```
/* ==========================================================================
  $Day: $ (001)
  $Date: $
  $Creator: Emmanuel Vaccaro $
  $Notice: (C)Copyright 2014 by Emmanuel Vaccaro. All Rights Reserved. $
  ========================================================================== */
```

Hand made hero Day 001 - Set up Workspace in Windows

Required software:

- Visual Studio 2013 (Free Community Edition)
- Text editor (Sublime Text, Brackets or Emacs)
- Command Prompt (Custom Shortcut if needed)

Step 1). Set up virtual drives –

To set up virtual drive, open up your command prompt and enter the following code:

```
-    wmic logicaldisk get deviceid, volumename, description
```

This will get a list of all of the devices connected to your pc will display.

Now get the DeviceID of the drive in which you are going to use to store the workspace. In this particular case, we will be using F:\ which is our 16gb USB flash drive.

In command prompt, you can substitute the drives into a virtual drive using the command "subst".

FACT: 'subst' is a command on the DOS, Microsoft Operating Systems used for substituting paths on physical and logical drives as virtual drives.

Now we're going to substitute the "F:\" directory to the virtual drive "W:\" with the following code:

```
-    subst w: f:\
//    ^       ^  ^
//    |       |  |
//    |       |  Existing drive
//    |       Virtual drive to create
//    Command
```

Now, if we type in the 'subst' command, we will see the following: "W:\: => F:\"

This means that we now have a virtual drive created called "W:\" which was substituted from "F:\" which means we can refer to "F:\" in future by using the virtual drive "W:\".

Step 1.5). [Optional]

The problem with this setup is the fact that it will need to be done every time you close your computer and start it up again. What you can do as a work-around is place a new .bat file inside of Windows' Startup folder located in Start > All Programs > Startup.

Note: The Startup folder is a directory in which Window's will execute any programs (.exe's) or any files that can be executed.

In this folder, we will create a new text document file and call it 'startup.bat'.

Next, you will need to open the file in notepad and type the following code:

- @echo off //<--- Tells the command script to not display the code after this line.
- subst w: f:\

Now upon each start up of Windows, it will automatically substitute your physical drive over to a new virtual drive under 'w:'

After the virtual drive has been created, we are going to create a custom command prompt shortcut to always start in the "W:\" virtual drive.

- Navigate to Start > All Programs > Accessories and hold down CTRL whilst clicking and dragging a shortcut of Command Prompt over to the desktop.
- Once you have a shortcut to Command Prompt on your desktop, Right-Click on it and select Properties.
- Underneath 'Start in:' replace the text with 'W:\'.

Now the custom command prompt will open up in the virtual drive 'W:\'

Step 2). Set up Workspace folders –

In your explorer, you should now see this drive layout:

- F:\ (Physical Drive)
- W:\ (Virtual Drive)

To set up the workspace folders, create the following folder structure within drive "W:\":

```
- W:\
   ∟ build
   ∟ handmade
          ∟code
          ∟data
          ∟misc
```

Step 3). Setting up Shell.bat & <TextEditorName>.bat –

In this step, we will be making a 'shell' batch file as well as a <TextEditorName>.bat for your text editor. This will allow us to use command prompt to access the text editor easier throughout the project. As well as performing other rudimentary tasks.

First, we will need to create a .bat file for your text editor:

- Open up the 'misc' folder located under "W:\handmade\misc", create a new text document and name it '<YourTextEditor>.bat'. In this case we will be using Sublime Text, so it would essentially be 'sublime.bat'.
- In notepad, we will need to add some batch code to locate our text editor installation directory:

```
    "C:\Program Files\<TextEditor>\<TextEditor>.exe"
               //         ^           ^
               //         |           |
               //         |           |
               //         Where '<TextEditor>' is the name of the text editor you are using
```

Since we are using Sublime Text, our code would look like this:

```
 @echo off <--- Turns off printing lines to console
 "C:\Program Files\Sublime Text 2\sublime_text.exe" %*
//        ^                ^
//        |                |
//        |                Command for passing command line arguments (also %1, %2 adds 1
//        |                or 2 empty string params)
//        |                (e.g. sublime test.txt) which opens sublime with txt file
//        Text editor installation directory
```

Note: Depending on your text editor, it may be installed in a different directory. Search through either 'Program Files' or 'Program Files (x86)' to locate the appropriate installation directory of your text editor.

FACT: The 'shell script' is a computer program designed to run by the 'Unix shell', a command line interpreter. Typical operations performed by shell scripts include file manipulation, program execution, and printing text.

Now that our text editor .bat file is created (the file we will be calling every time we want to open the text editor inside of the console) we will need to create a 'shell.bat' file underneath the directory "W:\handmade\misc".

- Open up Command Prompt and navigate to the "W:\" by going - cd W:\ Then type in the following code into CMD:

```
            cd handmade\misc && notepad shell.bat
//          ^  ^             ^  ^       ^
//          |  |             |  |       |
//          |  |             |  |       File to create, such as a batch file (.bat)
//          |  |             |  Text editor (e.g. Notepad, Sublime text, Brackets etc)
//          |  |             AND operator (runs following command(s))
//          |  Folder directory in drive
//          Change Directory Command
```

NOTE: If you get the warning "Cannot find the file shell.bat", select YES as it will create a new one in that directory.

The directory "W:\handmade\misc" should now contain a file named 'shell.bat'. This is basically going to be our script to tell Windows that we want to add another path to the "PATH" environment variable.

Note: You can see a list of environment variables by navigating to:

- (WINDOWS 7) Start > Right-Click on Computer > Properties > Advanced System Settings > Advanced Tab > Environment Variables... and ther will be a list of variables.

Now that you have created a 'shell.bat' file, you should have the text editor (which you specified in the above mentioned code) opened up and ready for edit. Within the text editor, type in the following code (Ignore the commented lines beginning with '<---'):

- @echo off
- set path=w:\handmade\misc;%path% <--- Sets 'path' environment variable to OUR path and it's OLD path (%path%)

Now the path variable should be set to our misc folder and whenever and wherever we wish to open up our text editor in Command Prompt we will type in the name of the .bat file we created earlier:

- sublime

Sublime text can even open up specific files by going:

- sublime test.txt

Which then opens up sublime and the directory of the test file I wish to add. Since we are already in the directory "W:\handmade\misc" and the file 'test.txt' is within this specific folder, this syntax is completely valid.

If we wish to add files that are in other locations, for example, one folder out of the current directory, we'd have to specify this way:

- sublime ..\test.txt

```
//          ^
//          |
//          The '..\' part of the command tells cmd prompt to go one folder up a directory
```

This directory means it is located within "W:\handmade" instead of "W:\handmade\misc".
We now have a way to call sublime text (or any other text editor) anywhere in cmd prompt.

Step 3.5). [Optional][Recommended]

We need to set up the custom command prompt to call the 'shell.bat' file every time we open up command prompt.

The reason is, every time we close command prompt, it doesn't save our changes made to the Environment Variables unless we actually change it manually (which I don't recommend doing for various reasons).

This is very simple and to do this we need to find the Command Prompt shortcut we made on the Desktop earlier, Right-Click and select Properties.

Under "Target:" type the following code:

```
    -   %windir%\system32\cmd.exe /k w:\handmade\misc\shell.bat
//                                ^  ^
//                                |  |
//                                |  Directory in which shell.bat is under on the
//                                |  virtual drive
//                                Runs a command and then returns back to the CMD prompt
```

In turn, what this will do is run the shell.bat before entering the cmd prompt. Meaning, you don't have to manually call it every time you open a new Command Prompt.

Step 4). Creating our cpp files (for C/C++)

Now that we have our text editor set up, we can start to create source files. In Command Prompt, navigate to "W:\handmade\code" by typing in the following code:

- cd W:\handmade\code

Once you are in this directory, we need to create a cpp file. Thanks to what we did earlier with environment variables, we are definitely able to do this by using the following:

- sublime win32_handmade.cpp

NOTE: This will NOT save the file straight away upon opening the file in sublime text. To save, you need to hit CTRL-S (together) for the file to be saved in the location you opened it. In this case "W:\handmade\code".

Once opened the file in sublime text, hit CTRL-S as mentioned above, to ensure it is saved in the directory you opened it. Once inside of sublime, type the following cpp code:

```
#include <windows.h>

int CALLBACK
    WinMain(HINSTANCE hInstance,
    HINSTANCE hPrevInstance,
    LPSTR lpCmdLine,
    int nCmdShow)
{
    MessageBox(0, "Hello World!", "HandMade",
                MB_OK|MB_ICONINFORMATION);
    return(0);
}
```

All this code will do, is open up a message box displaying "Hello World!" and titled "HandMade".

NOTE: If you are uncertain about how this code works, Casey Muratori explains this in thorough detail on his YouTube video at:
https://www.youtube.com/watch?v=Ee3EtYb8d1o&list=UUaTznQhurW5AaiYPbhEAKA&index=12

Once you have finished writing in the cpp file. Hit CTRL-S again to save your progress and exit sublime text.

Step 5). Edit Shell.bat file (To Enable VS compiler)

Now that we have established where our code is going to do, we need to enable the compiler and run it whenever we wish to build our project. To do this, we're going to need to locate Microsoft's VC compiler library.

If you have installed Visual Studio 2013 (Free Community) correctly, then this will be located under:
"C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\vcvarsall.bat".
Double check that this directory and file exists by navigating to that folder. If it does NOT exist, then you will have to re-install Visual Studio 2013 (Free Community):
http://www.visualstudio.com/

Once you have verified it does exist, then open up the custom Command Prompt and type in the following command:

- sublime handmade\misc\shell.bat

When sublime (or your own text editor) opens up, add the following line underneath what's already written in shell.bat:

```
-     call "C:\Program Files (x86)\Microsoft Visual Studio 12.0\VC\vcvarsall.bat" x64
//    ^                                ^                                           ^
//    |                                |                                           |
//    |                                |                                           Sets the
//    |                                |                                           compiler library
//    |                                |                                           to 64-bit
//    |                                Directory where VS compiler library is located
//    Command used to specifically run a call to a .bat file or subroutine
```

Step 6). Create a build.bat file (to compile project)

We are going to need to create a build.bat file so that if we wish to build our project throughout the course of using Command Prompt, we can by just typing 'build'. To do this, we will need to create the file in the appropriate file.

Open the custom Command Prompt and navigate to 'misc':

- cd handmade\misc

Next, create the build.bat file:

- sublime build.bat

Now, we need to write a fair bit of code for the compiler to work as expected:

```
    @echo off

    mkdir ..\..\build //<-- 'mkdir' Creates a new directory ('Build' folder)if the directory does not exist
    pushd ..\..\build //<-- 'pushd' Stores the name of the current dirtectory for use by 'popd' later

    cl /Zi ..\handmade\code\win32_handmade.cpp user32.lib
//  ^  ^                 ^                         ^
//  |  |                 |                         |
//  |  |                 |                         Adds windows.h library dependency for build
//  |  |                 Directory where our main cpp file is located
//  |  (Debug Information Format) Zi produces a program database (PDB) - /Zi implies /Debug
//  |  Other formats include: Z7, Zi, ZI (Look here for more details:
//  |  http://msdn.microsoft.com/en-us/library/958x11bc.aspx)
//  Uses the compiler (cl.exe) to compile and link source code files into apps, libraries and DLLs

    popd
//    ^
//    |
//    'popd' Uses the saved directory in 'pushd' command (In this case, build will be saved in pushd's
//    saved location)
```

Done! You are now able to call build inside of the custom Command Prompt whenever and wherever you wish. Your build files will then be located in "W:\build".