University of Florida
Electrical and Computer Engineering
EEL3923 & EEL4924
Microchip MPLAB and C18 compiler setup.
Start the MPLAB setup program.

Click next.
Accept and click next.
Click next.
Click next.
Click next.
Click next.
Click next.
The software will now install.
Click No.
Click No and then the Finish button, we will restart later.
Next we will install the compiler. Navigate to the MPLAB-C18-Lite directory and start the application.

Click Next.
Select I Accept and click next.
Click next.
Click Next.
Select all four boxes and click next.
Select all four boxes and click next.
Click next.
The software will now install.
Next click Finish.
Unselect the boxes you don’t want to open and read.
Now reboot.

After you log in this window will pop up.
You can just close it.

Click this icon to start MPLAB IDE.
The software is now installed.
The next section will cover the setting up and building of a new project.
This is what you should see when the program starts.
In the toolbar select Project and then Project Wizard.
Click Next.
Select PIC18F1320.
Make sure that the Active Toolsuite is Microchip C18 Toolsuite.
Also make sure that the toolsuite contents and the locations match. Click Next.
Next Select the Program button from the toolbar and click on Project wizard. Enter the name of your project. 18F1320Wizard in this case.
You can also select the project location with the Browse button.

This is where you can add your main.c file.
This is the Main.c file that is used for this example.
You must also add the linker file to match your processor.
In this case it is a 18F1320. After clicking the 18F1320_g file click Add.
Click Finish.
Next select the Project tool box and then select the Set Language Tool Locations tool.
Select the Microchip C18 Toolsuite toolbox.
Open the executables box.
Check that the executables are pointed to the correct locations as above.
Next select the project tool and then select the Build Options…Project.
Now select the Directories tab. Then select the Include Search Path in the drop down box. Next click new.

Click this button to open up the Directory to select the files.
Set the search path for the include and the library to match the above. Click apply and then OK.

Click the Build All button shown above. This will build and compile your project.

The last line in the output box should read “BUILD SUCCEEDED”.
Select the Debugger tool as above.
Select the View tool and then select the Watch option. This will open a new window.
In the watch window you can select the symbol section and select a variable to monitor. X has been selected here. After the variable is selected click the Add Symbol button.
In the watch window you can select the SFR section and select a SFR to monitor. Port B has been selected here. After the SFR is selected click the Add SFR button.
On the top of the page find the run button and click it.

Pause the sim. You should be able to see your port and variable data in the watch window.
Next select the Debugger tool and click the settings button.
Click the UART1 IO tab and set the boxes as shown above. This will route the printf data to the Output screen.
Running your program on your target device.

Select the debugger tool and click on Select Device and then 7PICkit 2.
Select the debugger tool again and then click the settings button.
Adjust your settings to match the above window.
Your program should be running on your hardware now.
If you right click in your main.c window you can set a breakpoint.
You will see a red circle with a “B” marking the breakpoint. The program will run until it reaches this point and then it will stop. To continue click the run button again.

You can single step your program by pressing the step Into button here.