

---

## Start Coding on Python Programming Language

---

### Installing Python and IDE (VS Code)

This section will guide students through setting up Python and VS Code on both Windows and Mac.

#### ► For Windows Users:

##### ✓ Step 1: Download and Install Python

1. Go to the official Python website: <https://www.python.org/downloads/>
2. Click the Download Python button for Windows.
3. Once the installer is downloaded, open it and check the box that says “Add Python to PATH” at the bottom of the setup window.
4. Click Install Now.
5. After installation, open Command Prompt (press **Win + R**, type **cmd**, and press Enter).
6. Type **python --version** to check if Python is installed successfully. You should see the Python version displayed (e.g., Python 3.x.x).

##### ✓ Step 2: Install Visual Studio Code

1. Go to the VS Code website: <https://code.visualstudio.com/download/>
2. Download the installer for Windows and run it.
3. During installation, check the box that says “Add to PATH” and click Next until installation is complete.
4. Once installed, open VS Code.

##### ✓ Step 3: Set Up Python in VS Code

1. Open VS Code and go to the Extensions tab (or press **Ctrl+Shift+X**).
2. In the search bar, type “Python” and install the official Python extension by Microsoft.
3. Once installed, open the Command Palette (press **Ctrl+Shift+P**).
4. Type “Python: Select Interpreter” and press Enter.
5. Select the Python interpreter (the one you just installed, e.g., Python 3.x.x).

##### ✓ Step 4: Run a Python Script

1. Create a new file by going to File → New File. Name it **hello.py**.
2. Write the following Python code: `print("Hello, Python!")`

3. To run the file, press **Ctrl+** or open the terminal by going to Terminal → New Terminal.
4. In the terminal, type **python hello.py** and press **Enter**. You should see the output: Hello, Python!

► **For Mac Users:**

✓ **Step 1: Install Python Using Homebrew**

1. Open the Terminal (press **Cmd + Space**, type Terminal, and press Enter).
2. If you don't have Homebrew installed, run the following command to install it (you'll need an internet connection):

```
/bin/bash -c "$(curl -fsSL https://raw.githubusercontent.com/Homebrew/install/HEAD/install.sh)"
```

3. Follow the on-screen instructions and ensure Homebrew is installed correctly by typing **brew --version**.
4. To install Python, run the command: **brew install python**
5. Once installation is complete, check if Python is installed by typing:

```
python3 --version
```

You should see the installed Python version (e.g., Python 3.x.x).

✓ **Step 2: Install Visual Studio Code**

1. Go to the VS Code website: <https://code.visualstudio.com/download>
2. Download the installer for Mac (usually a .dmg file).
3. Open the .dmg file and drag the VS Code icon to the Applications folder.
4. Once installed, open VS Code from your Applications folder or by using Spotlight (**Cmd + Space**, then type "VS Code").

✓ **Step 3: Set Up Python in VS Code**

1. Open VS Code and go to the Extensions tab (press **Cmd + Shift + X**).
2. In the search bar, type "Python" and install the official Python extension by Microsoft.
3. Once installed, open the Command Palette (press **Cmd + Shift + P**).
4. Type "Python: Select Interpreter" and press Enter.
5. Select the Python interpreter (look for Python 3.x.x).

✓ **Step 4: Run a Python Script**

1. Create a new file by going to File → New File. Name it hello.py.
2. Write the following Python code: **print("Hello, Python!")**

3. To run the file, open the terminal by going to Terminal → New Terminal.
4. In the terminal, type: `python3 hello.py`

You should see the output: Hello, Python!

❑ **Troubleshooting Tips:**

- If Python isn't recognized in the terminal, ensure Python is added to the system PATH (for Windows) or that you are using python3 on Mac.
- For VS Code, make sure the Python extension is installed, and the interpreter is correctly selected.

## Important Instruction for Week 01 Lab:

These exercises are designed to help you understand how to translate C programming concepts into Python. **Do not use AI tools or external code generators** to complete these exercises. They are intended for you to practice your own coding abilities. Think through the logic, reference Python syntax, and work through the challenges independently.

---

## Your task is to Write a Python Programming Language to complete the exercises below:

1. Write a Python program that takes two integers as input and performs basic arithmetic operations (addition, subtraction, multiplication, division). Output the results to the console.
  - a. In C, you used `scanf` and `printf`. In Python, use **`input()`** and **`print()`**.
  - b. Example input/output:

```
Enter first number: 10
Enter second number: 5
Sum: 15
Difference: 5
Product: 50
Quotient: 2.0
```
2. Write a Python program that takes three numbers as input and prints the largest number.
  - a. Use Python's **`if-elif-else statements`** to replace C's if-else or conditional operator.
  - b. Example input/output:

```
Enter first number: 10
Enter second number: 20
Enter third number: 15
The largest number is 20
```
3. Write a Python program that prints numbers from 1 to 100. But for multiples of 3, print "Fizz" instead of the number, and for multiples of 5, print "Buzz". For numbers which are multiples of both 3 and 5, print "FizzBuzz".
  - a. This exercise focuses on using **`for loops`** and **`if`** conditions.
  - b. Example output:

```
1
2
Fizz
```

4  
Buzz  
Fizz  
7  
8  
Fizz  
Buzz  
11  
Fizz  
13  
14  
FizzBuzz

4. Write a Python function that takes an integer as input and returns its factorial. Use this function in your program and print the factorial of a number entered by the user.
  - a. Use a **for** or **while** loop to calculate the factorial.
  - b. Example input/output:

Enter a number: 5  
Factorial of 5 is 120
  
5. Write a Python program that takes a string as input and checks whether it is a palindrome (reads the same forward and backward).
  - a. Use Python's string slicing or loops to achieve this.
  - b. Example input/output:

Enter a string: madam  
madam is a palindrome

### (Bonus) Password Strength Checker

Write a Python program that checks the strength of a password entered by the user. The program should classify the password as **Weak**, **Moderate**, or **Strong** based on the following criteria:

1. A password is **Weak** if it is less than 8 characters long.
2. A password is **Moderate** if it contains at least 8 characters but has no uppercase letters, numbers, or special characters.
3. A password is **Strong** if it contains at least 8 characters, includes both uppercase and lowercase letters, has at least one digit, and has at least one special character (!@#\$%^&\*()).

Encourage students to use basic string operations, loops, and conditionals to check these criteria.


#### Example input/output:

1. Enter a password to check its strength: abc  
Password strength: Weak
2. Enter a password to check its strength: abcdefghij  
Password strength: Moderate
3. Enter a password to check its strength: Abcdefl!  
Password strength: Strong

#### Tips:

- ✓ **Check for length** of the password.
- ✓ **Check for the presence of uppercase and lowercase letters, digits, and special characters.**
- ✓ **Classify** the password based on the criteria mentioned above.

***Note: Do not use AI tools to generate your solution. Think through how you would logically check each condition and write your code accordingly.***

 At the end of this lab, you need to produce one file named: **full\_name\_W01\_Lab.py**

- *Example: Chan\_Dara\_W01\_Lab.py*