Day 4 Training Report

26 June 2025

Hands-on Python Practice + Mini Exercises

On Day 4, the primary objective was to **strengthen the understanding of Python programming concepts** through a series of **practical exercises** and small problem-solving tasks. After learning the fundamentals of Python (variables, data types, loops, and conditionals) and exploring essential libraries (NumPy, Pandas, Matplotlib), this session focused on applying the concepts in real coding scenarios to build confidence and improve logical thinking.

---

1. Basic Python Programming Exercises

Students practiced simple but essential coding problems to revise the core concepts:

- **Arithmetic operations**: Writing Python programs to add, subtract, multiply, divide numbers, and find modulus.
- **String manipulation**: Using slicing, concatenation, and built-in functions like upper(), lower(), and replace().
- **List operations**: Creating lists, adding/removing elements, iterating over lists using loops, and performing basic computations.
- **Conditional statements**: Solving problems that required decision-making using if-elif-else.
- **Loops**: Writing for and while loops for pattern printing, factorial calculation, and summation tasks.

Example – Factorial Program:

```
num = 5
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print("Factorial of", num, "is", factorial)
```

---

2. NumPy and Pandas Practice

Hands-on tasks were given to **strengthen data handling skills** using NumPy and Pandas:

- Creating and manipulating NumPy arrays (1D and 2D)
- Performing vectorized operations and using mathematical functions like mean(), sum(), and reshape()
- Loading small datasets into Pandas DataFrames and performing operations such as:
    - Selecting rows and columns
    - Filtering data using conditions
    - Handling missing values

- Calculating basic statistics using describe() and info()

Example – Pandas DataFrame Filtering:

```
import pandas as pd

data = {'Name': ['Alice', 'Bob', 'Charlie', 'David'],
        'Age': [25, 30, 35, 28]}
df = pd.DataFrame(data)

# Filter rows where Age > 28
filtered = df[df['Age'] > 28]
print(filtered)
```

---

3. Mini Challenges

To enhance problem-solving skills, mini coding challenges were given, such as:

- Writing a program to check whether a number is **prime** or not.
- Reversing a string without using built-in functions.
- Generating the **Fibonacci sequence** using loops.
- Performing matrix addition and multiplication using NumPy.

These challenges helped improve **logical reasoning** and **algorithmic thinking**, preparing students for more complex AI tasks.