The source code for the HelloWorld program can be found on the OWL site.

To compile and run it, you will need to follow instructions in the SetUp file, also found on the OWL site to get

1. A C++ compiler, and preferably one you can run in a terminal window if you want help getting the code working. If you are already comfortable using something like Visual C++ on Windows or Eclipse on Mac that is fine, but neither the Instructor or TA use these so can't help you with them.
2. A syntax aware editor like notepad++, which is available free (google it).

Once you have these installed, you need to fire up a line terminal:

- Windows with WSL: click on the Ubuntu icon in the task bar (red circle with white dashed circle with 3 legs inside); Cygwin: click the "Cygwin Terminal" icon you hopefully created on the desktop (if you didn't, just rerun the setup program and click that box at the end).
- Mac: Use the Finder and go to programs-> Utilities and you should find a "terminal" program.

Open "HelloWorld.cpp" in your editor. You need to save this file in the same directory that your terminal is open to. To find out where your terminal is open to (i.e. what local directory you are in) type "pwd" on the command line and hit <return>. Now either

1. save "HelloWorld.cpp" into that same directory (Using cygwin, "pwd" in the bash shell will report that the local directory is something like /home/yourname which can be found in the normal windows explorer by starting in "This PC" and then navigating to C:\cygwin64\home\yourname for the Cygwin drive. Using WSL, it does not really want you to change linux files with a windows app. The easiest solution is to create a symbolic link to a windows directory. For example, you can create a "Projects" folder in your windows Documents directory. Then in your /home/yourname WSL you can type:

   ```
   ln -s "/mnt/c/Users/<WindowsUser>/Documents/Projects" /home/<LinuxUser>/Projects
   ```

   if you change directories in the bash terminal using "cd Projects" you will now be in the windows projects directory and can access files you put there..
2. Or you can move your terminal to a different directory using the 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;
   - Using cygwin,
     cd "/cygdrive/c/Users/yourusername/My Documents" will put you in your Documents folder. The quotes are necessary if there is a space in your directory name. Using WSL, cd "/mnt/c/Users/yourusername/Documents" will do the same.

We can now compile the test.cpp source code by running the compiler using the following in the terminal:

```
g++  HelloWorld.cpp –o hello
```

and then run it using

```
./hello
```

The “./” at the beginning tells the computer to look in the current directory. This will output the results to the terminal screen. To put the result in a data file "data.dat" instead of on the screen then redirect the output using

```
./hello > data.dat
```

While not needed for this program, now would be a good time to review section 1.3.3 in the text on compiler flags so that you know them for future programs.

Another couple of useful tips for using the command line:

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.