Windows

This guide will only cover using the Windows Subsystem for Linux (WSL). There are a lot of other ways to get set up with Windows. The alternative I would suggest is MSYS2 with 64-bit mingw; it's a more complicated setup but is much more "Windows native."

- Have a relatively up to date version of Windows 10
    - 2004+ preferred
- Open the Start Menu and search for "Turn Windows Features On/Off"
- Enable Windows Subsystem for Linux
- (Build 2004+) Enable Virtual Machine Platform
- After setup completes, restart
- (Build 2004+) From Powershell: `wsl --set-default-version 2`
- Open Microsoft Store, search for Linux
- I recommend installing Debian
    - Ubuntu is also a good choice
- First boot takes a while
- After you select username and password, run the following commands:
    - `sudo apt update`
    - `sudo apt upgrade -y`
    - `sudo apt install build-essential git vim -y`
        - vim is optional, but I do recommend it
        - Installing git in the WSL environment will be for using git with your code, to keep up to date with lectures, Windows also needs git. It can be installed from git-scm.com
- VS Code has an extension for integrating with WSL (search for Remote WSL)
    - Subsequent extensions should be installed in the WSL

macOS

- Install the XCode Command-Line Tools (CLT)
    - 1st method: Install XCode from the App Store, run it, CMD + , (comma) to get to settings, Locations tab, ensure an XCode CLT is selected
    - 2nd method: From Terminal app run: `xcode-select –install`
- Install homebrew
    - Follow instruction at https://brew.sh
- Install gcc and git
    - `brew install gcc git`
- Verify that the homebrew gcc is the default
    - From Terminal app: `which g++`
    - If a homebrew directory isn't showing, you will need to edit the file `/etc/paths`