

# Lab Session

## Week 1

### Objective:

The goal of this lab is to set up the foundational tools required for programming robots in Python. Students will install and configure Visual Studio Code (VS Code), and link GitHub for version control.

Students work individually.

---

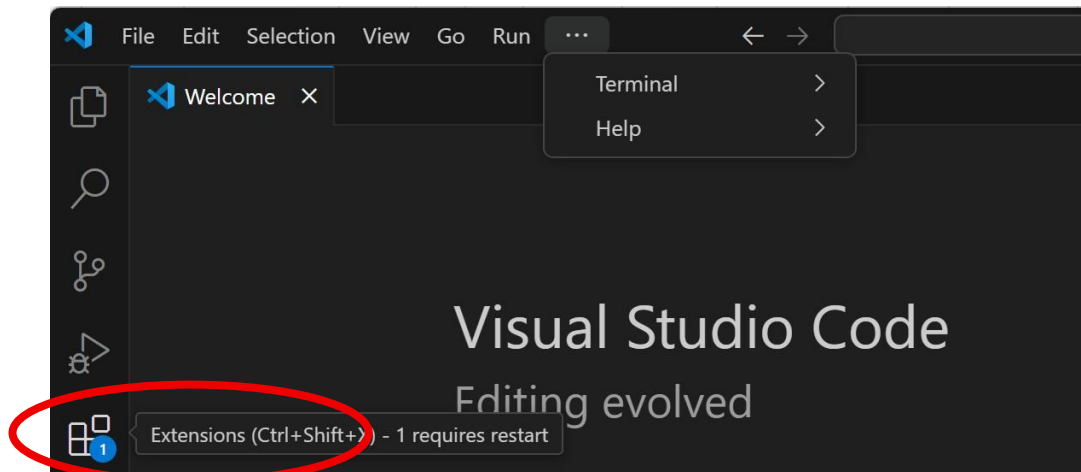
### PART 1 – Setting Up

#### Step 1: Install Python Interpreter

1. Go to the official Python website: <https://www.python.org/downloads/>
2. Click the download link for the latest stable version (preferably Python 3.11+ or newer).
3. Run the installer.
  - **Important:** On the first install screen, check the box labeled **Add Python to PATH** before clicking "Install Now." This makes Python accessible from the command line.
4. Mac users: You can install Python via the terminal with Homebrew using `brew install python3` if you prefer command-line tools.

#### Step 2: Set up Visual Studio Code (VS Code)

1. Download VS Code from <https://code.visualstudio.com/> and install it.
2. Launch VS Code after installation.
3. Open the **Extensions** view by clicking the square icon on the left sidebar or pressing `Ctrl+Shift+X`.

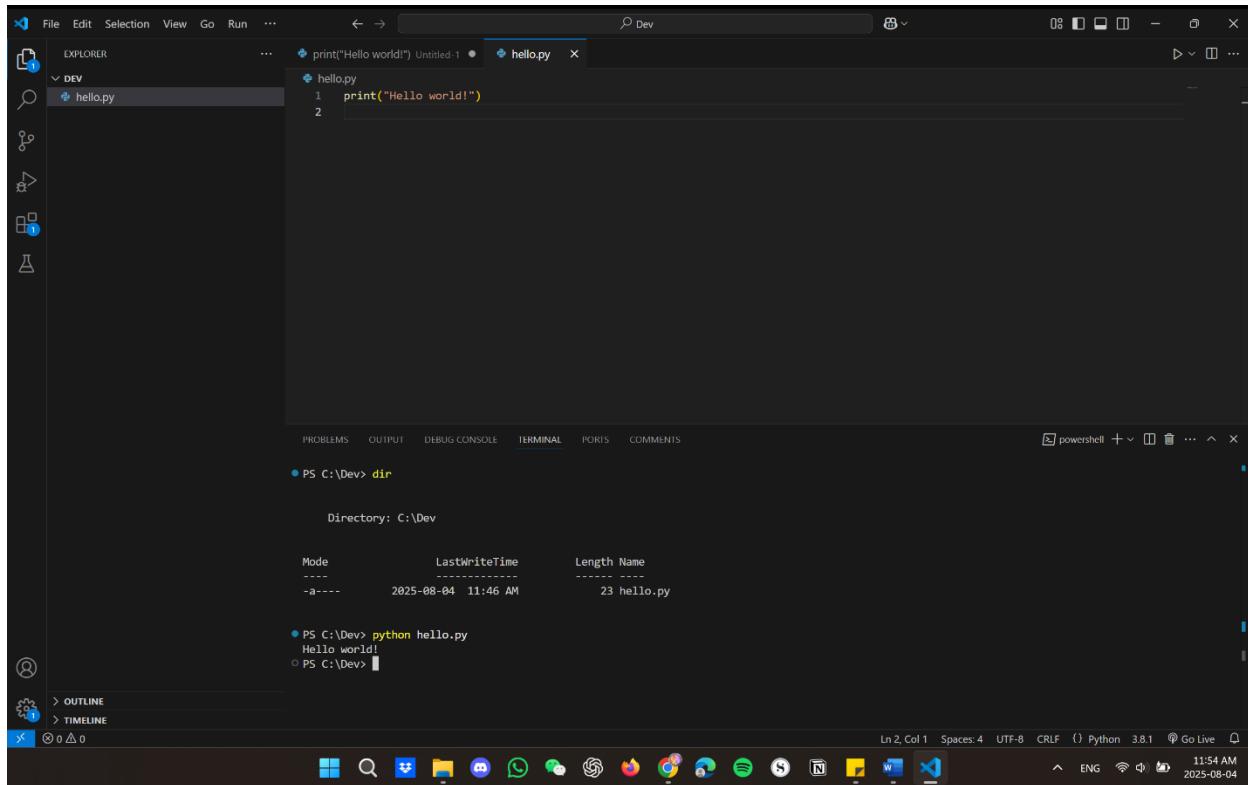


4. In the Extensions search box, type Python.
5. Find the Python extension published by Microsoft and click **Install**.
6. (Optional) Find GitHub Copilot and click **Install**.
7. After installation, open the Command Palette (Ctrl+Shift+P), type and select **Python: Select Interpreter**.
8. From the list, choose the Python interpreter you installed in Step 1.
9. Create a new Python file by choosing **File > New File**, then save it with .py extension (e.g., hello.py).
10. Write a simple program, like:

```
print("Hello, Python!")
```

11. Run the program by right-clicking inside the editor and selecting **Run Python File in Terminal** or press F5 to debug.

(Mac note: VS Code installation and Python extension setup is identical on Mac.)



### Step 3: Install Git and Register on GitHub

1. Download Git for Windows from <https://git-scm.com/download/win> and run the installer with default settings.
2. After installation, open **Git Bash** (installed alongside Git), or open **PowerShell**.
3. Verify Git installation by running:

```
git --version
```

4. Register for a free account on GitHub: <https://github.com/join>
5. After registration, you can set up Git to connect with GitHub by configuring your username and email:

```
git config --global user.name "Your Name"
git config --global user.email "your.email@example.com"
```

6. To link repositories with GitHub, you will eventually generate SSH keys or use HTTPS authentication; details are in GitHub's official docs.

(*Mac note:* Git is usually pre-installed on Mac. You can check by `git --version`. Registering on GitHub and configuring is the same.)

## PART 2 – Python Refresher

Refresh your Python programming knowledge quickly:

1. **Download the Python Cheat Sheet:**

Go to the official "Python Crash Course" cheat sheets by Eric Matthes at this link:  
[https://ehmatthes.github.io/pcc\\_3e/cheat\\_sheets/](https://ehmatthes.github.io/pcc_3e/cheat_sheets/)

2. Use the Cheat Sheet to review key Python syntax like variables, data types, loops, functions, and classes. Use it while practicing your Python exercises to remind yourself of code structures and commands.

---

### Exercise 1: Variables and Input/Output

Create a file in VS Code and call it `name.py`

Write a program that asks the user for their name and age, then prints a greeting message.

Sample solution:

```
name = input("Enter your name: ")
age = input("Enter your age: ")
print(f"Hello, {name}. You are {age} years old.")
```

**Exercise 2: Conditional Statements**

Create a file in VS Code and call it `num.py`

Write a program that checks if a number entered by the user is positive, negative, or zero.

Sample solution:

```
num = float(input("Enter a number: "))
if num > 0:
    print("The number is positive.")
elif num < 0:
    print("The number is negative.")
else:
    print("The number is zero.")
```

**Exercise 3: Loops**

Create a file in VS Code and call it `range.py`

Write a program that prints all even numbers from 1 to 20 using a loop.

Sample solution:

```
for i in range(1, 21):
    if i % 2 == 0:
        print(i)
```

**Exercise 4: Functions and Loops Combined**

Create a file in VS Code and call it `factorial.py`

Write a function that takes a number `n` and prints the factorial of `n`.

Sample solution:

```
def factorial(n):
    result = 1
    for i in range(1, n + 1):
        result *= i
    return result

num = int(input("Enter a number: "))
print(f"The factorial of {num} is {factorial(num)}")
```

### Exercise 5: Robot Speed Monitor

#### Problem Statement:

Robots in a factory are moving on a track. You are asked to monitor their speed to make sure none are moving too fast or too slow.

Write a Python program that:

1. Asks the user how many robots they want to check (min 1, max 5).
2. For each robot, ask for its speed (in cm/s).
3. Use a function `check_speed(speed)` to evaluate the speed:
  - If `speed < 10` → "Too slow"
  - If `speed > 50` → "Too fast"
  - Else → "Speed OK"
4. Print the result for each robot using this format:  
"Robot 1: Speed OK"  
"Robot 2: Too slow"  
etc.

Create a file in VS Code and call it `monitor.py`

### Lab Submission

1. Create a folder on your local computer named **Week 1**.
2. Place all Python code and files you created in Week 1 inside this folder.

3. In your GitHub account, create a repository named ***AIRO4000\_FirstLastname*** (replace *FirstLastname* with your first and last names, first letter capitalized).
4. Add the Week 1 folder to this repository. Each week you will create a new folder (Week 2, Week 3, etc.) inside the same repository, so all weeks are organized in one place.
5. **Invite your professor** as a collaborator on the repository.
6. **Submit the link** to your repository on **D2L Brightspace** before the end of Week 1's lab session.