Setting up your computer to compile and run C/C++

You will need to compile/run C/C++ for this course. Here are some suggestions if you do not have something set up already. Keep in mind that I don't have all these systems so may only be able to offer limited suggestions on getting them to work. I am most familiar with LINUX variants, WSL and Cygwin on windows.

Apple Owners: Xcode is a collection of compilers, header files, libraries, and software development tools for Mac. You will need the C/C++ compiler. The DVD to install Xcode used to come with your system if you have an older Mac. Now, Xcode is available at https://developer.apple.com/technologies/tools/. Its compiler is a hacked version of gcc to support Objective C. The Xcode is multiple gigabytes so make sure you have space.

Many Unix applications run in X Windows environment. Apple's implementation of X Windows X11 for Mac OS X uses native graphics engine and is fast. It uses its own window manager that looks just like the rest of Mac Aqua interface. It can be installed from the Snow Leopard system DVD. You also need "X11 for Mac OS X SDK" from the system DVD.

Some sites with tips on how to get this working on different versions of the OS are:
https://wiki.helsinki.fi/display/HUGG/GNU+compiler+install+on+Mac+OS+X
http://en.wikibooks.org/wiki/C%2B%2B_Programming/Compiler/Where_to_get

Once you get Xcode installed you need to figure out how to create and save a C++ program. At this stage it is probably easiest to just use Xcode as the editor. However, Xcode can do a lot of different things so it is important to figure out how to use it to edit C++ programs. A useful tutorial that will help you create the program for the HelloWorld example can be found on YouTube,
eg: https://www.youtube.com/watch?v=-H_EyIqBNDA

Once you have the C++ code typed in and saved, take note of where you saved it. Now fire up the terminal program (search for terminal and then run it). Instructions for using the terminal are at the bottom of this page, basically you need to first navigate to the directory where your program is saved using the "cd" command (the "ls" command shows you what is in the current directory). Once you get the directory with your program you need to compile it and run it as described in lecture and in chapter 1 or the text.

Windows Owners: There are a couple of options for windows.
WSL(Windows Subsystem for Linux): A recent addition to windows 10 (summer 2016) is the Ubuntu Bash shell. Though part of windows 10, it has to be enabled. It is a good idea to update windows first, before enabling the Ubuntu bash shell. Directions on doing this can be found at:
https://solarianprogrammer.com/2017/04/15/install-wsl-windows-subsystem-for-linux/
A quick starter on getting a simple program up and running using the WSL can be found at:
http://www.developerinsider.in/compile-cpp-program-with-gplusplus-compiler-on-bash-on-ubuntu-on-windows-10/
Cygwin: If you do not have 64-bit Windows 10 you will need to use Cygwin. Go to http://x.cygwin.com/ and run the setup program as described on this page. In most cases, taking the default settings should work. At some point you will have to select a "mirror" site. A good choice is something that is close, like one of the waterloo sites, or a large US government site. When the "Select Packages" window appears, scroll down to the heading

"Devel" and click on the "+" by it. In the list of packages that now displays, scroll down and find the "gcc-c++" package; this is the compiler. Click once on the word "Skip", and it should change to some number like "5.4" etc. (the version number), and an "X" will appear next to "gcc-core" and several other required packages that will now be downloaded. You should also select the xinit package from the X11 menu as mentioned in the website I gave above (click on the + on the X11, scroll down until you see the xinit package and select it). You may also wish to look through some of the other packages that are available, for example "octave" is a useful package that is a clone of matlab.

You can alternatively download and use a c++ compiler with a gui. Some free ones are listed at http://www.cprogramming.com/compilers.html Previous students have had good experience using the code::blocks compiler or MS Visual C++ (there are free versions for students).

You will probably also want a syntax aware editor like notepad++ (free, google it).

Linux owners: You should have the g++ (gcc) compilers installed already. Some people also like developer packages like Qt that has lots of tools for building c++ programs and debugging them.

If none of the above work for you, please inform the instructor. There may be some Applied Math Linux systems you can use to do your assignments.

Line Terminal

Using a line terminal is not essential, but can be an incredibly fast and effective way to interact with your computer. There are lots of online resources to help (google "getting started on the line terminal"). Some other useful tips are:

1. The "tab" key can be used to auto-complete commands.
   For example "cd" plus the first couple of letters of the directory you want to go to can autocomplete with the tab key if the first few letters lead to a unique directory. If not unique it will tell you the possibilities;
2. You can use the up/down arrows to go to previously executed commands to execute them again.
3. Some useful commands:
   - "pwd" tells you where you are now;
   - "ls" shows you the files in the current working directory;
   - "cd newdirectory" changes the current directory to "newdirectory";
   - "cd .." moves you up one in the directory tree;
   - "ps" tells you about processes running on your computer, also "ps –a" and "ps –u" can be useful;
   - "top" starts a program which shows how the computer resources are being used (ctrl-c to stop it);
   - "kill" can be used with various options to stop processes that shouldn't be running. Look up variants online or try "info kill" to get some options.
   - If you do not set a path variable then to access files or programs in the current directory (say file "afile") then you should preface it with ./ which just indicates the current directory (i.e. ./afile is the file with name "afile" in the current directory).