/. Inside this directory, there is a file called hello.xcodeproj, the main C++ source code file, which is called main.cpp, a subdirectory called build, and a documentation file called hello.1. Inside the build directory, there is another subdirectory called Debug; and inside the Debug directory, there is a file called hello. This file is the executable, or runnable file that is generated by the compiler. You could execute this file from the Terminal application command line, as we have done in class.

The overall project file structure is organized like this:

```
~/hello/
|
+---- hello.xcodeproj
main.cpp <<=== this is your C++ source code
hello.1
build/
|
+---- hello <<=== this is your program executable</pre>
```

or as shown in the Mac Finder window:



When you save your work, you should save the entire ~/hello/ project folder.

When you submit assignments, you should ONLY submit the C++ source code, i.e., the main.cpp file.