cis1.5, spring 2009, Xcode "how to"

1. Start up Xcode

There are three ways to start up the **Xcode** application:





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.

Open Open Quickly Open Dictionary Open With Finder Open Aa Recent Projects	NO	New Project
Get Info	30 ¹	CrPupit fuilite Doce Bunde
Import Project		Generic C++ Rugen T Command Units Units
Trew Croup Croop New Smart Croup	CMN	Celli Teri Cerdiovices Tool Postituines Tool
Rename Make Localizable Make Cichal		* Downer Ubery * Downer Ubery 10 Downer Ubery C++ Dysam CUrbry C++ Dysam CUrbry C++ Baniad Opumer Ubery +
Close Window Close Current File Close Project Save Save AL Save AL Biovert to Saved	900/ 2010/ *610/ 전문 전문 전문 전문	Caterof Presson Room

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		New C	Assistant	
Project Name: Project Directory: -	Ŧ	Choose)	Project Name Project Directory The project directory helo-acodeproj all'o	hello -/hello/ -/hello/ all be counted if no a counted therein.	Choose Choose
Cancel		(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	000	🛅 hello	D
(the hello	- 🔨 🔬 🕘 🥥 🖃	Q+String Matching	📼 🔨 🏡 👅 🚺 🗖	Q+ String Matching
Active Larget Action Active Larget & Trics Active Larget & Trics Courses	nino euro ano con tanta imo contra Trio tanta helo. belo. ≥ main.cpp ✓	Code A A Code A Code A Code A Code A Code A Code Code	Lation Sund and do fairs into color helo helo main.com	Code • A • C A

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	C
Build Build and Co Build Build and Co Read Read Streams Int wain (Int argc, el Vineert code std::sout << 'Hei return 8; }	<pre>tks Fix Breakpoints : <no selected="" symbol=""> : thar 4 corst argv[]) { rs ite, KartdlYn";</no></pre>	Project Ungrouped	Build Build and Go Build Build and Go main.cop Winolude clothroat using namespace at int main Q [roturn 8; }	Tasks Fix Breakpoints	Project Ungrouped
		li.	hello exited normally.		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.