

CSCI 1300 CS1: Starting Computing

Naidu/Yeh/Hoefer - Fall 2022

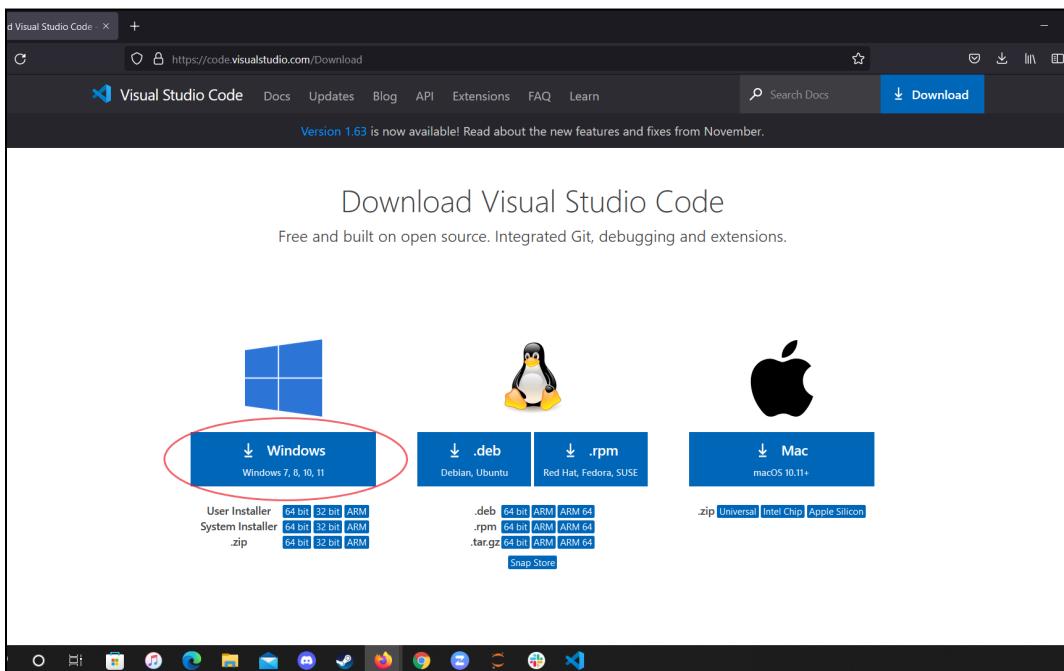
Visual Studio Code - Windows

You will use Visual Studio Code (VS Code) to write and execute your programs locally.

Important: Before proceeding with this document, make sure that you have run Windows Update within your Windows 10 or 11 environment. You must have the latest updates installed.

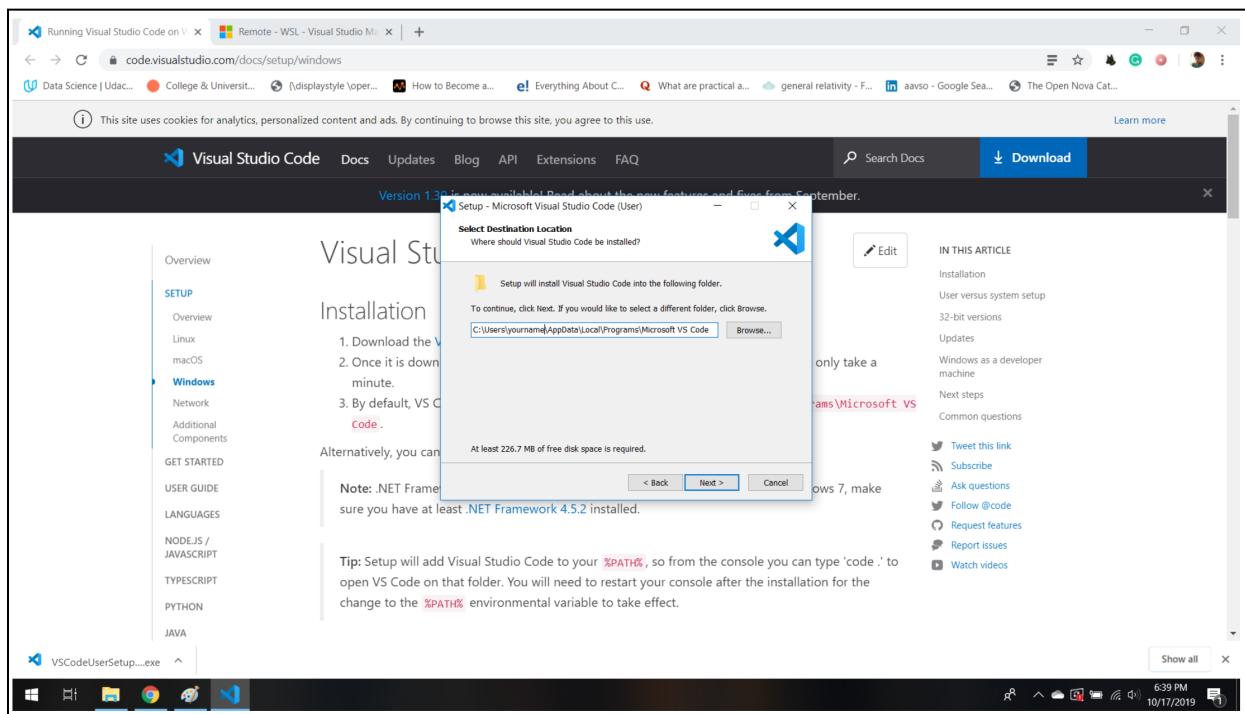
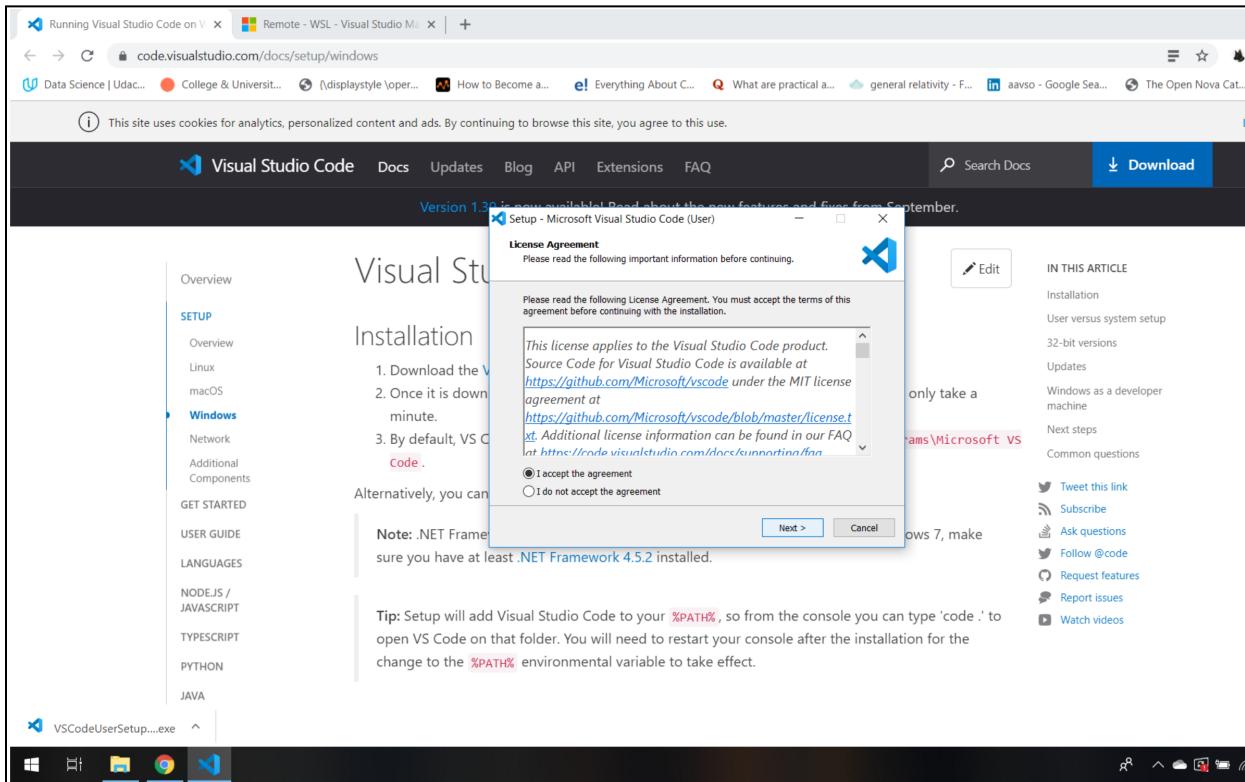
Windows Installation Guide (Part 1)

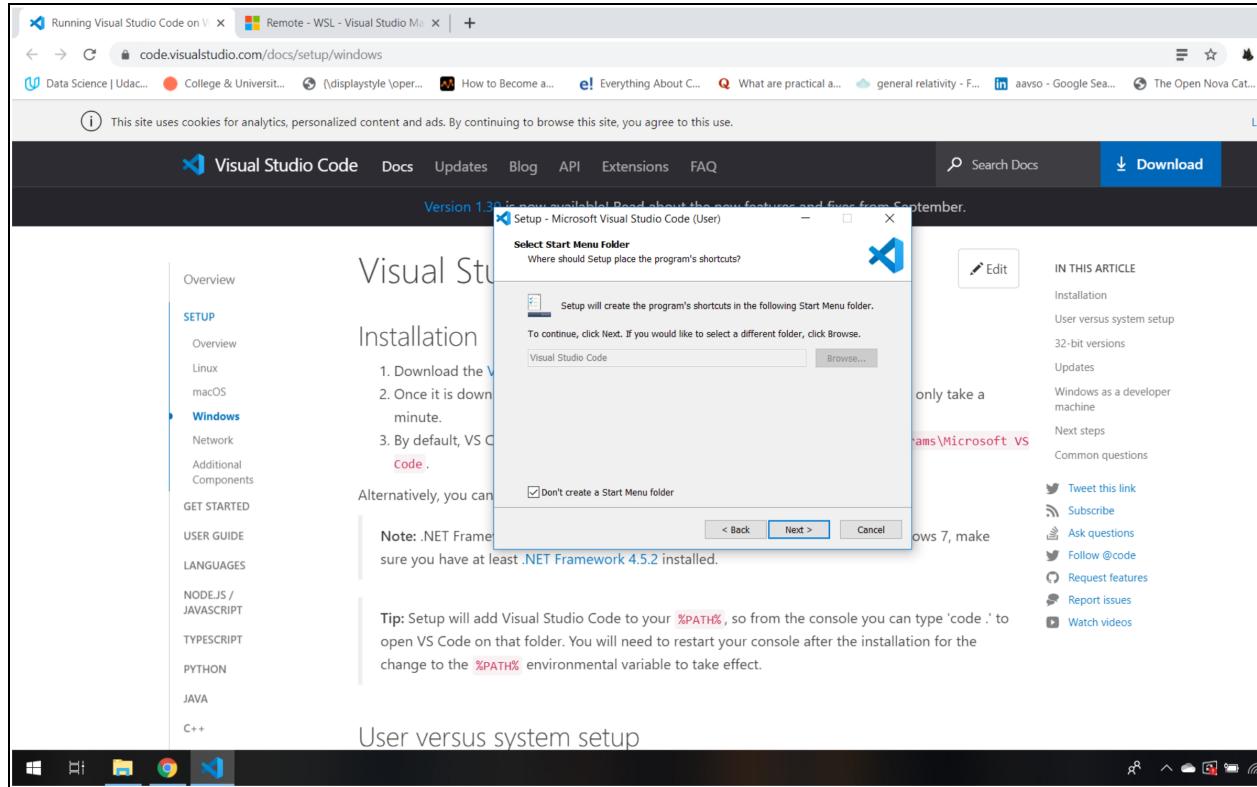
Step 1: Go to the VS code [download page](https://code.visualstudio.com/Download), and download for Windows.



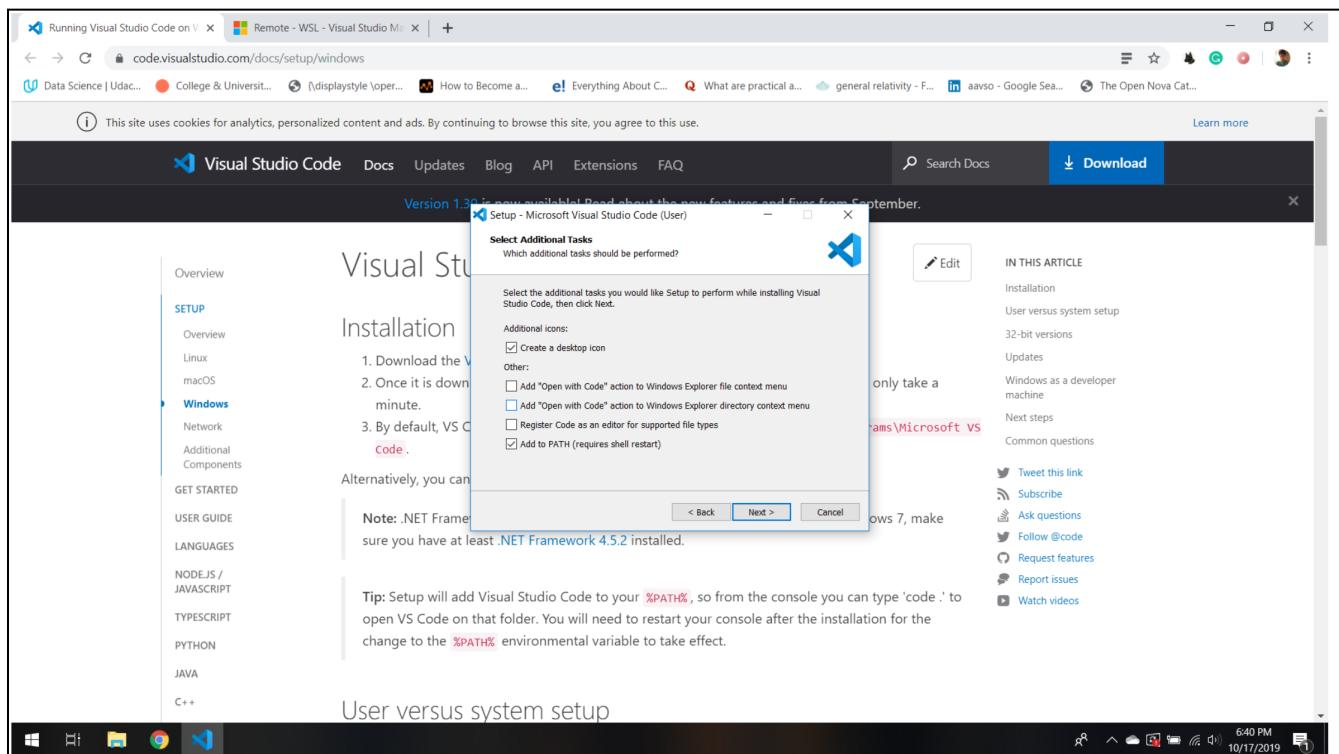
The screenshot shows the "Getting Started" documentation page. On the left, there's a sidebar with links like Overview, SETUP, GET STARTED, USER GUIDE, LANGUAGES, NODE.JS / JAVASCRIPT, TYPESCRIPT, PYTHON, JAVA, C++, CONTAINERS, DATA SCIENCE, AZURE, and REMOTE. The main content area has a green header box saying "Thanks for downloading VS Code for Windows!" with a link to a direct download. Below it is the "Getting Started" section, which introduces VS Code as a lightweight editor for Windows, macOS, and Linux, supporting various languages and extensions. It includes a screenshot of the code editor showing some JavaScript code. To the right, there's a sidebar with links for "GETTING STARTED" (VS Code in Action, Top Extensions, First Steps, Keyboard Shortcuts, Downloads, Privacy) and social media links (Twitter, YouTube, etc.).

Step 2: Run the installer. Accept the user agreement and proceed to follow the steps for installation

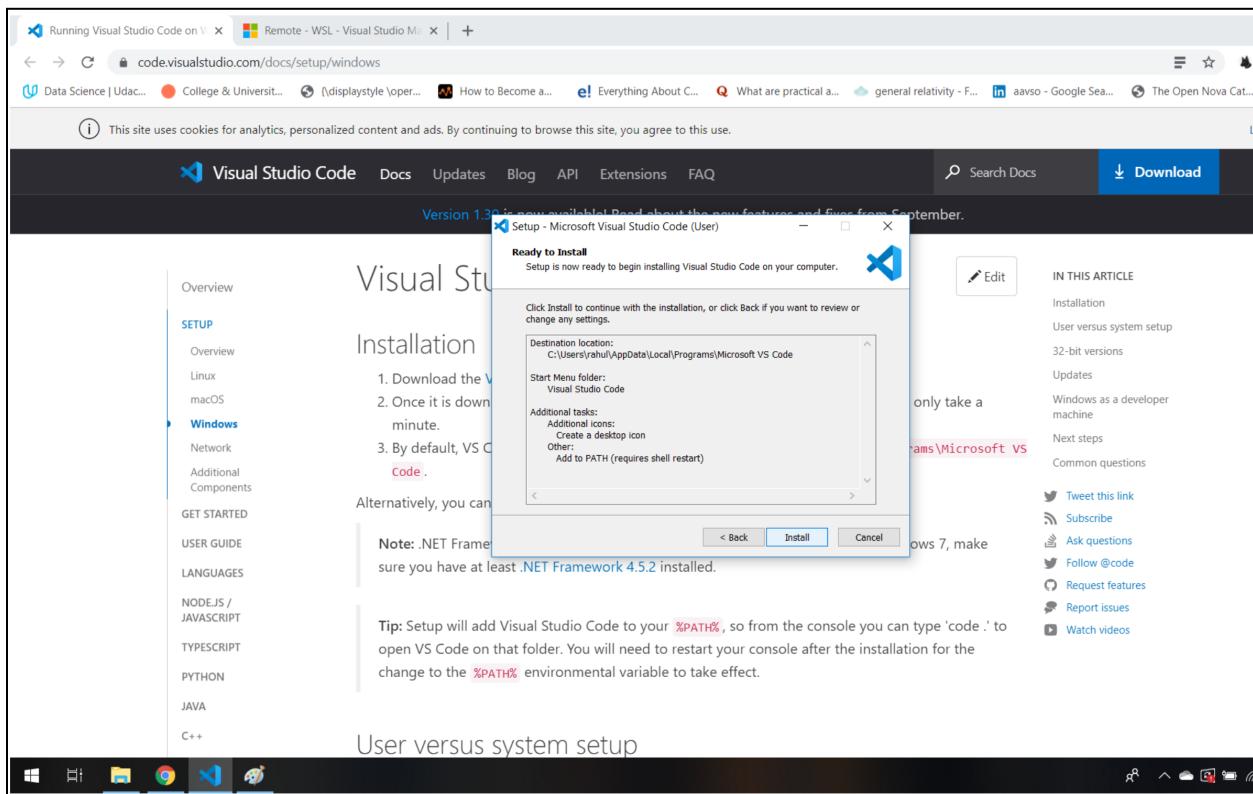




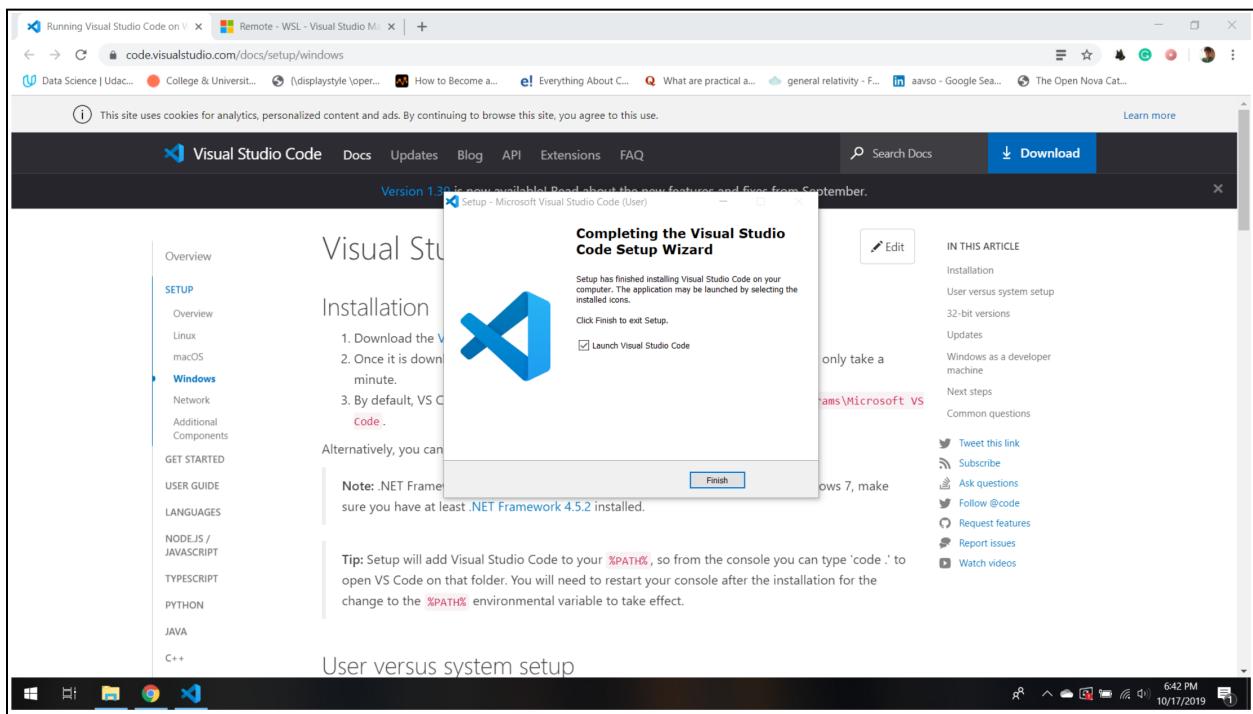
Step 3: Choose Options based on your preferred settings. If you're not sure, you can stick with the ones used in this tutorial.



Step 4: Click on Install and wait for Visual Studio Code to finish installing.



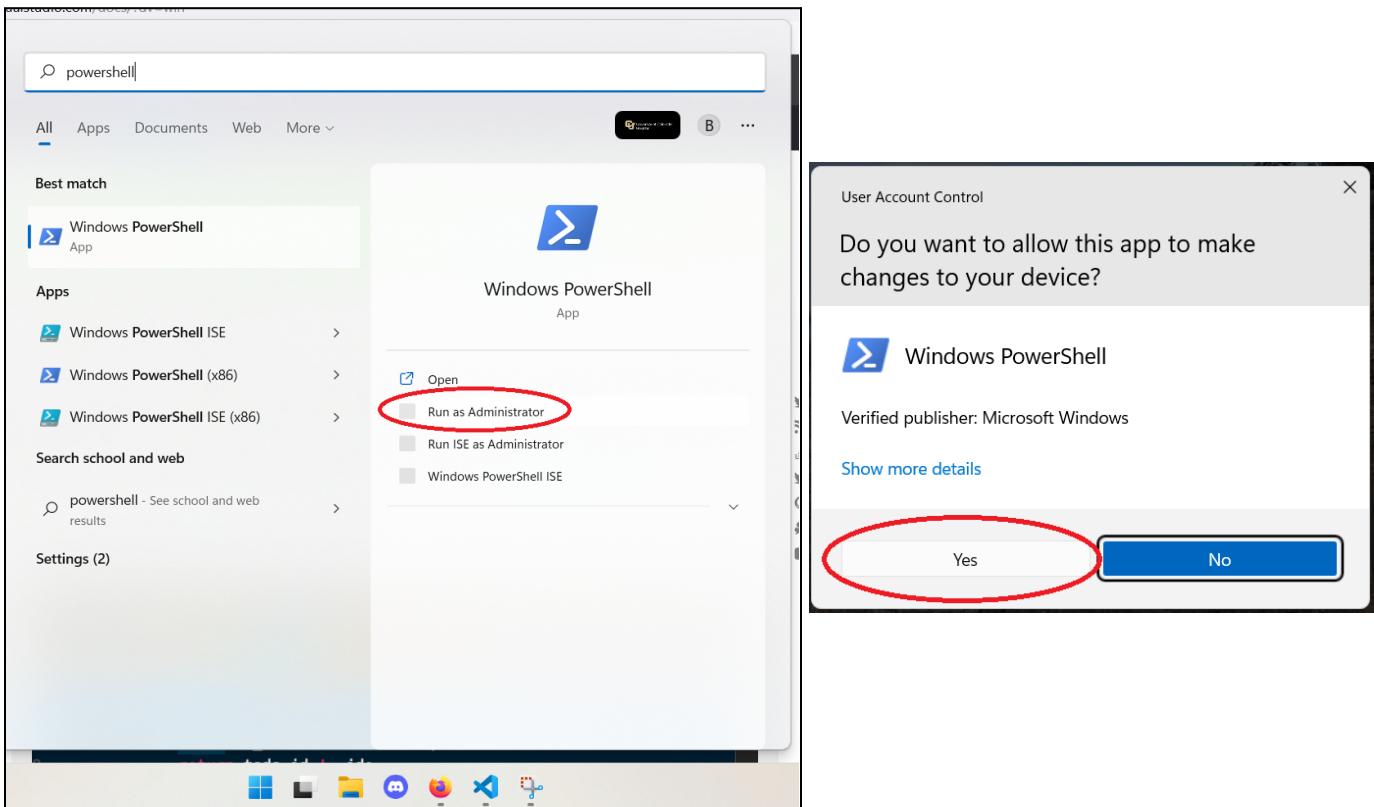
Step 5: You may check out Visual Studio Code by launching it.



Installing Windows Subsystem for Linux (WSL) (Part 2)

Let us now install WSL so that we can emulate linux commands and functionality on our Windows Machine.

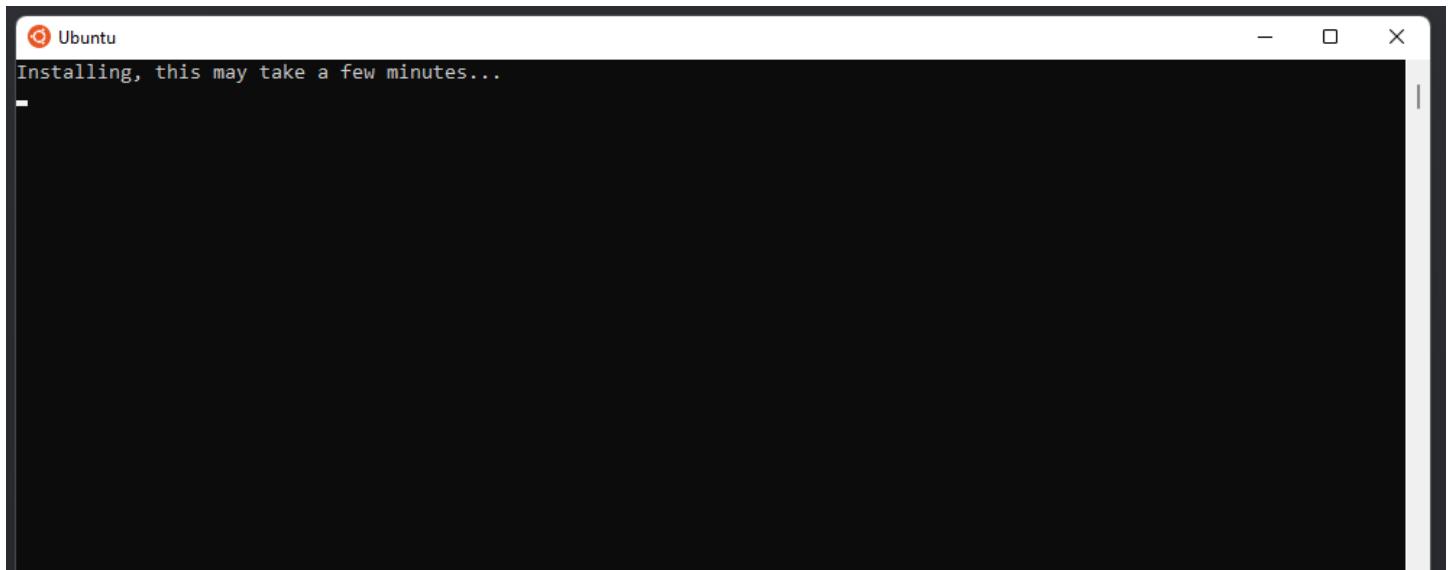
Step 1: Open a PowerShell window as an Administrator. Click “**Yes**” when asked if you want to allow this app to make changes to your device.



Step 2: Type the command “`wsl --install`” and press Enter. When finished, restart your computer. This will start the installation of Windows Subsystem for Linux and the installation of Ubuntu. It may take several minutes to complete.

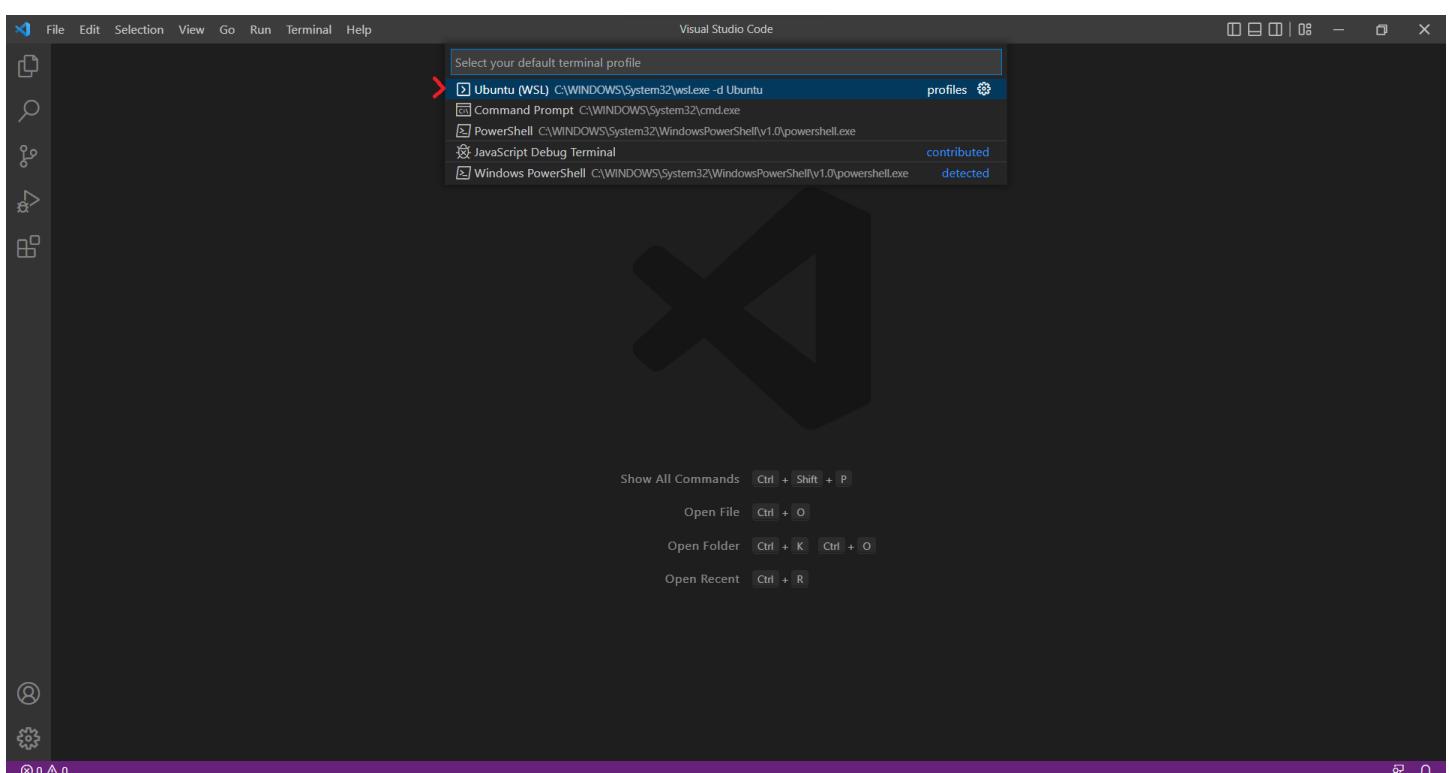
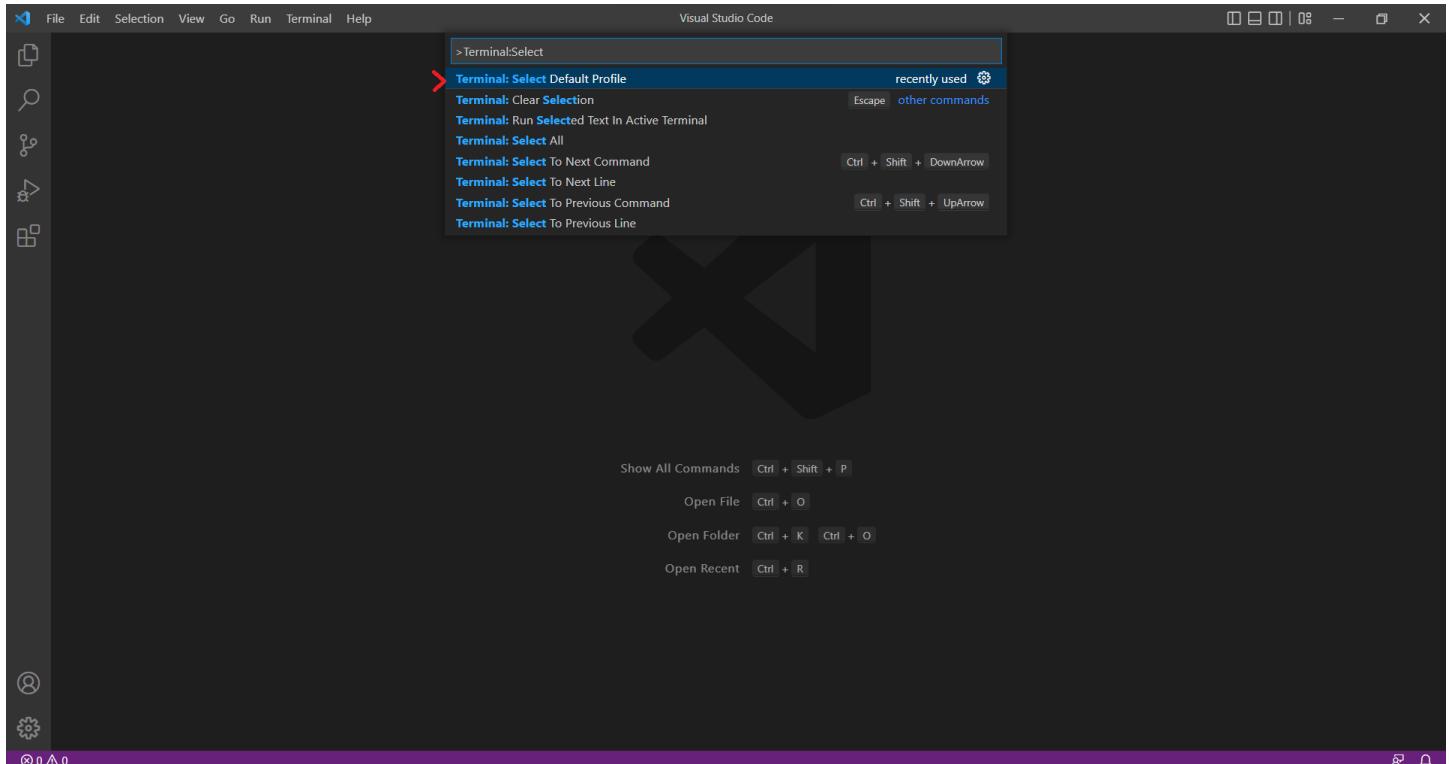
A screenshot of a Windows PowerShell window titled "Administrator: Windows PowerShell". The command `wsl --install` is being run, and the output shows the installation of the Virtual Machine Platform, Windows Subsystem for Linux, WSL Kernel, GUI App Support, and Ubuntu. The message "The requested operation is successful. Changes will not be effective until the system is rebooted." is displayed at the end.

Step 3: When your computer reboots, the Ubuntu terminal will appear and continue installation. Then, you will be prompted for a username and password. Enter an appropriate username and password of your choice. The password will be hidden from you as you type but do not panic, just type your password and hit enter. It will prompt you to re-enter the password, please do so and wait for the installation process to complete. If you have any issues with the password please refer to the debugging file in HW0. Note: **remember your password** as you will need to use it quite often. You can also store your password somewhere secure.

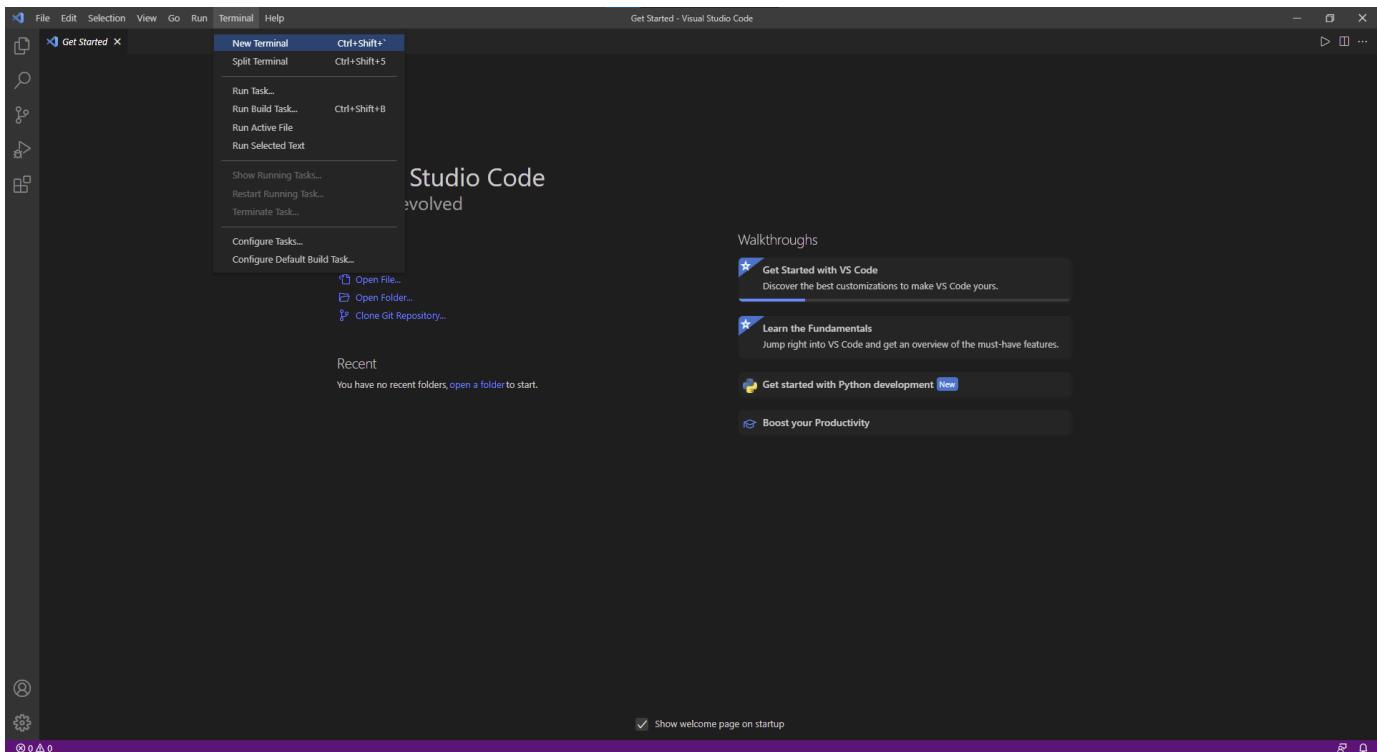


```
brcr5319@BMC-DESKTOP:~  
Installing, this may take a few minutes...  
Please create a default UNIX user account. The username does not need to match your Windows username.  
For more information visit: https://aka.ms/wslusers  
Enter new UNIX username: brcr5319  
New password:  
Retype new password:  
passwd: password updated successfully  
Installation successful!  
To run a command as administrator (user "root"), use "sudo <command>".  
See "man sudo_root" for details.  
  
Welcome to Ubuntu 20.04 LTS (GNU/Linux 5.10.16.3-microsoft-standard-WSL2 x86_64)  
  
 * Documentation: https://help.ubuntu.com  
 * Management: https://landscape.canonical.com  
 * Support: https://ubuntu.com/advantage  
  
System information as of Wed Aug 17 15:59:42 MDT 2022  
  
System load: 0.0 Processes: 8  
Usage of /: 0.4% of 250.98GB Users logged in: 0  
Memory usage: 3% IPv4 address for eth0: 172.19.168.200  
Swap usage: 0%  
  
0 updates can be installed immediately.  
0 of these updates are security updates.  
  
The list of available updates is more than a week old.  
To check for new updates run: sudo apt update  
brcr5319@BMC-DESKTOP:~$  
  
This message is shown once once a day. To disable it please create the  
/home/brcr5319/.hushlogin file.  
brcr5319@BMC-DESKTOP:~$
```

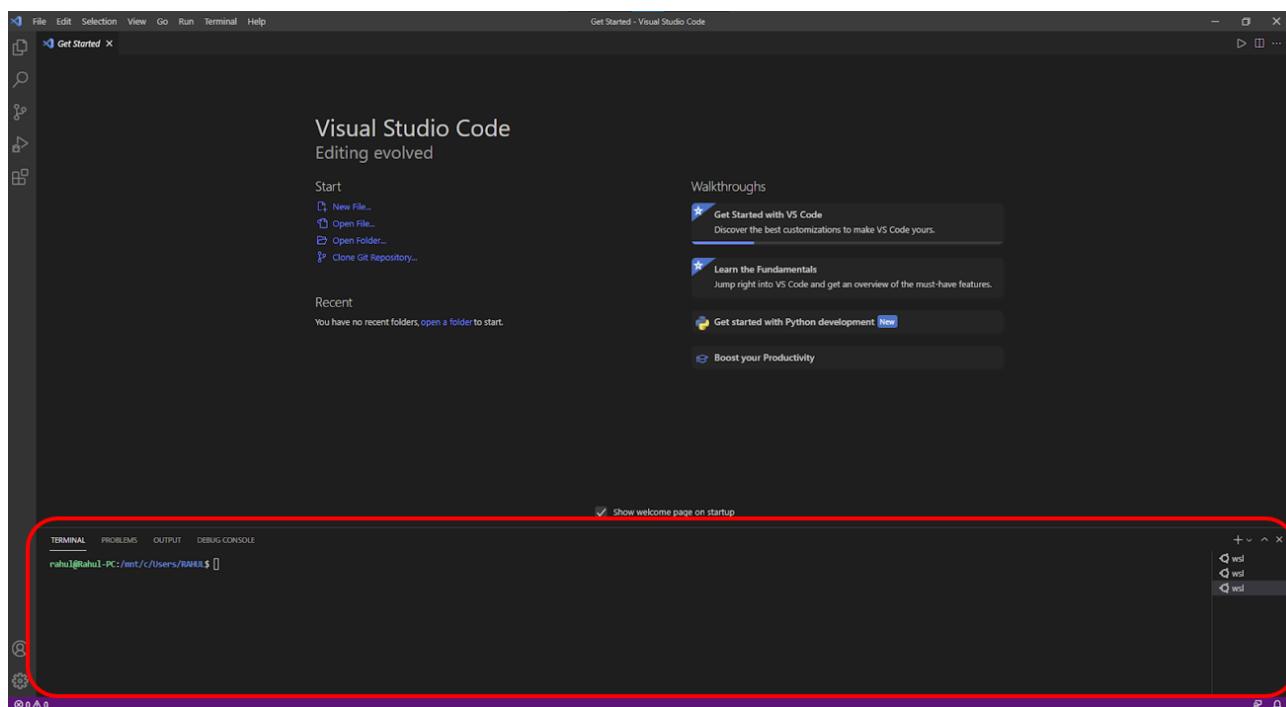
Step 4: Now start Visual Studio Code again to add the final touches. Press on **ctrl+shift+p** to get a drop down and search for Terminal: Select Default Profile. (If you already have an older version of VS Code, you could also look for Terminal: Select Default Shell. Both are the same options.) Choose “Ubuntu (WSL)” as your default terminal.



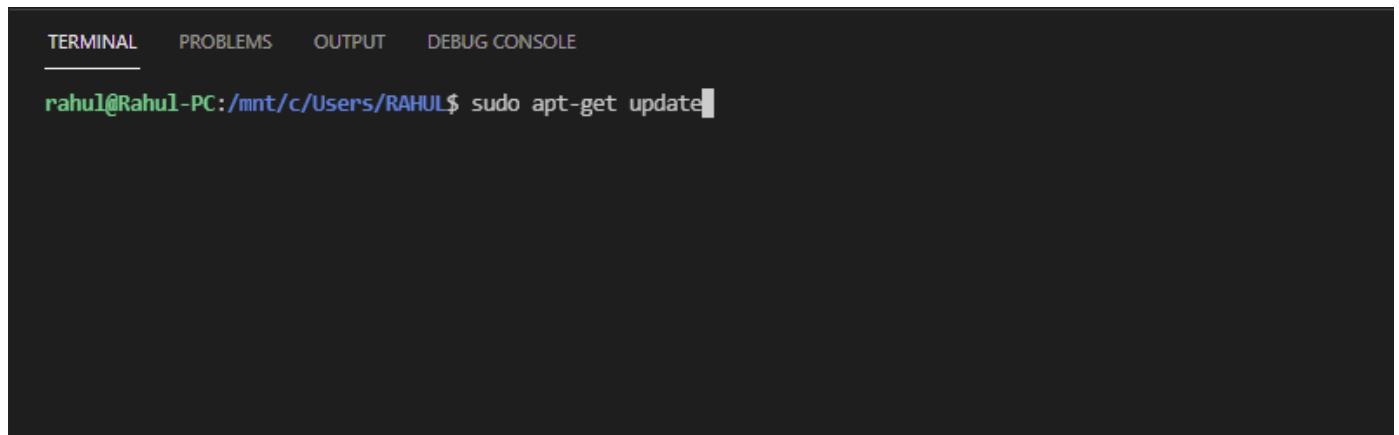
Step 5: We can now click on New Terminal from the drop down menu on the top of Visual Studio Code. This will open a new terminal on the bottom of your screen.



Note about what we're looking at here: The below screenshot shows a terminal where we can execute bash commands. Your terminal may look slightly different, as everyone's computer is different. In the screenshot below, **rahul** is the username of the person using the terminal, and **Rahul-PC** is the name of the computer. The blue text that reads **/mnt/c/Users/RAHUL** is the exact file path where the user is currently located in the terminal. Don't worry about copying this file path, yours will be different, more specifically it will be the default user path with your computer's username. The user types commands after the **\$** symbol.

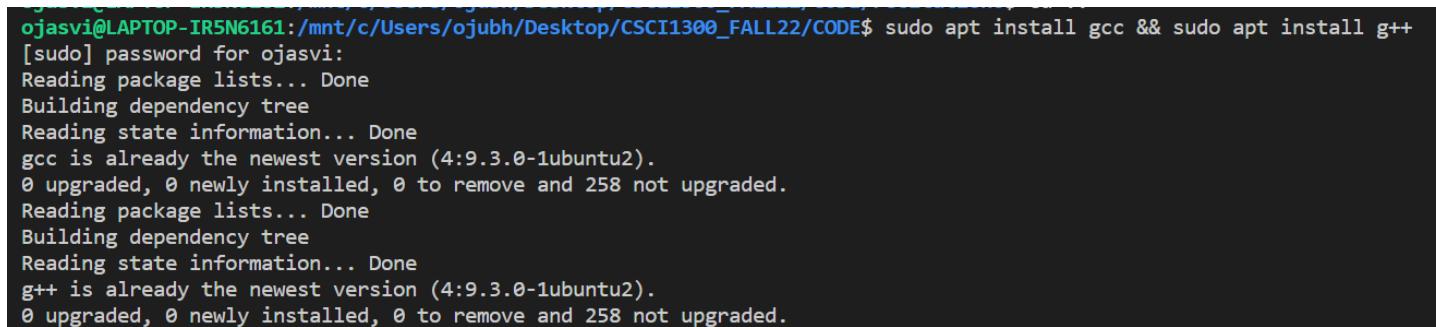


Step 6: We need to run an update so that we can use the C++ compiler. In the terminal window, run the command “`sudo apt-get update`” and hit Enter. It may take several minutes to complete. Then, use the following command to install updates to Ubuntu: “`sudo apt update && sudo apt upgrade`”. When prompted, type “`y`” and hit enter to accept changes.



A screenshot of a terminal window. At the top, there are tabs labeled TERMINAL, PROBLEMS, OUTPUT, and DEBUG CONSOLE. The TERMINAL tab is selected. Below the tabs, the command `rahul@Rahul-PC:/mnt/c/Users/RAHUL$ sudo apt-get update` is visible, with the cursor at the end of the command line. The rest of the terminal window is blank, showing a black background.

Step 7: We also need to install g++. In the terminal window, run the command “`sudo apt install gcc && sudo apt install g++`” and hit Enter. It may take several minutes to complete. Once it finishes, restart VS Code. Type “`y`” and hit enter when prompted. (Twice)
Note: In the screenshot below, the dependencies were already installed, it may look a bit different when installing for the first time.

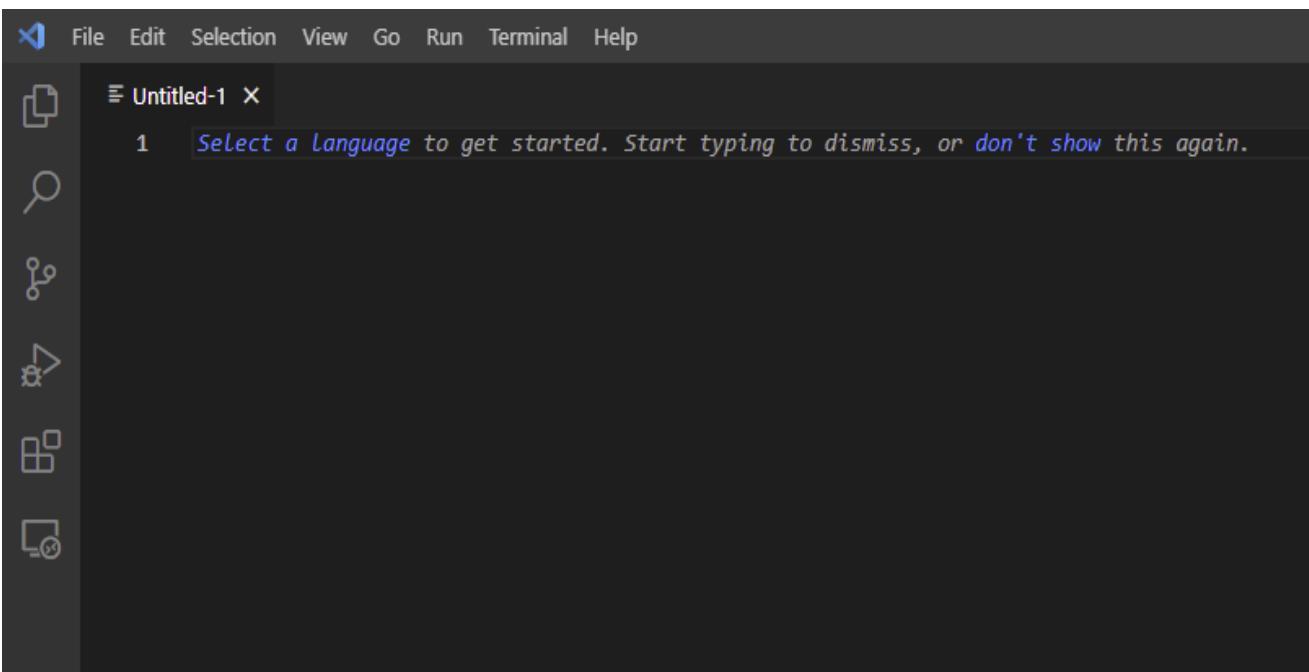
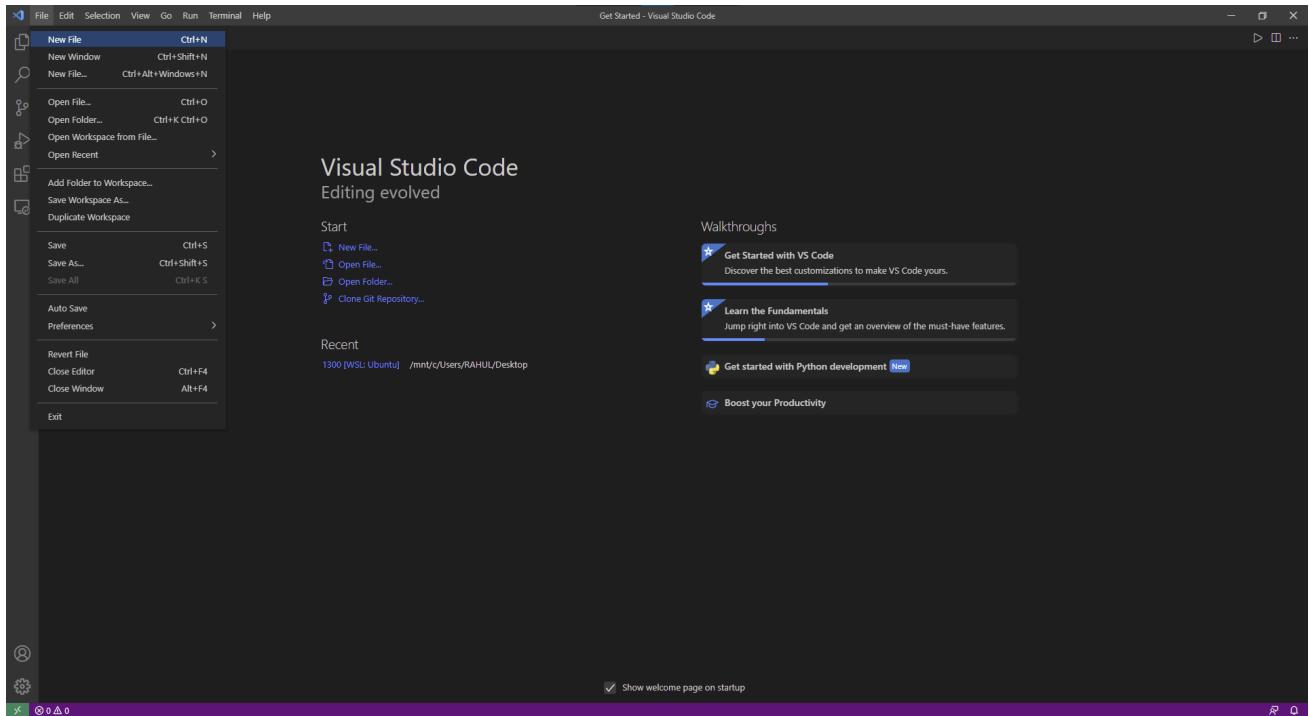


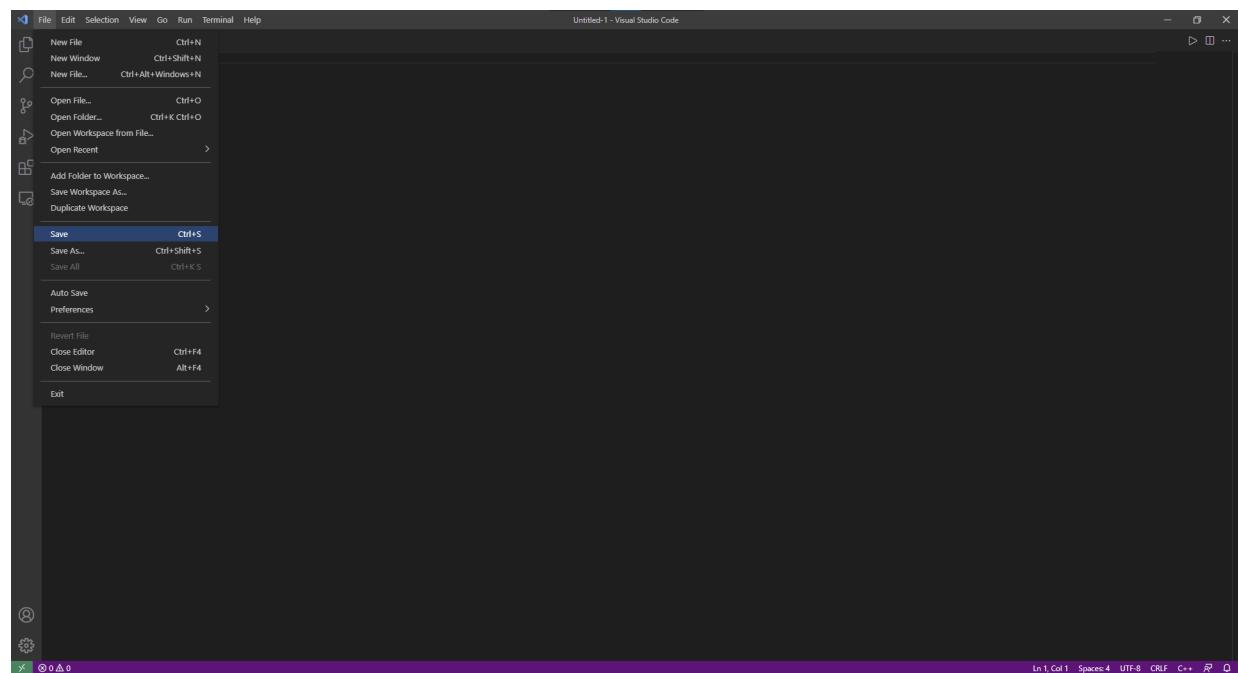
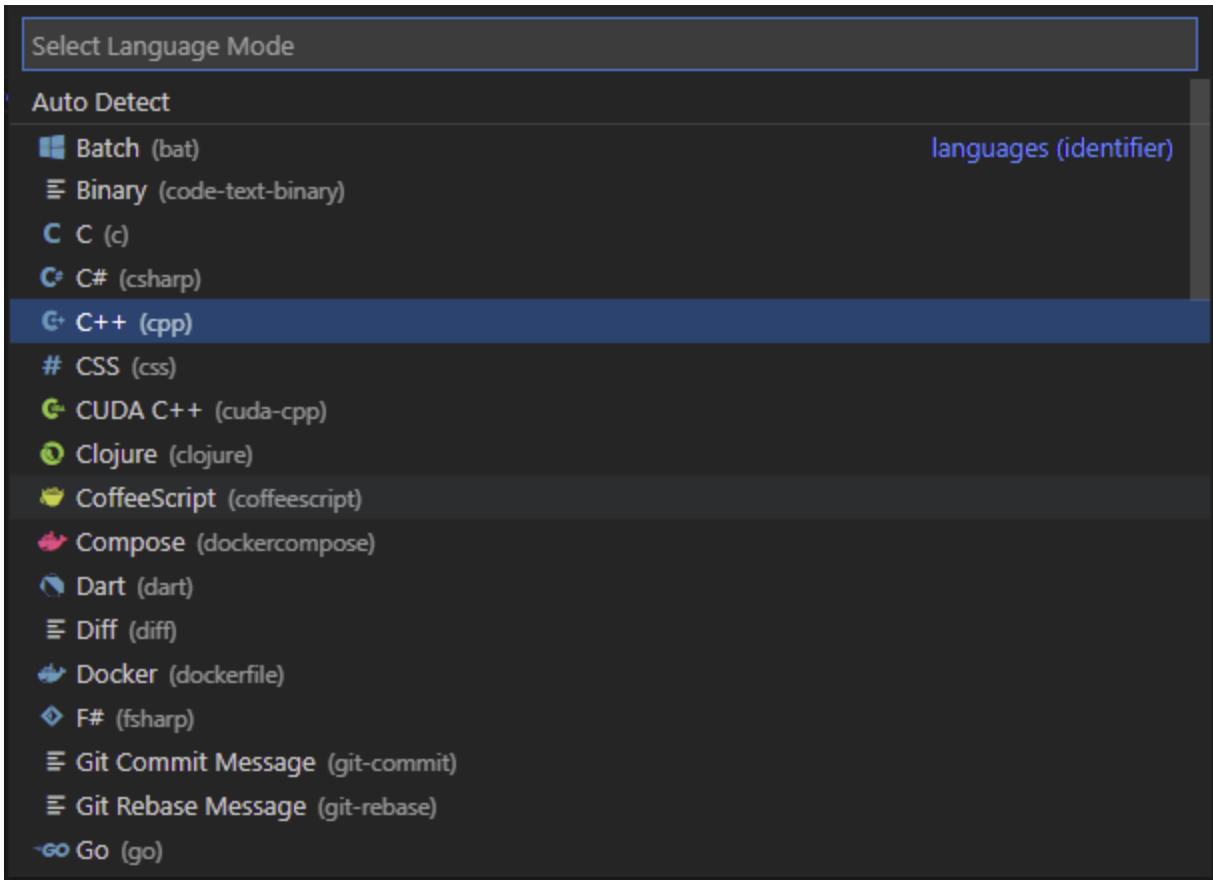
A screenshot of a terminal window. The command `ojasvi@LAPTOP-IR5N6161:/mnt/c/Users/ojubh/Desktop/CSCI1300_FALL22/CODE$ sudo apt install gcc && sudo apt install g++` is visible, with the cursor at the end of the command line. The terminal then displays the output of the package manager, which shows that both gcc and g++ are already the newest version and no upgrades are needed. The output ends with a message indicating that 0 upgraded, 0 newly installed, and 258 not upgraded.

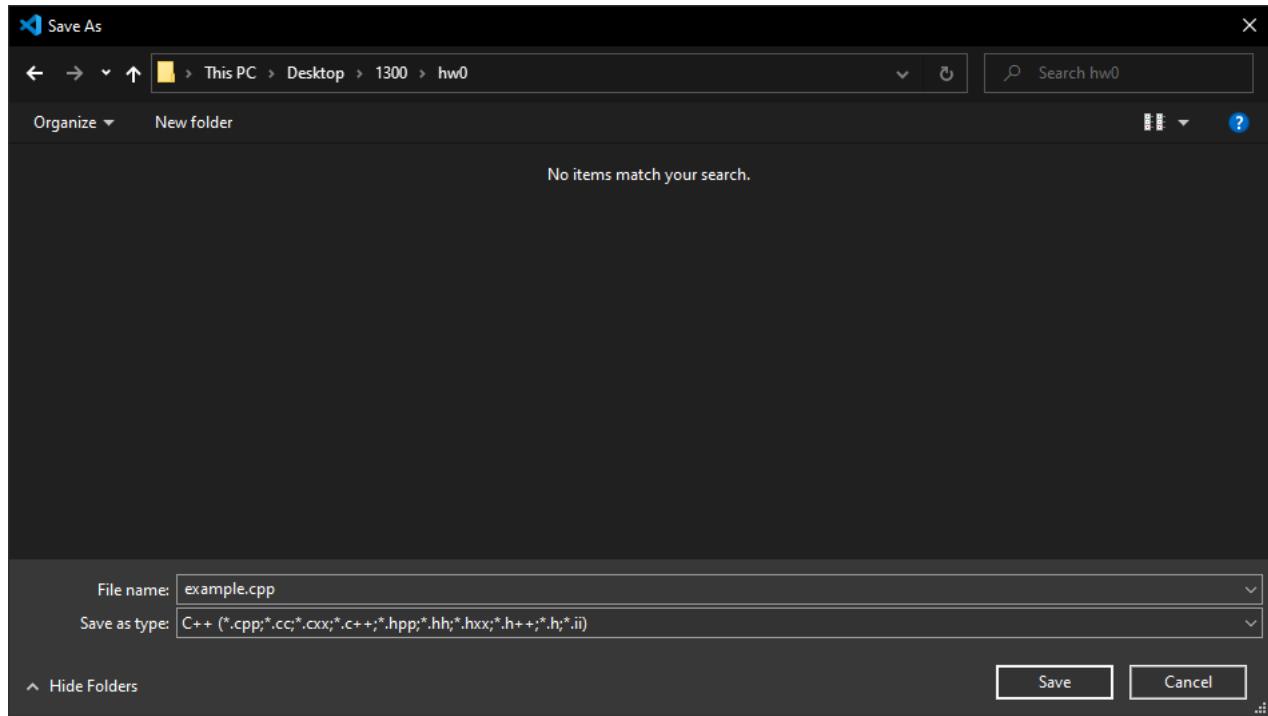
Well done! Go to the section Compiling Code In Terminal (next page) to compile code on your computer.

Compiling Code In Terminal

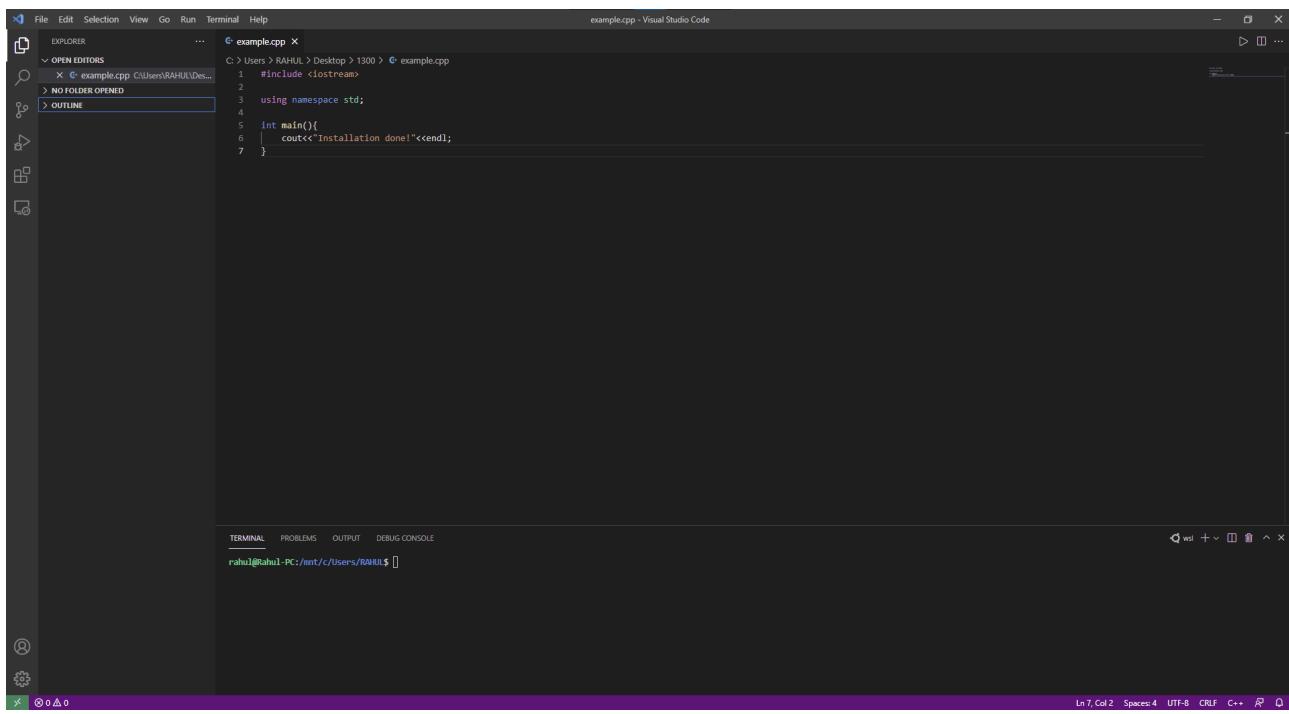
When you open VSCode you should be able to see some options like creating a new file, opening an existing file etc. You will also find these options in the File menu from the top menu bar. Create a new file and save that file as filename.cpp. The filename can be anything as you like. Open that file, select the language c++(cpp) and type the code to be compiled.







Now open a terminal by opening the Terminal menu from the top menu bar and selecting “New Terminal” as shown in previous steps.



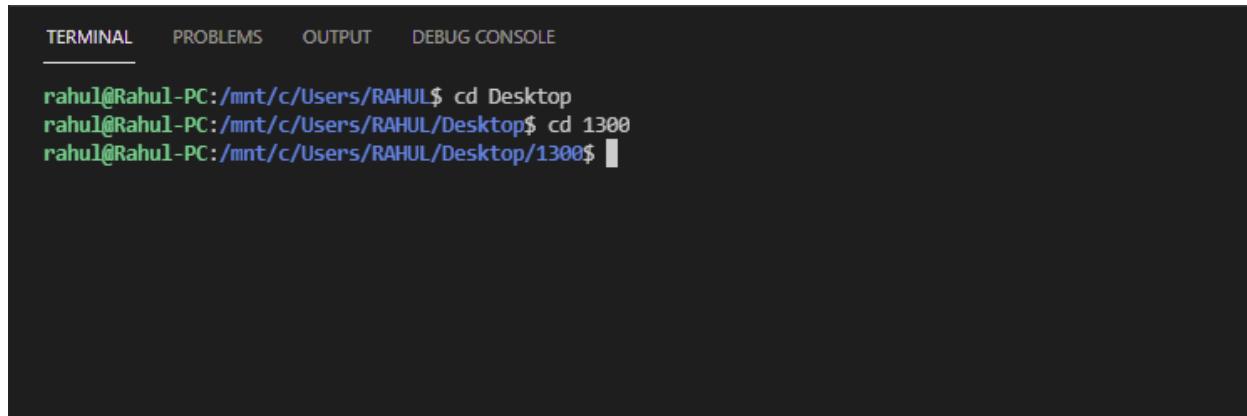
In order to compile and run your code you first need to navigate to where your code is saved through the terminal. You do this using the `cd` (change directory) command (see the Homework 0 document on Canvas if you need an explanation on how to do this). If you have a folder open in VSCode by default the terminal will open to the location of that folder.

You can visualize your location using `pwd` (print working directory).



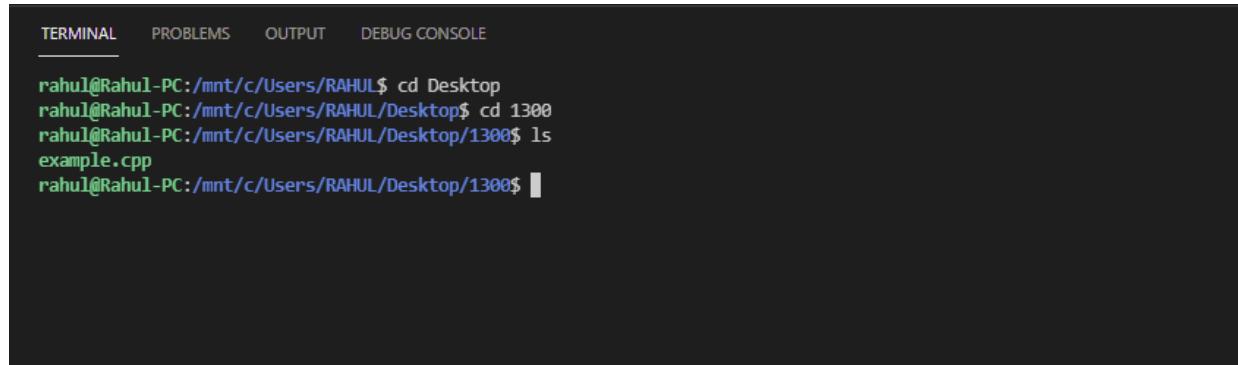
A screenshot of a terminal window in Visual Studio Code. The title bar shows 'TERMINAL' and other tabs like 'PROBLEMS', 'OUTPUT', and 'DEBUG CONSOLE'. The status bar at the top right says 'ws' and has icons for file operations. The terminal itself is dark-themed and displays the following text:
rahul@Rahul-PC:/mnt/c/Users/RAHUL\$ pwd
/mnt/c/Users/RAHUL
rahul@Rahul-PC:/mnt/c/Users/RAHUL\$

But let's say the code I want to run is located in a subdirectory called **1300**, I can navigate to **1300** using **cd**.



A screenshot of a terminal window in Visual Studio Code. The title bar shows 'TERMINAL' and other tabs like 'PROBLEMS', 'OUTPUT', and 'DEBUG CONSOLE'. The terminal displays the following text:
rahul@Rahul-PC:/mnt/c/Users/RAHUL\$ cd Desktop
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop\$ cd 1300
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300\$

Now let's double check the files we are looking for are actually there by using the **ls** (list) command.



A screenshot of a terminal window in Visual Studio Code. The title bar shows 'TERMINAL' and other tabs like 'PROBLEMS', 'OUTPUT', and 'DEBUG CONSOLE'. The terminal displays the following text:
rahul@Rahul-PC:/mnt/c/Users/RAHUL\$ cd Desktop
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop\$ cd 1300
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300\$ ls
example.cpp
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300\$

Now to compile our code we use the program **g++** in the following command.

"**g++ -std=c++17 example.cpp**"

g++ is the compiler program

-std=c++17 specifies the version of C++ we want to use

example .cpp is the file we want to compile

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300$ g++ --std=c++17 example.cpp
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300$
```

This command creates a file named `a.out` which is the compiled version of the code in `example.cpp`, which can be executed. Now to execute our code we run `a.out` by typing the following and hitting enter.

`“./a.out”`

TERMINAL PROBLEMS OUTPUT DEBUG CONSOLE

```
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300$ g++ --std=c++17 example.cpp
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300$ ls
a.out example.cpp
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300$ ./a.out
Installation done!
rahul@Rahul-PC:/mnt/c/Users/RAHUL/Desktop/1300$
```

Congrats on compiling and running a C++ program from a terminal!