COMP 30400 – Programming I & II
Lab 1

1) Install C compiler suite:

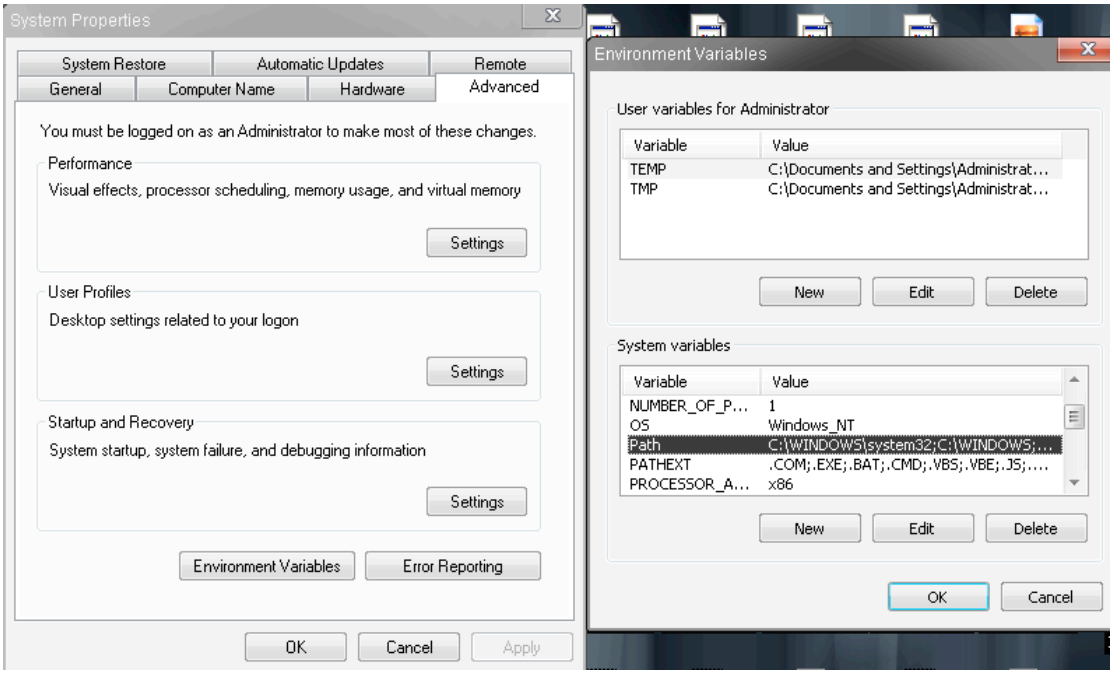
Windows – gcc (MinGW or Cygwin) http://www.mingw.org/

Installing MinGW:

Add `C:\MinGW\bin;` to the PATH environment variable.

On different versions of windows this will be slightly different.

1) Right click "My Computer"
2) Select "Properties" from the popup menu.
3) Select the "Advanced" tab
4) Click the "Environment variables button"
5) In the "System variables" section, select the "Path" variable and click edit.
6) add `C:\MinGW\bin; to the start of the variable`



Download and run the "Automatic installer" from:
http://www.mingw.org

Close all open command prompt windows.

Linux – gcc (any version)

First check if it's already installed (open a terminal and type: gcc –v)
If it says "command not found":
Debian/Ubuntu: (as root in terminal) apt-get install gcc
Gentoo: (as root in terminal) emerge gcc
Fedora: (as root in terminal) yum install gcc

Mac – gcc (as part of XCode) available from:
http://developer.apple.com/technologies/xcode.html

When installed, you should be able to enter the following into a terminal/command prompt window:

gcc –v

And if it's working you'll see an output:

gcc version <gcc version> (<operating system version>)

If gcc is not found, search for where it was installed to, and run it from that directory. Installing gcc can be slightly different on every machine, so it make take some searching to find where it was installed on your machine.

Ensure that you have a working gcc (where you can see the version number from gcc –v) before proceeding to the next step.

1.5) Install a 'good' text editor.

Any text editor is fine, but some are easier to use than others and will have colour (syntax highlighting) support.

Recommended editors:
Windows: Notepad++ http://sourceforge.net/projects/notepad-plus/
Mac: TextWrangler http://www.barebones.com/products/textwrangler/
Linux: Emacs, Gedit, VIM, many other options.

With gcc and a text editor installed we are now ready to program!

2) Create a folder for your programming portfolio.

Inside this folder, each of your programs should have their own folder, where the name of the folder is the program number. Example:

ProgrammingPortfolio
ProgrammingPortfolio/1

3) Portfolio – Program 1
Write the hello world program (C source code files are plain text files that should end with .c):

```
#include <stdio.h>

int main(void)
{
        printf("hello, world\n");
        return 0;
}
```

Save this to a new file: helloworld.c
This file should be saved inside: ProgrammingPortfolio/1

NOTE: If you are using Microsoft notepad, be careful about notepad adding '.txt' to the end of the filename. C source code files MUST end with .c

Compile and run the program:
Use the 'cd' command in the terminal window to change the current directory into the directory with your source code. E.g.: cd ProgrammingPortfolio/1

To compile and link the program (single step):
gcc –o helloworld helloworld.c

This tells gcc to compile the helloworld.c source code file, and to output (-o) a program called helloworld.

Run the helloworld program to verify that it works:
Windows command prompt: helloworld
Linux/Mac terminal: ./helloworld

4) Portfolio – Program 2

Create a new folder for Program 2.
Recall that \n means new line. Create a program which prints:

hello,
world

5) Your portfolio should now look like this:

ProgrammingPortfolio/1/helloworld.c
ProgrammingPortfolio/2/helloworldnewline.c

Make a backup of your portfolio.