Faculty Development Program (FDP) on Next Gen B.Pharm Curriculum

Organised By

Institute of Pharmacy, School of Life Sciences
Assam Don Bosco University, Tapesia Campus
In collaboration with the Pharmacy Council of India (PCI)

Participant Handout

Foundations of Computing & Environment Setup

Resource Person: Dr. Sonia Sarmah
Assistant Professor
Department of Computer Applications
Assam Don Bosco University

---

1. Introduction

Computing has become an integral part of modern healthcare and pharmaceutical sciences. From electronic health records and hospital information systems to bioinformatics, drug discovery, clinical decision support systems, and artificial intelligence, computing tools are transforming the pharmacy profession.

Python has emerged as one of the most widely used programming languages in healthcare due to its simplicity, extensive scientific libraries, and strong support for data analysis and machine learning. Before learning advanced topics, it is important to understand the basic computing environment and the tools required for Python programming.

This handout introduces the essential concepts required to set up a Python programming environment and understand the basics of Python.

---

2. Computer Operating Systems

An **Operating System (OS)** is **system software** that manages the computer's hardware resources and provides essential services for application programs. It acts as an **interface between the user and the computer hardware**, ensuring that software and hardware work together efficiently. Without an operating system, users would not be able to interact with the computer or run applications such as web browsers, word processors, or games.

Common Operating Systems

- **Microsoft Windows** – The most widely used operating system for personal computers, known for its user-friendly graphical interface.
- **Linux** – An open-source operating system widely used in servers, cloud computing, embedded systems, and by developers.
- **macOS** – Apple's operating system designed for Mac computers, known for its performance, security, and seamless integration with Apple devices.

Major Functions of an Operating System

1. File and Folder Management

The OS organizes, stores, retrieves, copies, moves, renames, and deletes files and folders. It also manages permissions and maintains the file system structure.

2. Memory Management

The OS allocates memory (RAM) to different programs, keeps track of memory usage, and frees memory when programs are closed, ensuring efficient utilization of system resources.

3. Device Management

The OS controls communication between the computer and hardware devices such as printers, keyboards, monitors, USB drives, and scanners through device drivers.

4. Process Management

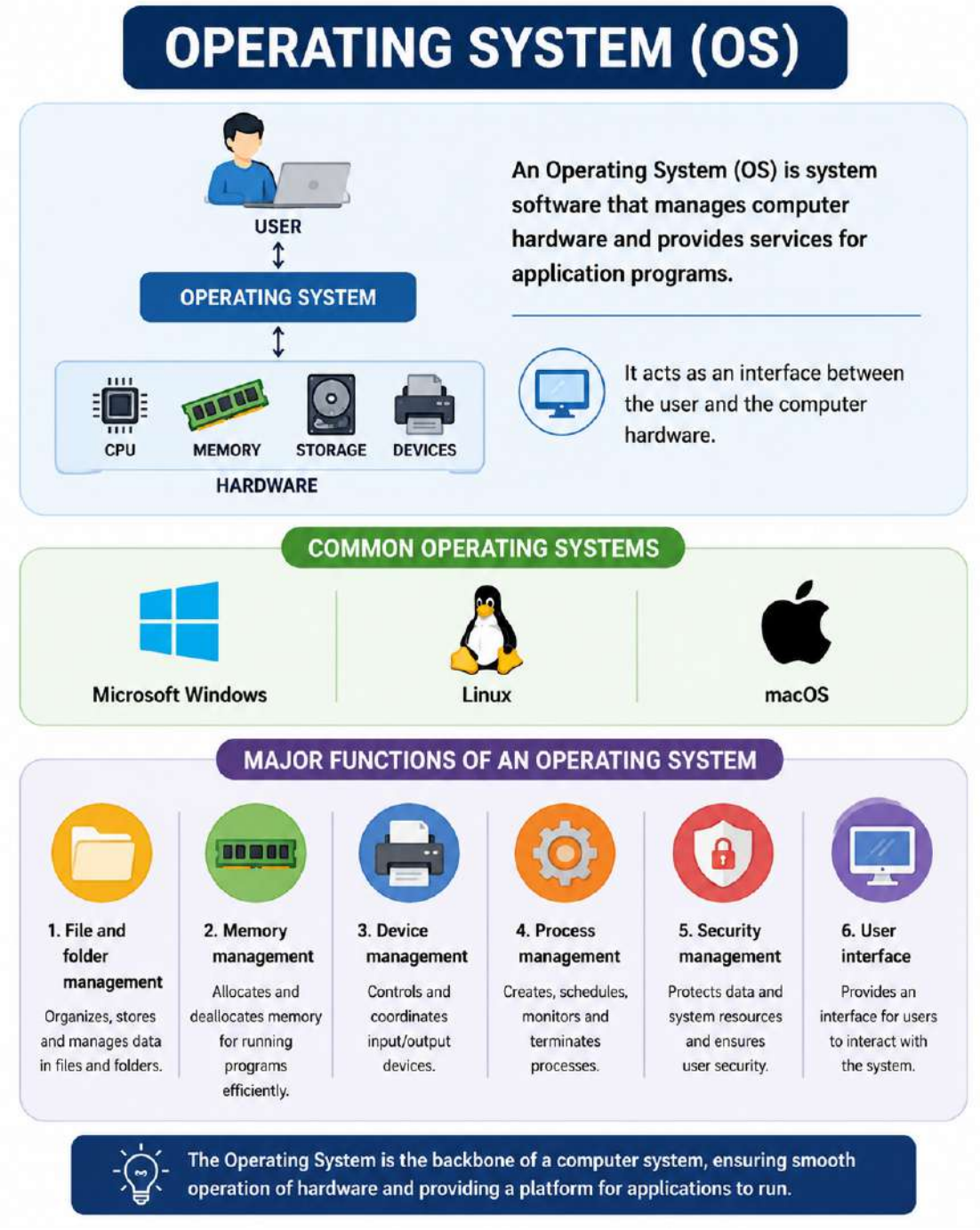
The OS creates, schedules, executes, and terminates processes. It enables multitasking by allowing multiple applications to run simultaneously while efficiently sharing CPU resources.

5. Security Management

The OS protects the system from unauthorized access through user authentication, passwords, permissions, encryption, and other security mechanisms.

6. User Interface

The OS provides a **Graphical User Interface (GUI)** or **Command Line Interface (CLI)** that allows users to interact with the computer, execute commands, and access applications easily.



Useful Shortcuts

Function	Windows	macOS	Linux (Ubuntu/ GNOME)
Open File Manager	**Win + E**	**⌘ + N** (Finder)	**Super + E**
Open Run/Search	**Win + R**	**⌘ + Space** (Spotlight)	**Alt + F2**
Copy	**Ctrl + C**	**⌘ + C**	**Ctrl + C**
Paste	**Ctrl + V**	**⌘ + V**	**Ctrl + V**
Cut	**Ctrl + X**	**⌘ + X**	**Ctrl + X**
Undo	**Ctrl + Z**	**⌘ + Z**	**Ctrl + Z**
Switch Between Applications	**Alt + Tab**	**⌘ + Tab**	**Alt + Tab**
Lock Screen	**Win + L**	**Control + ⌘ + Q**	**Super + L**

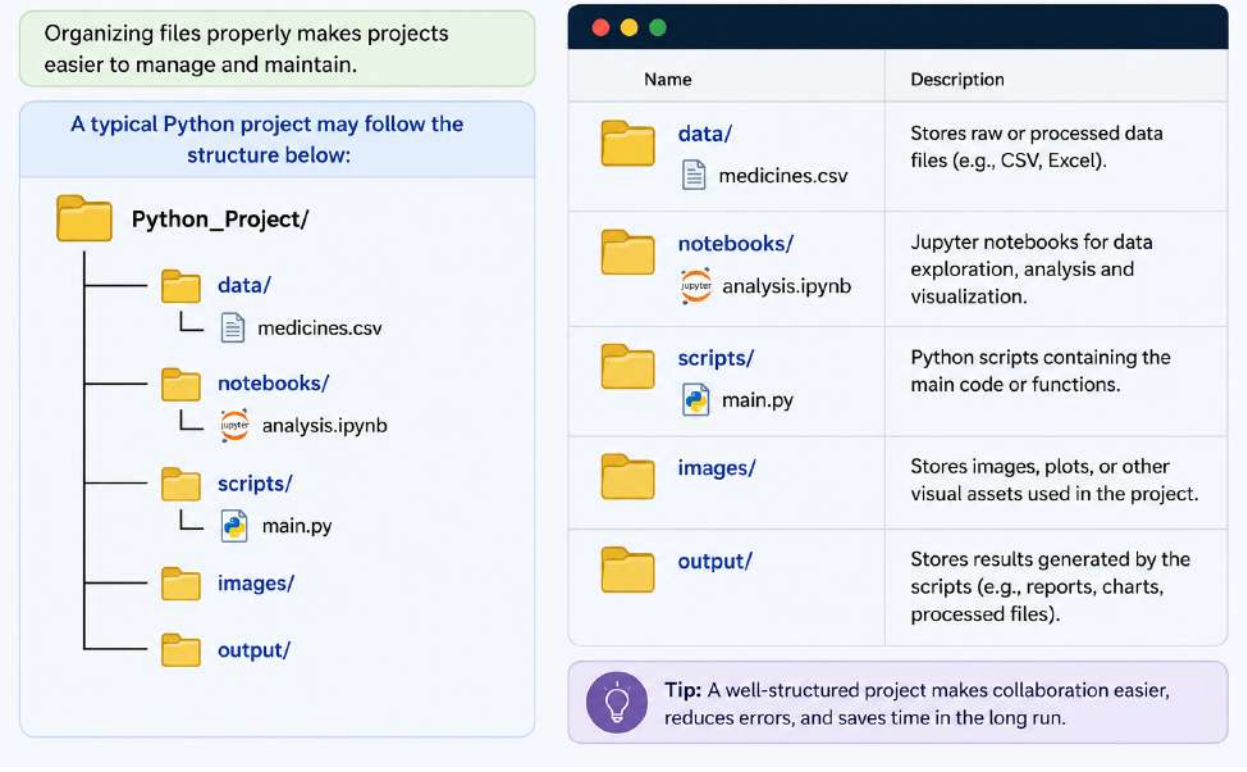
---

3. Understanding File Architecture

Organizing files properly makes projects easier to manage and maintain.

A typical Python project may follow the structure below:

```
Python_Project/
│
├── data/
│   └── medicines.csv
│
├── notebooks/
│   └── analysis.ipynb
│
├── scripts/
│   └── main.py
│
├── images/
│
└── output/
```



Common File Extensions

Extension	Description
.py	Python source code
.ipynb	Jupyter Notebook
.csv	Comma Separated Values dataset
.xlsx	Microsoft Excel workbook
.txt	Plain text file
.pdf	Portable Document Format
.jpg/.png	Image files

Good Practices

- Create separate folders for datasets, programs, and outputs.
- Use meaningful file names.
- Avoid spaces and special characters in file names.
- Keep regular backups of your work.

---

4. Understanding System Properties

Before installing Python or any software, it is useful to know your computer's specifications.

Important system information includes:

- Processor (CPU)
- Installed RAM
- Storage (HDD/SSD)
- Operating System Version
- System Type (32-bit or 64-bit)

How to View System Information (Windows)

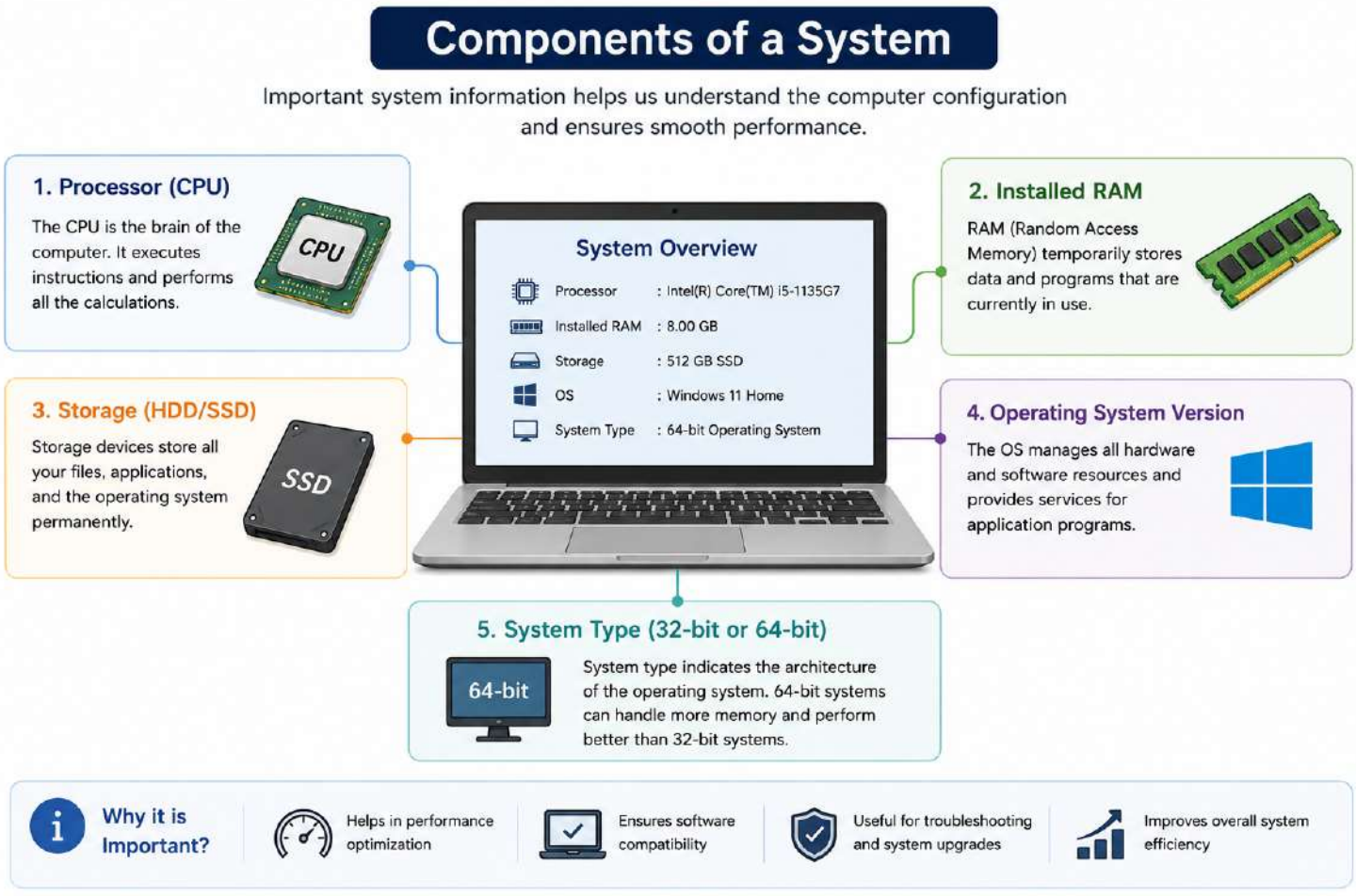
Settings → System → About

How to View System Information (Linux)

Settings → About *(GNOME/Ubuntu and many desktop environments)*

How to View System Information (macOS)

Apple Menu (🍎) → About This Mac



---

5. Python Development Environment

Python programs can be written using different development environments (IDEs).

Jupyter Notebook

Jupyter Notebook is an interactive environment that allows users to combine code, text, equations, and visualizations within a single document.

JUPYTER NOTEBOOK

Jupyter Notebook is an interactive environment that allows users to combine code, text, equations, and visualizations within a single document.

SUITABLE FOR

- Learning Python
- Data analysis
- Scientific computing
- Machine learning

STEPS IN USING JUPYTER NOTEBOOK

1 Install Jupyter Notebook

Install using pip:

```
pip install notebook
```

(or use **Anaconda** Distribution)

2 Launch Jupyter Notebook

Open terminal or Anaconda Prompt and run:

```
jupyter notebook
```

This opens Jupyter in your default browser.

3 Create or Open a Notebook

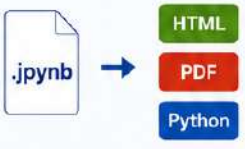
Click "New" and select "Python 3" to create a new notebook or open an existing one (.ipynb file).

4 Write and Run Your First Code

Type the code in a cell and press **Shift + Enter** (or click **Run ▶**) to execute.
The output appears below the cell.

5 Save and Export

Notebooks are auto-saved.
You can also save manually (**File → Save and Checkpoint**).
The file is saved with **.ipynb** extension.



Export notebook as HTML, PDF, or Python script (**File → Download As**).
Share the .ipynb file to reproduce your work.

TIP Use notebooks to document your analysis, explain your thoughts, and combine code with results for better understanding.

Suitable for

- Learning Python
- Data analysis
- Scientific computing
- Machine learning

Visual Studio Code (VS Code)

VS Code is a lightweight and powerful source-code editor developed by Microsoft.

Features include:

- Intelligent code completion
- Integrated terminal
- Debugging support
- Extension marketplace
- Git integration

PyCharm

PyCharm is a professional Integrated Development Environment specifically designed for Python development.

It provides:

- Advanced code analysis
- Project management
- Built-in debugger
- Version control integration
- Testing support

VISUAL STUDIO CODE (VS CODE)

VS Code is a **lightweight** and powerful source-code editor developed by Microsoft.

STEPS IN INSTALLING AND USING VS CODE

1 Download VS Code

Go to the official website:
https://code.visualstudio.com/

Click the Download button for your operating system (Windows, macOS, or Linux).

2 Install VS Code

- Run the downloaded installer.
- Follow the setup instructions.
- (Windows) Make sure to check "Add to PATH" option.
- Click "Install" and then "Finish".

3 Launch VS Code

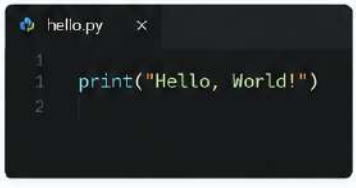
Open VS Code from the Start menu (Windows), Applications (macOS), or your app menu (Linux).

4 Open a Folder / Workspace

Click on File > Open Folder... and select your project folder. This is where your files and folders will be managed.

5 Create or Open a File

- Click the New File icom to create a file.
- Save the file with an appropriate extension (e.g., .py, .js, .html).
- Start writing your code. VS Code provides intelligent suggestions as you type.



6 Use the Integrated Terminal

- Open the terminal with (Ctrl + ` (backtick) or View > Terminal.
- Run commands, install packages, and execute your programs without leaving VS Code.

7 Run and Debug Your Code

- Click the Run ▶ button to run your code.
- Use the Debug view (Ctrl + Shift + D) to set breakpoints, inspect variables, and debug efficiently.

8 Git Integration

- VS Code has built-in Git support.
- Initialize a repository or open an existing one.
- Use the Source Control view (Ctrl + Shift + G) to commit, push, pull, and manage changes.

Useful Shortcuts

Ctrl + P	: Quick Open (files, symbols, commands)
Ctrl + Shift + P	: Command Palette
Ctrl + `	: Toggle Terminal
Ctrl + /	: Toggle Line Comment
Ctrl + S	: Save File

TIP VS Code is highly customizable. Use themes, settings, and extensions to create your perfect coding environment!

PYCHARM

PyCharm is a professional Integrated Development Environment specifically designed for Python development.

IT PROVIDES:

- Advanced code analysis
- Project management
- Built-in debugger
- Version control integration
- Testing support

STEPS IN INSTALLING AND USING PYCHARM

1 Download PyCharm

- Go to the official website: https://www.jetbrains.com/pycharm/
- Click the Download button and choose the edition:
  - Professional (paid)
  - Community (free)
- Select your operating system (Windows, macOS, or Linux).

2 Install PyCharm

- Run the downloaded installer.
- Follow the setup instructions.
- (Windows) Make sure to check "Add to PATH" option.
- Click "Install" and then "Finish".

3 Launch PyCharm

- Open PyCharm from the Start menu (Windows), Applications (macOS), or your app menu (Linux).
- On first launch, you may be asked to import settings or install plugins – choose as needed.

4 Create or Open a Project

- Click "New Project" to create a new Python project.
- Choose a location and Python interpreter.
- Or click "Open" to open an existing project folder.

5 Create a Python File

- Right-click on the project folder (in Project view) → New → Python File.
- Enter a file name (e.g., hello.py) and click OK.
- The file will open in the editor.

6 Write and Run Your Code

```
hello.py
1
1  print("Hello, World!")
2
```

- Write your Python code in the editor.
- Click the green ▶ Run button (top right) or right-click → Run 'hello'.
- Output will appear in the Run tool window below.

7 Use the Built-in Debugger

- Set breakpoints by clicking on the left gutter next to line numbers.
- Click the bug icon or press Shift + F9 to start debugging.
- Use the Debug tool window to step through code, inspect variables, etc.

8 Version Control Integration

- Click the VCS icon (or go to VCS menu).
- Enable Version Control Integration (e.g., Git).
- Commit, push, pull, and manage branches directly from PyCharm.

9 Run Tests

- PyCharm supports unittest, pytest, and other testing frameworks.
- Right-click on the test file or function → Run 'Tests'.
- View results in the Run tool window.

TIP

Useful Shortcuts

Ctrl + N	New File
Ctrl + Shift + N	New Project
Ctrl + R	Run
Shift + F9	Debug
Ctrl + /	Comment / Uncomment

PyCharm is highly productive!

Explore settings, plugins, and customizations to make it work the way you like.

Comparison of IDEs

IDE	Best Used For
Jupyter Notebook	Learning, Data Analysis
VS Code	General Python Programming
PyCharm	Large Software Projects

---

6. Installing Python

Installation Steps

1. Download Python from the official website. https://www.python.org/downloads/
2. Run the installer.
3. Select **“Add Python to PATH.”**
4. Complete the installation.
5. Verify the installation using the command:

```
python --version
```

To check the installed package manager:

```
pip --version
```

7. Python Package Management using pip

Python includes **pip**, the default package manager used to install external libraries.

Frequently Used Commands

Install a package

```
pip install pandas
```

Install multiple packages

```
pip install numpy matplotlib scipy
```

Upgrade a package

```
pip install --upgrade pandas
```

View installed packages

```
pip list
```

Display package details

```
pip show pandas
```

Remove a package

```
pip uninstall pandas
```

Export installed packages

```
pip freeze
```

Common Installation Issues

Problem	Possible Cause	Solution
Python not recognized	PATH not configured	Reinstall Python and enable “Add Python to PATH”
ModuleNotFoundError	Required package missing	Install using pip
Permission denied	Limited user privileges	Run terminal as Administrator

---

8. Python Fundamentals

Python is a high-level, interpreted, and easy-to-learn programming language.

Writing Your First Program

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

Output

```
Hello, Pharmacy!
```

Variables

Variables store data values.

```
medicine = "Paracetamol"
price = 25
temperature = 37.5
available = True
```

Basic Data Types

Data Type	Example	Description
Integer	25	Whole numbers
Float	37.5	Decimal numbers
String	“Tablet”	Text values
Boolean	True	Logical values

Checking Data Types

```
age = 30
print(type(age))
```

Output

```
<class 'int'>
```

Type Casting

Convert one data type into another.

```
quantity = "50"
quantity = int(quantity)

price = 25
price = float(price)
```

---

9. Practice Exercise

Write a Python program that stores the following information using variables:

- Medicine Name
- Manufacturer
- Price
- Expiry Year
- Availability

Display all the values using the `print()` function.

10. Key Takeaways

At the end of this session, you should be able to:

- Understand the role of an operating system.
- Organize files and folders effectively.
- Check important system specifications.
- Install Python and development environments.
- Use the pip package manager.
- Create variables in Python.
- Work with basic data types.
- Perform simple type conversions.

Additional Resources

- Python Download : https://www.python.org/downloads/
- Official Python Documentation: https://docs.python.org
- Jupyter Project: https://jupyter.org
- Visual Studio Code: https://code.visualstudio.com
- PyCharm: https://www.jetbrains.com/pycharm
- How to install Python 3.13.5 on Windows 11
- Install Python 3.12 and PyCharm on Windows 10/11
- How to Run Python 3.14 in Visual Studio Code on Windows 11 | Run Sample Python Program
- How to install Jupyter Notebook on Windows 11
- Basics of Python