

Visual Studio Code Installation Guide (Windows)
CS205: Computing Foundations for Computational Science
Spring 2020
Author: Wen Rui Liao

An Introduction to Visual Studio Code

Visual Studio Code (VSC) is a FREE lightweight development tool that combines the simplicity of a source code/text editor with powerful tools like break-point debugging. VSC combines the best parts of classical text editors like **Sublime Text** (syntax highlighting/easy setup) and powerful Integrated Development Environments like **PyCharm** (step-by-step debugging and inbuilt terminal). For the purposes of CS205, VSC makes it a breeze for **Windows** ☐ **Users** to setup Python3 and run the class examples/assignments within 1 program (instead of flipping between terminal windows and text editors).

Evolve from Jupyter Notebooks & Anaconda, use VSC today!






Setting up VSC

Let's get started!

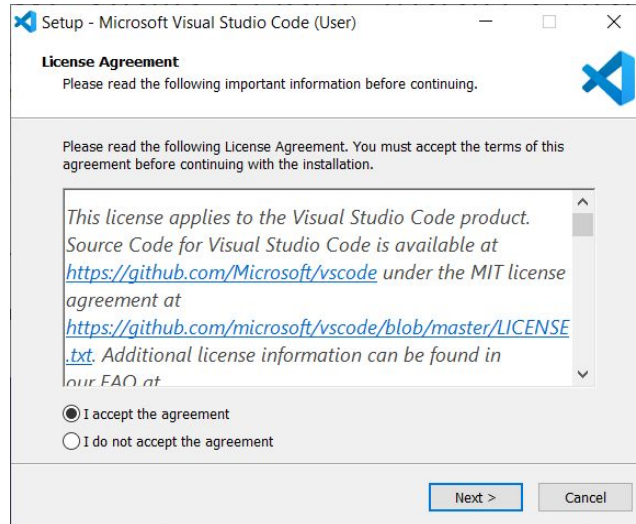
1. First, download the relevant installer from [here](#). For the purposes of this guide, I will be walking through the Windows Installation (User Installer, 64-bit version)

Download Visual Studio Code

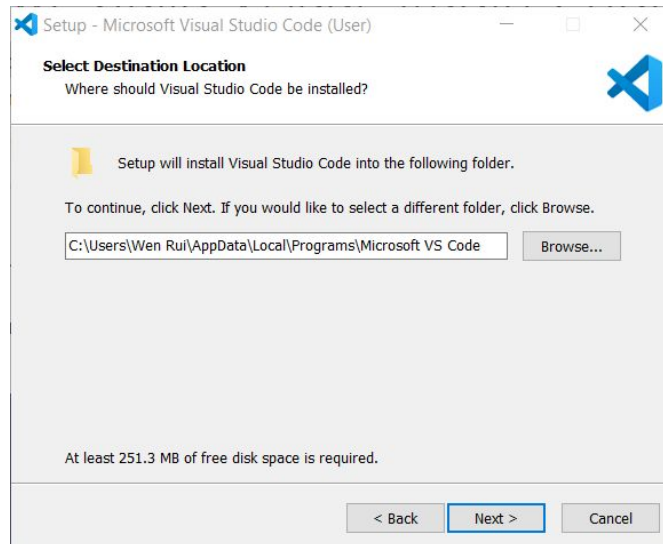
Free and built on open source. Integrated Git, debugging and extensions.

 <div style="background-color: #0070C0; color: white; padding: 5px; margin: 5px 0;">↓ Windows <small>Windows 7, 8, 10</small></div> <div style="display: flex; justify-content: space-around; font-size: small;"><div>User Installer 64 bit 32 bit</div><div>System Installer 64 bit 32 bit</div><div>.zip 64 bit 32 bit</div></div>	 <div style="display: flex; justify-content: space-around;"><div style="background-color: #0070C0; color: white; padding: 5px; margin: 5px 0;">↓ .deb <small>Debian, Ubuntu</small></div><div style="background-color: #0070C0; color: white; padding: 5px; margin: 5px 0;">↓ .rpm <small>Red Hat, Fedora, SUSE</small></div></div> <div style="display: flex; justify-content: space-around; font-size: small;"><div>.deb 64 bit</div><div>.rpm 64 bit</div><div>.tar.gz 64 bit</div><div>Snap Store</div></div>	 <div style="background-color: #0070C0; color: white; padding: 5px; margin: 5px 0;">↓ Mac <small>macOS 10.10+</small></div>
--	---	--

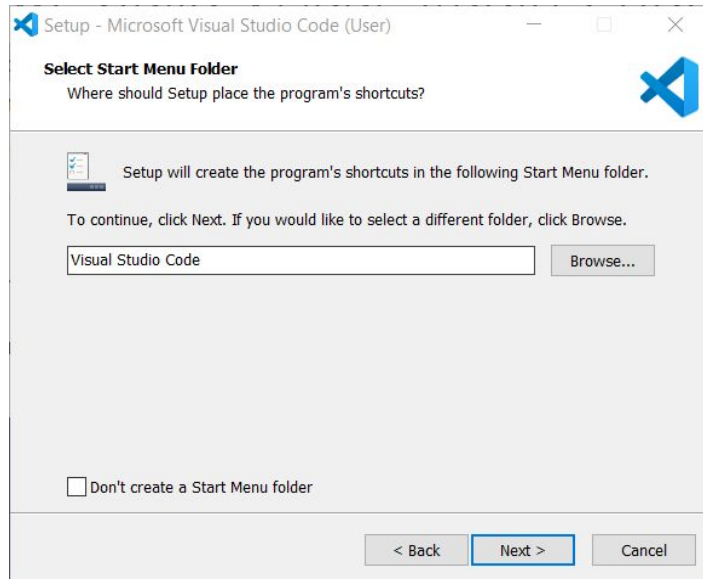
2. Accept the *License Agreement*



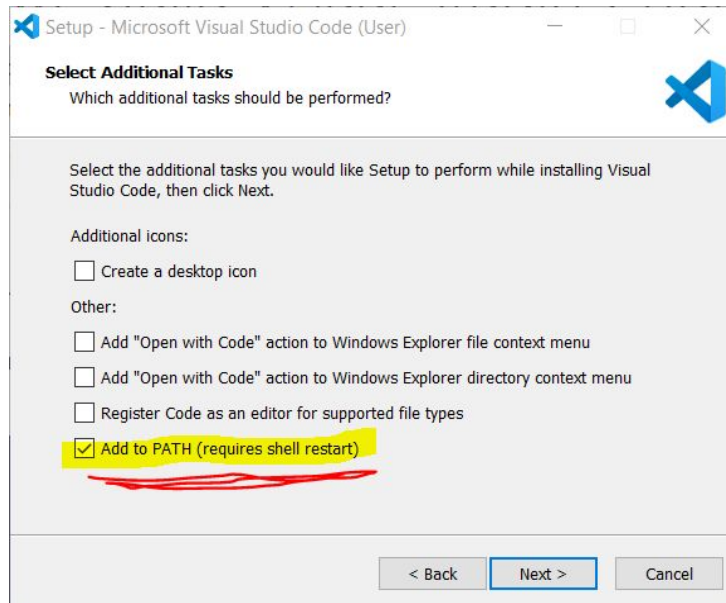
3. Set a Destination Location (default is fine)



4. Create a Start Menu Folder (default is fine)

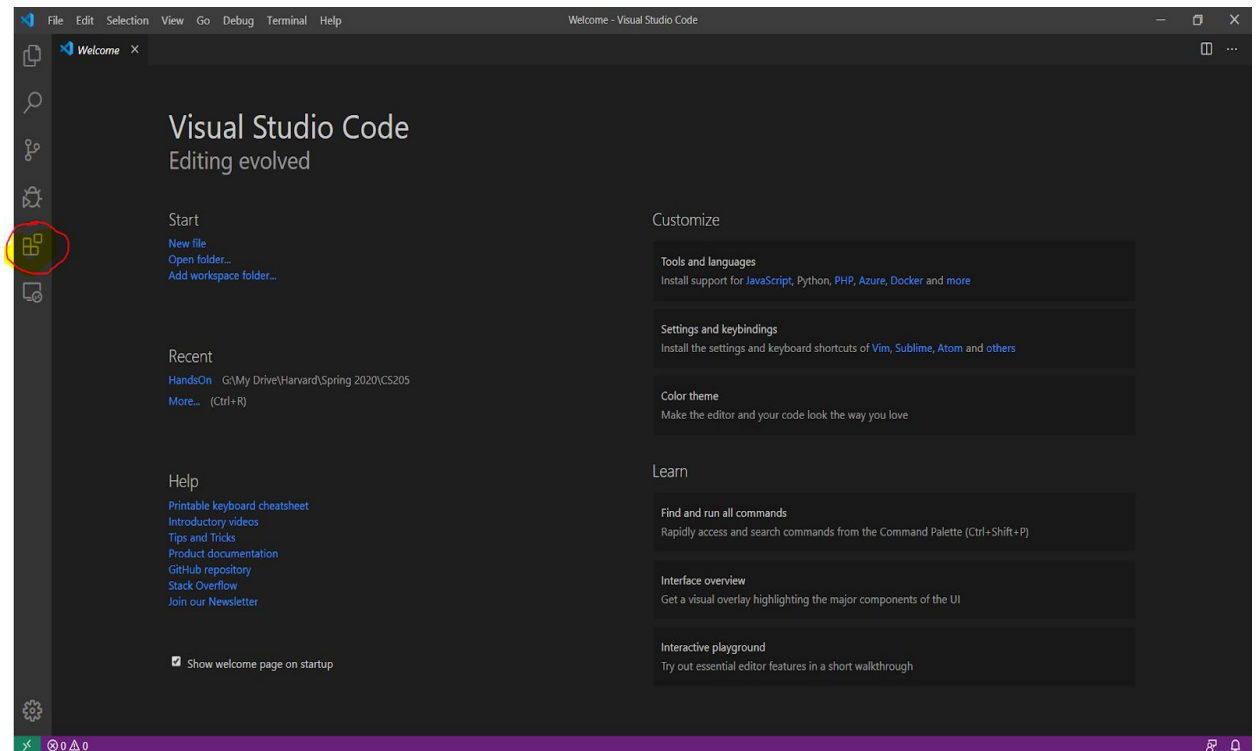


5. In the next screen, make sure that "Add to PATH" is selected

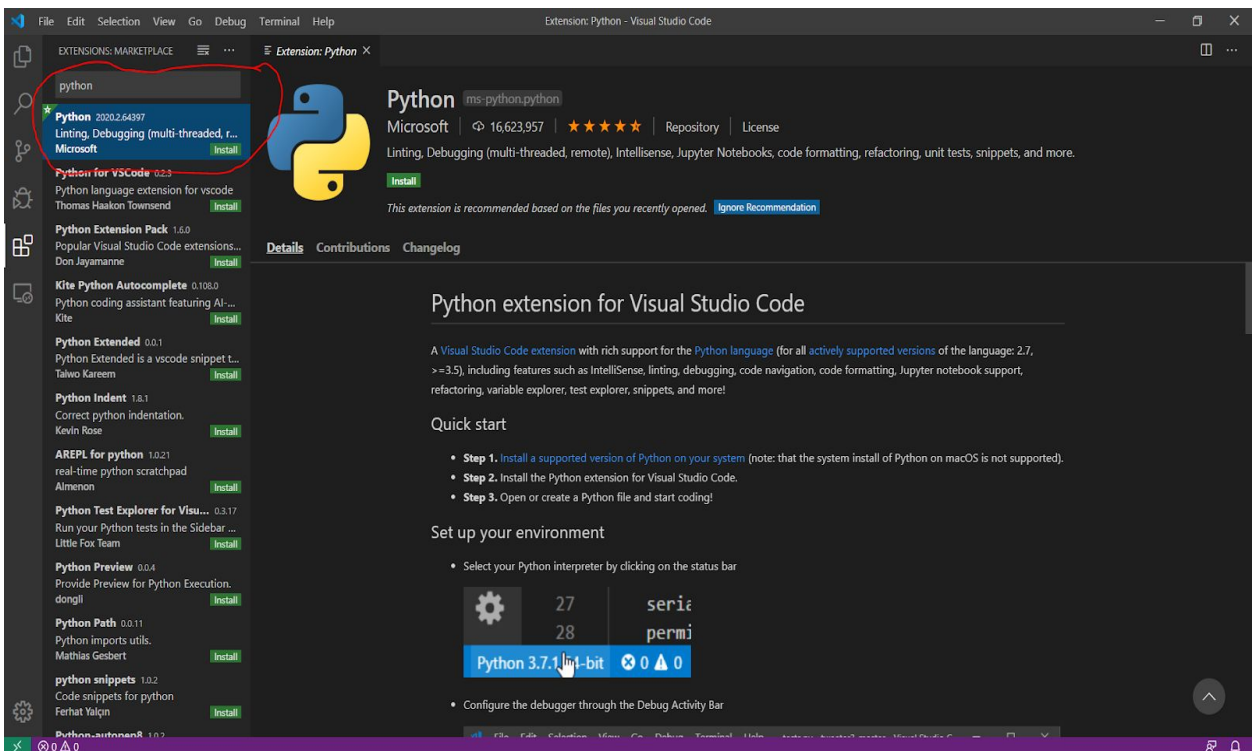


6. Press *Install*

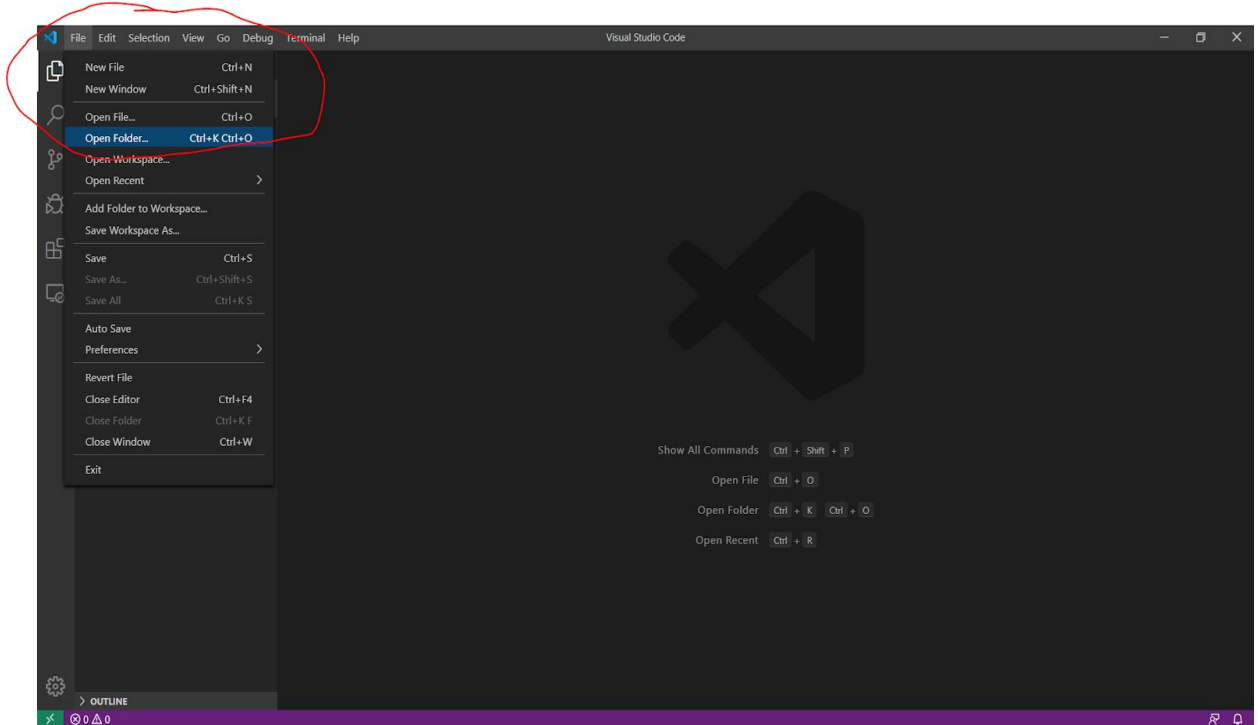
7. Launch *Visual Studio Code* and go to **Extensions** (icon highlighted below, shortcut is Ctrl + Shift + X)



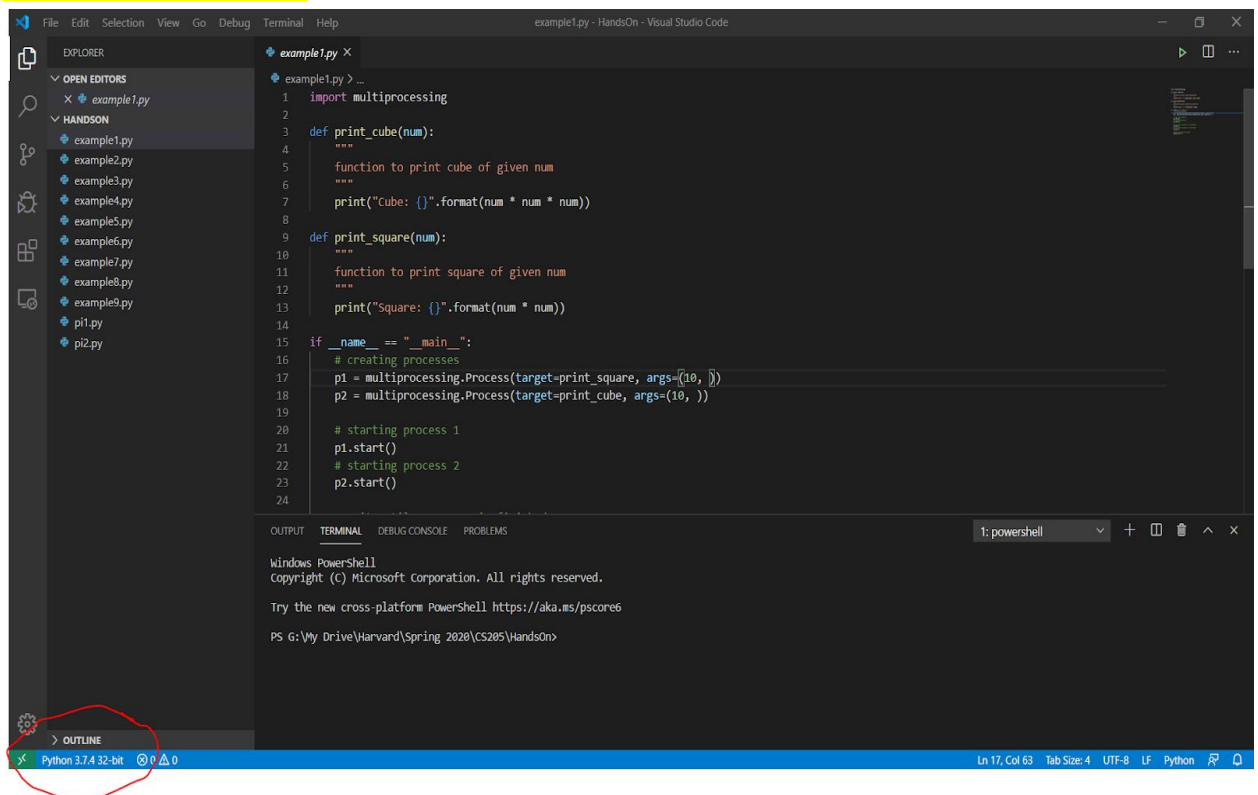
8. Search for **Python** in the search box and install the first result



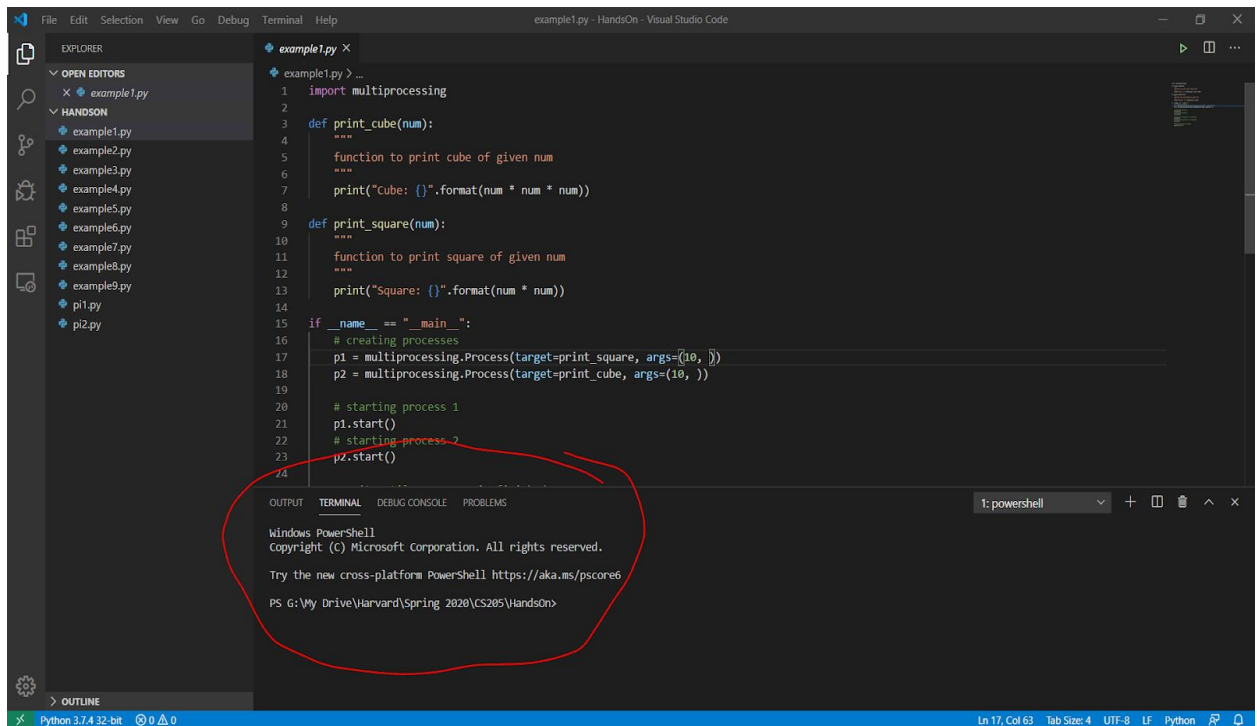
9. And that's it, hopefully Python is installed! To test it out, let us store our [class examples from 02/20/20](#) in a folder and open this folder at **File > Open Folder**.



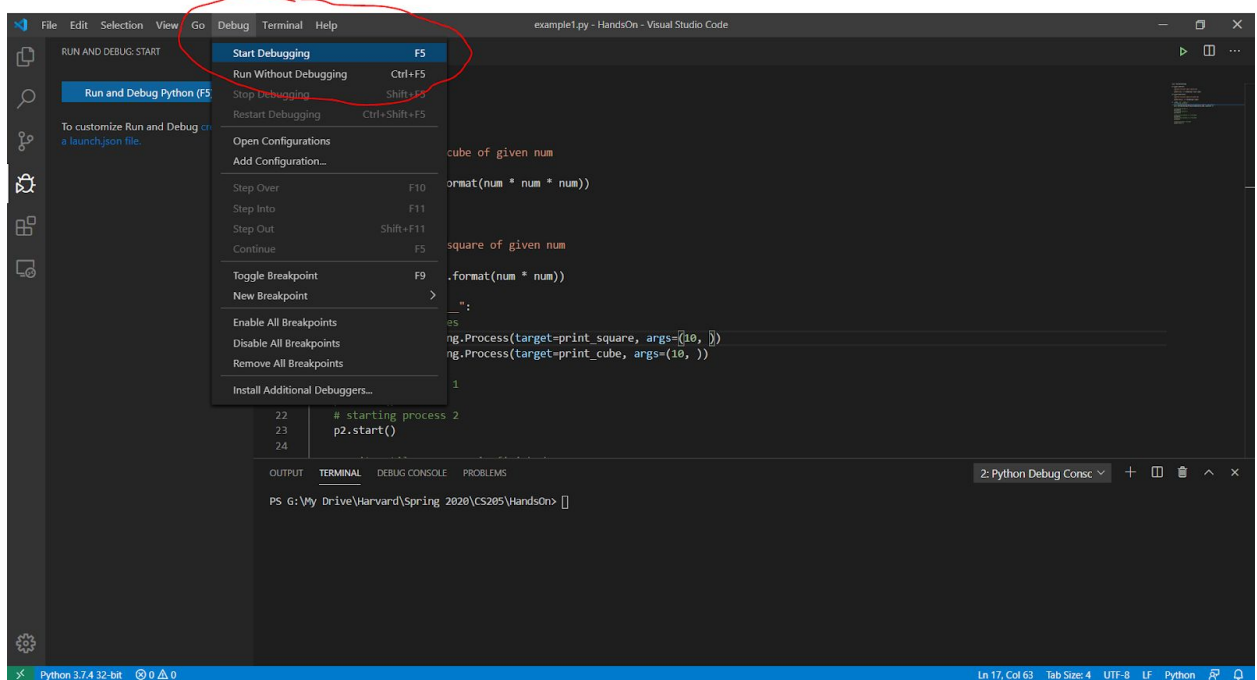
10. Open `example1.py` and you should see it pop up in the editor. Make sure that the **Python 3.7.4 interpreter** is selected in the bottom left corner of your screen.

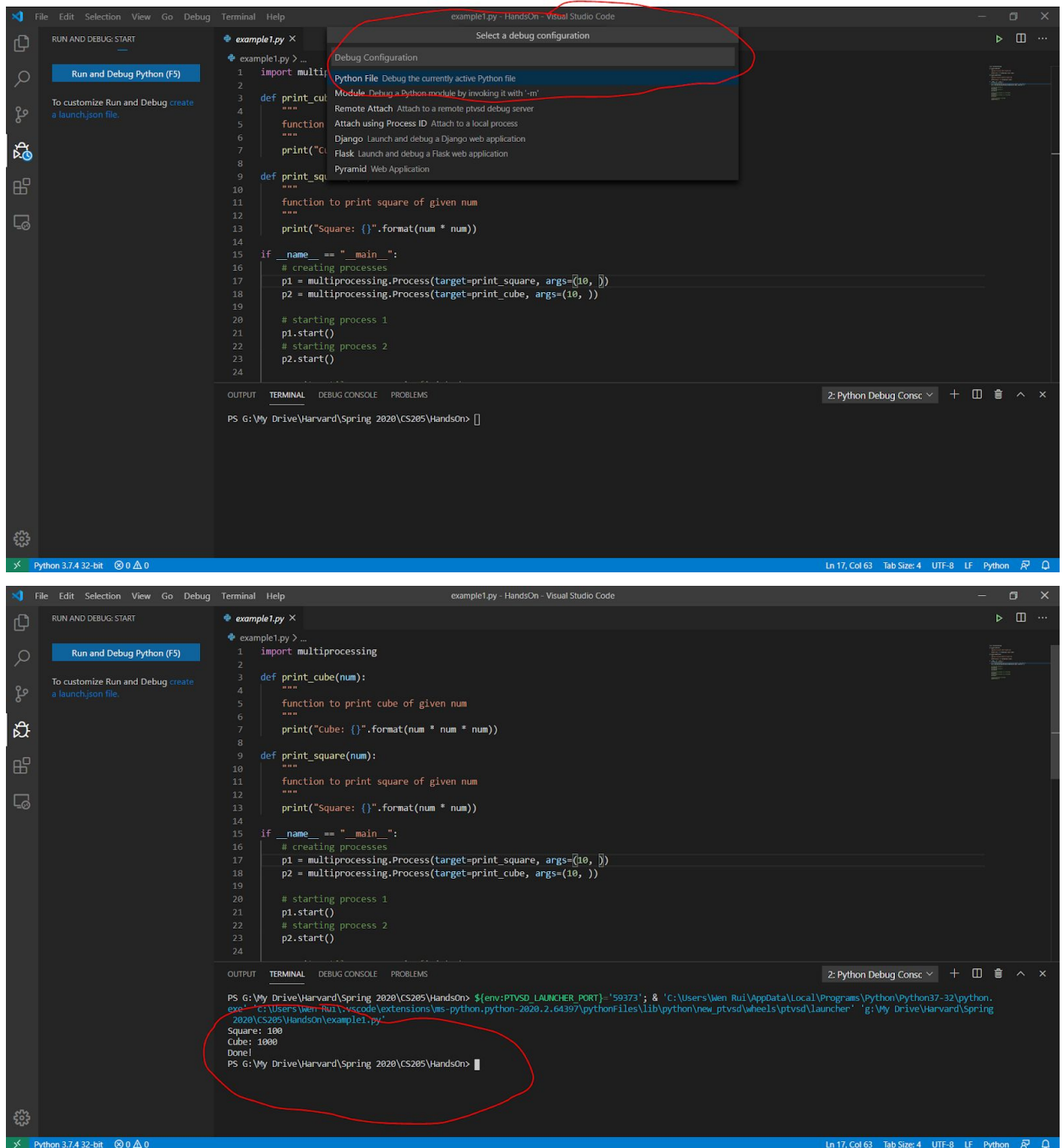


11. There's a lot happening on this screen which makes VSC powerful. On the bottom of the screen there is a terminal window that is pointing to the directory where the folder is located. You can run your familiar terminal commands here. Other windows here include an *Output* window and a *Debug Console*.



12. Moreover, we can run our code from within VSC (which makes it great). To check if everything is working, run the code with **Debug > Start Debugging (Shortcut F5)**. If everything works, you should see the output in the terminal below!





13. And that's it! You now have a great tool for running Python files in this class. If you are feeling adventurous, you can use "break-points" as well; which stops the execution of the code at that particular point during debugging and allows for step-by-step debugging. To

set a breakpoint, simply click on the empty space next to a line-number.

The screenshot shows the Visual Studio Code interface with a Python file named `example1.py` open. The file contains the following code:

```

1  import multiprocessing
2
3  def print_cube(num):
4      """
5      function to print cube of given num
6      """
7      print("Cube: {}".format(num * num * num))
8
9  def print_square(num):
10     """
11     function to print square of given num
12     """
13     print("Square: {}".format(num * num))
14
15  if __name__ == "__main__":
16     # creating processes
17     p1 = multiprocessing.Process(target=print_square, args=(10, ))
18     p2 = multiprocessing.Process(target=print_cube, args=(10, ))
19
20     # starting process 1
21     p1.start()
22     # starting process 2
23     p2.start()
24

```

A breakpoint is set at line 9, which is circled in red. The left sidebar shows the 'RUN AND DEBUG' panel with the 'Run and Debug Python (F5)' button highlighted. Below it, there is a link to 'To customize Run and Debug create a launch.json file.' The bottom panel shows the 'TERMINAL' with the following output:

```

PS G:\My Drive\Harvard\Spring 2020\CS205\HandsOn> $env:PTVSD_LAUNCHER_PORT="59373"; & 'C:\Users\Wen Rui\AppData\Local\Programs\Python\Python37-32\python.exe' -c "import sys; sys.path.append('G:\My Drive\Harvard\Spring 2020\CS205\HandsOn\extensions\ms-python.python-2020.2.64387\pythonFiles\lib\python\new_ptvsd\wheels\ptvsd\launcher'); g:\My Drive\Harvard\Spring 2020\CS205\HandsOn\example1.py"
Square: 100
Cube: 1000
Done!
PS G:\My Drive\Harvard\Spring 2020\CS205\HandsOn>

```

The bottom status bar indicates the Python version is 3.7.4 32-bit, and the file is at line 17, column 63.

Conclusion

VSC is great as it abstracts away a lot of the installation pains for Windows Users. Moreover, it allows you to live within this program for the purposes of this class.

Thank you Daddy Nadella for blessing us with this great tool!

If you have any installation questions, feel free to ping me at wliau@g.harvard.edu.

