Coding Remotely (incl. Windows/Mac)

SSH:

SSH is the typical way to access Unix and Linux machines remotely. If you already know how to SSH, the important detail is this: You can only do the work for this lab on the machines in the lab, but, you *can* SSH into those machines. Check out the stickers on the machines or login screens to find their hostnames.

Importantly, you *cannot* do work for these labs on `lore.cs.purdue.edu`.

You can remote on any of the machines in LWSN B146, B148, B158.
-mooreXY
-sslabXY

Windows:

Text Based

My favorite way to write, test, and run C programs from my Windows PC is PuTTY. PuTTY is a program to connect to Linux/Unix PCs and do things in command line with them. You can open code in Vim, Emacs, jedit, etc., compile it, and run it just like to would on a Linux PC in Lawson.

You can download PuTTY here: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

After downloading PuTTY, you type the address of the computer you wish to access in the Host Name field. Some example addresses are "pod2-2.cs.purdue.edu", "moore10.cs.purdue.edu", "pod4-3.cs.purdue.edu", and "moore22.cs.purdue.edu". You can connect to any of the Lawson Linux and Unix PCs in the labs.

After that, hit the Open button on the bottom right hand corner, and a command prompt will open up and ask you first for your purdue username, and then for the password you normally use to connect to Linux PCs in Lawson. You can view more advanced PuTTY options here: http://www.jfitz.com/tips/putty_config.html

I would suggest running multiple PuTTY windows at a time so that you can multitask just like you would in Lawson.

Once you are done editing and testing your program and you wish to submit it, follow Tyler Hoffman's guide here https://piazza.com/class#spring2012/cs240/71 to move the files to your desktop computer and submit them from there.

Linux Remote Desktop (GUI)

Another edit: If you would like to use visual based editors on your Windows PC and would still like to be able to test your code easily and from your own computer as well, this section is for you. Here is an example of what you can do. http://i.imgur.com/wl3Kq.png

To go about doing this, navigate to http://sourceforge.net/projects/xming/ to download and install Xming. It will allow you to see windows from the Linux desktop. Use all of the default configurations unless you know what you are doing. When it is installed, it will show up in the taskbar on the bottom right. You can either keep it running indefinitely, or only use it when you want to code remotely (only needed for this GUI demonstration)

You still need PuTTy, so follow Caleb's link from above and install that.

- Type in the server as a lab computer (not Lore).
- Click on data in the left column to type in your username
- Open up more options on 'SSH' and click on X11. Check Enable X11 forwarding.
- Click on session in the left and then click on Default settings (only once).
- Click save. This saves your preferences so you don't have to constantly type things in.

Now once everything is ready, click on Open, type in your username password. You should be logged into a lab computer now.
If you want to run gedit remotely, type in

```
$ gedit &
```

This will bring up gedit with the ability to click open and open up files that are stored on your CS account. You can even compile from the PuTTy terminal.

FAQ:

- If you get an error saying the display could not be found, make sure Xming is running by going to start->Xming-> Xming.
- If something doesn't work, the best way to shut off Xming is to right click in the task bar on the icon and click exit. Then start over.

Mac OS X (Snow Leopard and Lion):

Text Based

If you want to use vim or emacs, the simplest way to connect in that way is to pick a lab machine to ssh into and load up vim. Make sure it's a lab machine and not Lore because Lore is Solaris (not guaranteed to run code properly) Basically the same thing as PuTTy from above.

```
$ ssh <username>@<labcomp>.cs.purdue.edu
$ cd cs240
$ vim file.c
```

Linux Remote Desktop (GUI)

If you are wanting more of a GUI way to program remotely, then LISTEN UP! I never did like using the virtual machine, because I had to run Mac OS along with a Linux distro virtualized. I have a better way.

Load up terminal. Type in the command.

```
$ ssh -X <username>@<labmachine>.cs.purdue.edu
ex. $ ssh -X smithrob@moore14.cs.purdue.edu
```

NOTE: You will need the -X. It's important. Also, try to pick different numbers of a lab machine. If everyone logs into number 14, it will get overloaded.

Next, type this command:

```
$ gnome-session
```

You will now see come crazy things going on. You will soon have a linux desktop intertwined with your Mac interface. You can load up programs and run them side by side with your Mac programs. (You can even copy paste between them too). To compile, load up the linux terminal and continue on as if you were in the lab.

If you just want to load gedit for instance, one you have completed the first command, instead of 'gnome-session', you can type in

```
$ gedit &
```