COMPILING THE PROJECTS

We'll be using VS, g++ and CMake throughout this quarter. It's your chouce.
We'll also be using "outofsource" builds.

If you're using the lab machines, g++ and CMake are already installed. To compile and run your code from the command line, follow these steps from the folder containing the file `CMakeLists.txt`.

```
> mkdir build
> cd build
> cmake ..
> make -j4
> ./Lab01
```

You should see some output on the console. Look at `main.cpp` to see what the command line arguments mean. Once you enter the correct arguments, you should see the following output file in the build directory.

If you're using your own computer, follow the steps below to install g++ and CMake. Otherwise, you can skip to Step 2.

Linux

Use a package manage ro download cmake. For example, with Ubuntu,

```
> sudo apt-get update
> sudo apt-get g++
> sudo apt-get install cmake
```

Follow the steps above to build and run the program from the command line.

OSX

You can use home brew /macports or install these manually.

- Xcodedevelopertools.You'll need to login with your AppleID.
- CMake (http://cmake.org/download/)

Make sure the commands `g++` and `cmake` work from the command prompt.

If you want to use Xcode as your IDE, then do the following from the folder containing `CMakeLists.txt`.

```
> mkdir build

> cd build

> cmake -G Xcode ..
```

This will generate`Lab01.xcodeproj` project that you can open with Xcode.

- To run, change the target to `Lab01` by going to Product-> Scheme-> Lab00.Then click on the play button or press Command+R to run the application.
- Edit the scheme to add command-line arguments or to run in release mode.

Windows

You'll need to download these manually.

- VisualStudio. Anyversion should work. I've tested VisualStudio 2015 (aka version 14) on Windows 8.
- CMake (http://cmake.org/download/). Make sure to add CMake to the system path when asked to do so.

Make sure the command `cmake` works from the command prompt. On Windows, you'll be using Visual Studio as the IDE. Run the following from the folder containing `CMakeLists.txt`. to generate a solution file.

```
> mkdir build

> cd build

> cmake ..
```

By default on Windows, CMake will generatea VisualStudio solution file,`Lab01.sln`, which you can open by double-clicking.

If you get a version mismatch,you may have to specify your visual studio version, for example:

```
> cmake -G "Visual Studio 14 2015" ..
```

Other versions of Visual Studio are listed on the CMake page (http://cmake.org/cmake/help/latest/manual/cmake-generators.7.html).

- To build and run the project,right-click on `Lab01` in the project explorer and then click on "SetasStartup Project." Then press F7 (Build Solution) and then F5 (StartDebugging).
- To add a command-line argument, right-click on `Lab01` in the project explorer and then click on "Properties" and then click on "Debugging."