

# CSCE 155 – Developing in C on Windows

---

## *Handout*

Developing C programs on the Microsoft Windows operating system is easy, but you need to install a few tools: a plain text editor with code mark-up capabilities, a C-compiler, and the appropriate C libraries that you wish to develop with. This document will walk you through the basic steps to download, install, configure, and use these tools. This tutorial is for your benefit, but we are not able to provide technical support for your particular machine or for the programs mentioned here. We cannot be held responsible for any problems you may encounter while installing or using these tools, so install them at your own risk. There are many possibilities and alternative tools for developing in C on Windows.

## **1. Installing MinGW**

MinGW is a minimal GCC (GNU Compiler Collection) environment for Windows. This step will enable you to compile C programs using the GCC compiler on a Windows machine. Code compiled on Windows is executable on windows (and will not be executable on a non-windows machine without first recompiling).

1. Download MinGW from <http://sourceforge.net/projects/mingw/files/latest/download?source=files>
2. Install it (use the default options) by following the installation wizard.

## **2. Installing GTK for Windows**

Later on, we will be developing programs with a Graphical User Interface (GUI) using a library called GTK. To be able to develop and execute GTK programs you will need the GTK library installed.

1. Download the libraries from [http://ftp.gnome.org/pub/gnome/binaries/win32/gtk+/2.24/gtk+-bundle\\_2.24.8-20111122\\_win32.zip](http://ftp.gnome.org/pub/gnome/binaries/win32/gtk+/2.24/gtk+-bundle_2.24.8-20111122_win32.zip)
2. Open the zip file, and extract the files to C:\MinGW\gtk\

## **3. Setting up MinGW and GTK**

Now that MinGW and GTK have been installed, you need to add them both to your environmental path variable by following these instructions:

1. Right-Click on “My Computer” (or “Computer” for Windows 7 under the start menu) and select “Properties”
2. Click Advanced -> Environment Variables
3. In the box entitled “System Variables” scroll down until you find “Path” or “PATH” and double click it.

4. Do NOT delete anything in the second box. Instead ADD to the end of the text that is already there, the following:  
`;C:\MinGW\bin;C:\MinGW\gtk\bin`  
Note: if you installed either MinGW or GTK in an alternate location, then you'll need to modify the above string accordingly.
5. Click OK, OK and you're done

## 4. Writing/Editing Code

Now you have an environment on your Windows machine that lets you compile code. There is no editor like Pico, Nano, VI, or Emacs built into Windows/MinGW. Remember, you should NOT use a word processor like MS Word to edit code since word processors insert extra characters into the text for formatting purposes.

You have a few options for editing code on your local Windows installation with MinGW, but this tutorial will only walkthrough what is likely the most familiar to you: Notepad++.

1. Download Notepad++ if you don't already have it:  
<http://download.tuxfamily.org/notepadplus/5.9.8/npp.5.9.8.Installer.exe>
2. Install it like you normally would a Windows program (just use the default options if you're unsure of what to do).
3. Run the program from the start menu or its installation location. An editor not unlike Word or Text Edit will appear. You can enter your code here like you otherwise would on the CSE server in pico.

## Writing, Compiling, Running

1. Open a new file in Notepad++ and save it as `hello.c`
2. Type out a simple Hello World program:

```
#include<stdio.h>
int main() {
    printf("Hello\n");
}
```
3. Open a command prompt (click on the start menu, type 'cmd', enter)
4. Navigate to the directory where you saved your file (using the 'cd' command)
5. Compile the source file using:

```
gcc -o hello hello.c
```
6. This will create an executable file called `hello.exe` that can be run (only on) Windows from the command prompt. To run it type:

```
hello
```

As previously mentioned, there are other options for editors on Windows. Emacs has a Windows version available ([Link](#)), as does Vim ([Link](#)). Each of these have a GUI interface (unlike their Unix command line counter parts), which might make learning them a bit easier.

As an aside, you may have noticed that the Windows prompt doesn't have the same commands as a Unix prompt (for example, 'ls' on Unix is comparable to 'dir' on the Windows command prompt). [This](#) page serves as a fairly good introduction to navigating on the Command Line Interface (CLI) on Windows.

## Misc Issues

Some users have experienced issues with this tutorial. Possible solutions (DO THESE AT YOUR OWN RISK) are included:

1. If adding the MinGW directories to your PATH variable don't take effect, try the following:
  - a. Start the Registry Editor: click the start menu and type regedit
  - b. Navigate to the following registry key directory:  
HKEY\_LOCAL\_MACHINE\SYSTEM\CurrentControlSet\Control\Session Manager\Environment
  - c. If the "type" of the Path key is "REG\_SZ" it should be changed to "REG\_EXPAND\_SZ" to do this:
    - i. Right click on the registry screen and select New > Expandable String Value
    - ii. Temporarily name it "FOO"
    - iii. Double click it and cut and paste the value from your Path variable into the new FOO variable.
    - iv. Delete the old PATH variable and rename FOO to be Path
    - v. Restart your computer