COMP 222 Computer Organization
Accessing/Using the GNU C Compiler

Intro:

The GNU C compiler can be accessed using free software depending on the operating system. Different installations will be discussed below.

I. Operating System

Option 1: Using Linux:

1) The GNU C compiler **gcc** should come with the installation (typically under **usr/bin**). To find where it is located, type at the command prompt: $ **which gcc**
2) If not installed, install via the command: **install gcc**

Option 2: Using MacOS:

1) The GNU C compiler **gcc** usually does not come pre-installed, but is available via **xcode.**
2) Download the **xcode** package from: **http://connect.apple.com.** You will need to register for an Apple Developer Connection account. Once you have registered, login and click Download Software and then Developer Tools. Find the Download link next to Xcode Tools (version) – CD Image, and click it.
3) Find the downloaded package, double click it, and follow the installation instructions to install GCC and a host of other development applications. The GCC compiler will be located at **/usr/bin/gcc**.

Option 3: Using Windows:

1) The best/free GNU C compiler I have found is part of **mingw** (least warnings/errors when compiling), available at: **http://mingw.org** (it is also installed in the JD 2214 lab)
2) Click on **Download** under **Navigation** (don't worry about logging in or registering, unless you want to)
3) Download **mingw-get-inst-20120426.exe** (or whatever the latest version is—less than 700 kB) and follow the instructions w/ default select options
4) Compiler by default is located at: **c:\mingw\bin.** To set the path to easily access the compiler, set the access path via the following steps:
    a. Open the **Control Panel** (under Start), Click on **System→Advanced→Environment Variables**
    b. Edit **PATH** by adding the following to the end of the list: ;**c:\mingw\bin**

II. Editing:

For ease of use, the best editor is the one that comes with your operating system (i.e. Notepad or Wordpad for Windows). If you have JGrasp, you may be able to set it up to work with **mingw** (under **Settings→Compiler Settings→Workspace,** select **gcc – MinGW (C:\mingw\bin)**). Be aware, sometimes it works, sometimes it doesn't.

III. Compiling (at the command prompt):

To compile your file, make sure you are in the directory where the file is. To change to a subdirectory, type **cd** *directoryname.* To go back to a parent directory, type **cd ..**

To compile, type **gcc** *inputfilename* –o *outputfilename*. This will create an executable file called *outputfilename*.**exe** (in Windows). If you do not use the –o flag and specify *outputfilename* it will create a default file called **a.exe (**in Windows**)** or **a.out** (in Linux). I'm not sure what it produces in Mac OS, sorry.

IV. Running:

To run the executable file, simply type the file name: *outputfilename.* If a message ecomplains that it cannot find *outputfilename*, type *./outputfilename*. This forces it to look in the current directory.

V. Documentation for C syntax

A decent online "manual" for C syntax can be found at wikipedia.org, entering "C syntax" or "C language" as the search item.

VI. Debugging

1) For compile-time errors, trace the line (or general area) where the error occurred. Comment out the suspected line or area of code (using **/* ….. */)** to narrow down the problem. If unsure of the area, start with a largely commented area, compile, if no error, narrow down the area, and repeat the process till you find the error.
2) For run-time errors (ex: segmentation fault, memory error), check to see if the variable in scanf is missing the & symbol(ex: **scanf("%d", var**); will produce a run-time error). For any structure allocating memory, check the boundary of the range and if the index if outside the limits (ex: print out the value of the index: **printf("%d\n", index);**)
3) For logical errors (wrong value), print out the values of variables at periodic places within the code (ex: **printf("%d %d\n", varaiable1, variable2);**) to check if the value was set correctly or not.