Lecture Notes: Tools

Week 1 – Lecture Notes Combined
Lecture Notes adapted from [Nighthawkcoders Tools | Portfolio 2025]

To start programming, you need to have tools ready on your system. This document will tell you which tools to install and will have linked guides to official documentation for installing tools. This is not an instruction manual, you will be expected to use the links to install the tools. This is a simple list of what to have for tools.

MacOS

Basics

To install tools on MacOS, first, we will install **homebrew**, the package manager for MacOS.

To install homebrew, go to https://brew.sh/ and follow the installation guide. The guide clearly states that the installer will give you instructions, so when the installer asks you to add homebrew to your PATH, you must copy-paste the commands to add homebrew to your PATH or else the installation will not work and you will not be able to run the command.

Lecture Lesson 1:
Learn how to read error statements and outputs of commands. More often than not, they will tell you the problem

Then, you will install **Visual Studio Code**, the code editor for this class. The code editor can be found here: https://code.visualstudio.com/download.

Finally, although most MacOS systems have it, you should install **git** (if you want, you can install its counterpart, GitHub Desktop). You can install git here: https://git-scm.com/downloads.

The basic tools have been set up. Now, we will be moving onto more niche tools in the combined section.

Windows

Basics

For Windows, the first thing we want to do is to install **Windows Subsystem for Linux (WSL)**, which will allow us to have a Linux system on our Windows machine. If you want to know why, do some outside research. Both operating systems have their own advantages. Here is the installation guide: https://learn.microsoft.com/en-us/windows/wsl/install. Make sure you install ubuntu. If you want to know the different types of linux distributions and their advantages/disadvantages, don't hesitate to do some research.

Lecture Lesson 2:
lmgtfy. It's really nice when a student researches their own solution to a problem and builds off of it. This is an extra seed opportunity if you research, document, and fix the issue.

Lecture Lesson 2 WARNING:
BE CAREFUL when trying to solve problems on your machine. Make sure you know what you are doing, and if you do not or need help, do not hesitate to ask your TA's or a teacher. If you do not know what you are doing, you have a chance, depending on what you are doing, to break your project or even break your system.

Then, you will install **Visual Studio Code**, the code editor for this class. The code editor can be found here: https://code.visualstudio.com/download.

Finally, although most Windows systems have it, you should install **git** (if you want, you can install its counterpart, GitHub Desktop). You can install git here: https://git-scm.com/downloads.

The basic tools have been set up. Now, we will be moving onto more niche tools in the combined section.

Kasm

Basics

All Kasm images come with tools already installed on your system ready to go. Additionally, Kasm supports persistent profiling, meaning that the applications that you install on your system are backed up (make sure you enable persistent profiles). You do not need to worry about installing the basics for Kasm.

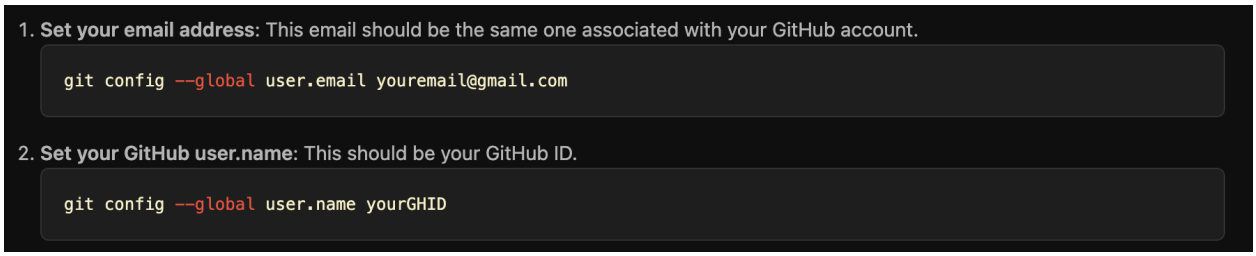
Student Blog Installation – Combined Installation

To run a blog, we need to have applications like bundle, python, and ruby, all of which can be found and installed here:

Program	Install Link
Python	https://www.python.org/downloads/
Ruby	https://www.ruby-lang.org/en/documentation/installation/ (use ubuntu)

To install bundle, we then run `bundle install` in our portfolio folder to install all dependencies. Also, make sure you run `./script/activate.sh` (may need to use `chmod +x scriptname.sh` to run the scripts).

Next, run your Git Identification.



Finally, run the following two commands:

- pip install -r requirements.txt
- bundle install

How to Run the Servers

There are two ways to run servers. Run one of these commands in the terminal. Ctrl-C to quit:

- bundle exec jekyll serve
- make