My Python Progress

Day 1 – Monday

Lecture 1: Introduction to Python – Data Types & Basics

Topics Learned:

- **Data Types**: Basic kinds of values like integers (`int`), floating numbers (`float`), text (`str`), and true/false (`bool`).
- **Print and Sum**: Use `print()` to show output and `+` to add numbers.
- **Comments**: Notes in code using `#`, ignored by Python.
- **Operators**: Symbols like `+`, `-`, `*`, `/`, `==` used to perform actions or compare values.
- **Input Function**: Take input from the user using `input()`.
- **Type Conversion**: Change data types using `int()`, `float()`, `str()`, etc.
- **First Program**: A basic interactive script using input/output.
- **Practice**: Created mini-programs using learned topics.

Lecture 2: Strings and Conditionals

Topics Learned:

- **String Concatenation**: Combine strings using `+`.
- **Indexing**: Access characters by their position (e.g. `name[0]`).
- **Slicing**: Extract parts of a string (e.g. `name[1:4]`).
- **`len()` Function**: Find the length of a string.
- **Conditional Statements**: Use `if`, `elif`, and `else` to make decisions.
- **Nested Conditionals**: Place one `if` statement inside another.
- **Practice**: Built basic logic programs using strings and conditionals.

Lecture 3: Lists and Tuples

Topics Learned:

- **List**: An ordered collection that can be changed (mutable).
- **List vs String**: Lists can hold multiple data types; strings are just characters.
- **List Methods**: Functions like `.append()`, `.remove()`, `.sort()` used to modify lists.
- **Tuple**: An ordered collection that cannot be changed (immutable).
- **Tuple vs List**: Tuples are faster and cannot be updated after creation.
- **Tuple Slicing**: Extract parts of a tuple using index ranges.
- **Tuple Methods**: Common methods like `.count()` and `.index()`.
- **Practice**: Worked on list and tuple-based data problems.

Day 2 – Tuesday

Lecture 4: Dictionaries and Sets

Topics Learned:

- **Dictionary**: A key-value pair data structure (`{"name": "Ali"}`).
- **Nested Dictionary**: A dictionary inside another dictionary.
- **Dictionary Methods**: Useful functions like `.get()`, `.keys()`, `.values()`.
- **Set**: A collection of unique, unordered items.
- **Set Methods**: Functions like `.add()`, `.remove()`, `.union()`.
- **Practice**: Built programs using dicts and sets to store and access data.

Day 3 – Wednesday

Lecture 5: Loops and Iteration

Topics Learned:

- **While Loop**: Repeats a block of code while a condition is true.
- **For Loop**: Iterates over sequences like lists or a range of numbers.
- `range()` **Function**: Generates numbers in a specified range.
- **Break Statement**: Exits the loop early.
- **Continue Statement**: Skips current loop iteration.
- **Pass Keyword**: Does nothing — used as a placeholder.
- **Practice**: Built loop-based tasks like countdowns, tables, factorials.

Day 4 – Thursday

Lecture 6: Functions in Python

Topics Learned:

- **Functions**: Blocks of reusable code defined using `def`.
- **Built-in Functions**: Pre-made functions like `print()`, `len()`.
- **User-defined Functions**: Custom functions made by the programmer.
- **Function Parameters**: Input values passed to functions.
- **Recursive Functions**: Functions that call themselves (e.g., factorial).
- **Logic with Loops & If/Else**: Combined for solving real problems.
- **Practice**: Wrote and reused functions for different tasks.

Day 5 – Friday

Lecture 7: Object-Oriented Programming (OOP)

Topics Learned:

- **OOP**: A method of structuring code using classes and objects.
- **Class and Object**: A class is a blueprint; an object is a real-world use of that blueprint.
- **`__init__()` Constructor**: Automatically runs when a new object is created.
- **Abstraction**: Hiding unnecessary internal details.
- **Encapsulation**: Keeping variables and methods inside one class.
- **Polymorphism**: Using one method in multiple ways (e.g., overriding).
- **Practice**: Built real-world models like Bank, Student using classes.

Common Programs You Practiced:

- Add two numbers
- Print name and age
- Temperature converter
- Simple interest calculator
- Find maximum of three numbers
- Even/Odd number check
- Prime number checker
- Multiplication table
- Reverse a string
- Vowel counter
- Bank app (OOP)
