What Is a Terminal & Installing Ubuntu with WSL

AiCE Software Engineering Foundations — Setting Up Your Developer Environment

Before you can write a single line of code in this program, you need a working terminal on your own machine. This lesson explains what an operating system and a terminal actually are, why this course uses Ubuntu rather than plain Windows, and walks you through installing a real Ubuntu terminal on your computer using a tool called WSL.

1. What Is an Operating System?

An operating system is the software that manages your computer's hardware and allows every other program to run. Your laptop has a processor, memory, and storage, but none of that hardware knows what to do on its own. The operating system sits between you and the hardware, translating your clicks, your typing, and your commands into something the machine can actually act on. Without an operating system, a computer is just a box of parts with nothing meaningful to do.

Ubuntu, Windows, Kali, macOS, Android, and iOS are all examples of operating systems. Android and iOS run on mobile phones; Ubuntu, Windows, Kali, and macOS run on computers.

Windows is the most widely used operating system among everyday users because of its clean, familiar interface. But a programmer cares less about how good-looking a tool is and more about how effective it is. Windows works well for day-to-day tasks, but for programming specifically, a family of operating systems called Linux distributions tends to perform better — Ubuntu and Kali are two well-known examples.

> **Ubuntu vs. Kali**
> *Kali is built primarily for cybersecurity work. Ubuntu is built for general-purpose programming, and it's what the overwhelming majority of programmers around the world use — including throughout this course.*

2. Three Ways to Run Ubuntu Alongside Windows

Since this course runs on Windows laptops but uses Ubuntu for coding, you need a way to access Ubuntu without giving up Windows entirely. There are three approaches:

Option 1 — Dual Boot (Full Ubuntu as a Second OS)

Installing Ubuntu as a completely separate operating system alongside Windows. Every time you switch the laptop on, you choose which one to boot into. This gives you the full Ubuntu experience, but it's risky — the installation process can wipe your existing Windows files if something goes wrong. This course does not use this method.

Option 2 — A Virtual Machine (Ubuntu Running on Top of Windows)

Installing software such as VirtualBox lets you run a complete Ubuntu operating system inside a window on your existing Windows installation. Windows is called the host operating system; Ubuntu, running inside it, is called the virtual operating system. This gives you the full Ubuntu experience too, but it demands a lot from your laptop — at minimum a Core i3 processor (3rd generation or newer), 4GB of RAM, a processor speed of at least 1.8GHz, and hardware support for a feature called virtualization. Older laptops often can't run this smoothly. This course does not use this method either.

Option 3 — WSL (Windows Subsystem for Linux)

WSL lets you run an Ubuntu terminal directly on top of Windows through lightweight virtualization, without the weight of a full virtual machine. As long as your laptop supports virtualization, this works on ordinary hardware without slowing your laptop down. The one trade-off is that WSL gives you only the terminal functionality of Ubuntu, not its full graphical desktop — which is exactly what makes it lighter and simpler than the other two options, and exactly what a programmer needs.

> **This Is the Method This Course Uses**
> *WSL is the simplest, lightest, and most widely used method among working programmers. It is what the rest of this program assumes you are using.*

3. GUI vs. Terminal — Why Lose the Graphical Interface?

GUI stands for Graphical User Interface — the familiar, point-and-click way of using a computer: icons, windows, buttons, a mouse pointer. It's friendly, which is why beginners and everyday users prefer it. A terminal, by contrast, is entirely text-based — you type commands, and the computer carries out exactly what you typed.

It's reasonable to ask why a programmer would deliberately give up the friendly GUI for a screen of plain text. The answer is that the terminal gives you speed, precision, and control that a GUI simply cannot match. Many of the tools, frameworks, and workflows you will use throughout this course — and throughout your career — are built to be operated from the terminal. Learning

to work confidently there is one of the first real steps that separates someone who merely uses a computer from someone who can truly command it.

4. Checking Whether Your Laptop Supports Virtualization

Before installing WSL, confirm your laptop supports virtualization:

1. Press Ctrl + Shift + Esc together to open Task Manager.

2. Click the Performance tab.

3. Look for the Virtualization line.

Status Shown	What It Means	What To Do
Enabled	Your laptop is ready	Skip straight to installing WSL
Disabled	Supported, but switched off	Enable it in your BIOS setup (covered below)
Not Supported	Your hardware cannot run WSL	Use dual boot instead, or use PowerShell (covered in a later lesson)

5. Enabling Virtualization in the BIOS

If your laptop shows Virtualization as Disabled, you'll need to enable it in the BIOS setup — a low-level configuration screen accessed before Windows even starts loading. Restart your laptop, and as it powers back on, repeatedly press the key your manufacturer uses to enter the BIOS. This varies by brand:

Common Key	Typical Brands
Esc or F10 or F12	HP, Dell
Delete	Lenovo (varies by model)
F2	Many other manufacturers

> **Trial and Error Is Normal**
> *If a key doesn't open the BIOS, the laptop simply boots normally into Windows — switch it off and try again with a different key. If none of the common keys work, a computer technician can help you locate and enable virtualization.*

Once inside the BIOS, the exact menu layout differs by manufacturer, but the setting you're looking for is usually under an Advanced or System Options section, labeled Virtualization Technology (VT-x). Enable it, save your changes, and exit — the laptop will restart. Boot normally this time, then reopen Task Manager ’s Performance tab to confirm Virtualization now shows Enabled.

6. Installing WSL

With virtualization enabled, installing Ubuntu itself takes a single command. Open PowerShell — the terminal built into every Windows computer by default — by searching for it in the Start menu.

```
wsl --install
```

Press Enter, confirm if prompted, and make sure your laptop stays connected to the internet. The installation happens in two parts. After the first part finishes, Windows will ask you to restart your laptop. Go ahead and restart it. What happens after the restart depends on your version of Windows:

On Windows 11, the terminal should reopen automatically after your laptop boots up and continue the second part of the Ubuntu installation on its own. Just keep your internet connection active and let the installation finish.

On Windows 10, the terminal may not reopen automatically. After your laptop boots up, search for Ubuntu in the Start menu and open it. Once Ubuntu opens, it will continue the installation automatically. Keep your internet connection active and wait for the installation to finish.

7. Configuring Your Ubuntu Account

Once installation finishes, Ubuntu asks you to create a username. It typically suggests one based on your Windows account name — but reformatted to fit programming conventions: all lowercase, no spaces, words joined with an underscore or hyphen instead.

> **Why No Capitals or Spaces?**
> *In programming, usernames and identifiers conventionally avoid capital letters and spaces. Where a space would normally separate two words, an underscore or hyphen is used instead — which is why 'Adesina Olagunju' becomes something like adesina_olagunju as a suggested Ubuntu username.*

Accept the suggested name or type your own, following the same convention, then press Enter. Next, Ubuntu asks you to set a password. As you type it, nothing will appear on screen — not even asterisks. This is normal; the terminal is capturing every keystroke silently. Type carefully, choose a password you won't forget, and press Enter. You'll be asked to retype it once to confirm.

With that, your Ubuntu terminal is installed, configured, and ready to use.

8. Your First Commands

Confirm everything works by running a few basic commands:

```
ls
clear
sudo apt update
```

ls lists the contents of your current directory. clear wipes the screen for a fresh view. sudo apt update refreshes Ubuntu's list of available packages — and it's the first time you'll be asked for the Ubuntu password you just created. As before, nothing will appear as you type it; that's expected.

> **What Is a Package?**
> *A package is simply a piece of software installed inside the terminal — the terminal's equivalent of an app on your phone or a program installed through a graphical installer. Anything you install via the terminal going forward, you'll hear referred to as a package.*

To close the terminal at any time, type exit and press Enter. To reopen it later, search for “Ubuntu” in the Windows search bar — it will appear as an installed application from now on.

Key Takeaways

WSL is the lightweight method this course uses to run a real Ubuntu terminal on top of Windows, chosen because it works on ordinary hardware without the risk of dual-booting or the overhead of a full virtual machine. Getting there means confirming virtualization is enabled — in the BIOS if necessary — then running a single wsl --install command and configuring a username and password that follow programming naming conventions. With Ubuntu installed and your first commands run, the next lesson covers installing the essential packages — vim, git, python3, and pip — every one of them will build on throughout this course.