Python Course Notes :

Session 1 :

1) An Overview of Python

- **Python** is a popular, easy-to-read programming language used for web apps, data science, automation, AI/ML, scripting, and more.

- It focuses on **readable code** (indentation matters) and has a huge set of ready-to-use tools ("**batteries included**").

- Runs on **Windows, macOS, Linux**. It's **free and open source**.

2) What is Python?

- A **language** (the way you write instructions) + an **interpreter** (the program that runs those instructions).

- You usually save code in files ending with `.py` and run them with `python` or `python3`.

Example:

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

- Python has an interactive prompt called the **REPL** (run `python` and type code line by line).

3) Interpreted Languages (in simple words)

- With **interpreted** languages like Python, code is run **line by line** by an interpreter.
- **Pros:** quick to try ideas, easy to debug, works the same on different OSes.
- **Cons:** usually **slower** than languages compiled to machine code ahead of time (like C/C++).
- Note: Python first makes **bytecode** internally (`.pyc`) and then runs it—still considered "interpreted."

4) Advantages & Disadvantages of Python

Advantages

- **Easy to learn & read** (great for beginners).
- **Huge community & libraries** (install with `pip`).
- **Cross-platform**; strong support for data/AI/web/automation.
- **Rapid development** and great **interactive** workflow.

Disadvantages

- **Speed:** slower than compiled languages for heavy CPU work.
- **GIL (Global Interpreter Lock):** limits multi-threading for CPU-bound tasks (workarounds: multiprocessing, C extensions, PyPy, etc.).

- **Packaging to a single EXE** needs extra tools (e.g., PyInstaller).
- **Memory use** can be higher than some alternatives.

5) Downloading & Installing (quick start)

Windows

1. Go to **python.org** → **Downloads** → **Windows** and get the **64-bit** installer.
2. Check **"Add Python to PATH"** during install.
3. Verify: open **Command Prompt** → `python --version` (or `python3 --version`).

macOS

1. Easiest: **Homebrew** → `brew install python` (or `python@<version>`).
   (Install Homebrew first from brew.sh)
   OR download from **python.org**.
2. Verify: **Terminal** → `python3 --version`.

Linux (Ubuntu/Debian)

- `sudo apt update && sudo apt install -y python3 python3-pip`
- For newer versions: consider **pyenv** or **Anaconda**.
- Verify: `python3 --version`.

Create a project folder & virtual environment (all OS)

```
# in your project folder
python -m venv .venv        # or: python3 -m venv .venv
# activate:
# Windows:
.\.venv\Scripts\activate
# macOS/Linux:
source .venv/bin/activate

pip install requests        # example package
```

python your_script.py

Editors

- **PyCharm / VS Code** (free) + Python extension (great for beginners).
- **IDLE** comes with Python.
- **Jupyter** for notebooks: `pip install notebook` then `jupyter notebook`.

6) Which Version of Python?

- Use the **latest stable Python 3.x** (avoid Python 2—it's retired).
- If a course or library requires a specific version (e.g., 3.10/3.11/3.12+), **match that version**.
- Prefer **64-bit**.
- For data science, **Anaconda** can simplify package management; otherwise **python.org + pip** is perfectly fine.
- Use **virtual environments** so each project has its own packages.

7) Where to Find Documentation & Help

- **Official docs & tutorial:** search "**Python documentation**" (has tutorials, library references, how-tos).
- **Built-in help:** in Python, try `help(print)`, `help(str)`, or `dir(object)`.
- **Package info:** `pip show <package>` after installing.
- **Beginner-friendly guides:** look for reputable sites and the official tutorial; stick to recent articles.

Tiny "First Program" Checklist

1. Install Python 3 (64-bit), add to PATH.
2. Open VS Code/ PyCharm → create `hello.py`:

```
name = input("Your name: ")

print("Hello,", name)
```

3. Run it:
    - Terminal → `python hello.py` (or `python3 hello.py`).
4. Make a virtual environment for real projects and install libraries with `pip`.