Training Day 5 Report

Date: 27 June 2025

Topic: Object-Oriented Programming (OOPs) in Python

Overview:

On Day 5, the training focused on the **fundamentals of Object-Oriented Programming (OOPs)** in Python. OOPs helps structure programs into reusable and organized blocks using **classes** and **objects**, making complex systems easier to design and maintain.

Key Concepts Covered:

1. **Classes and Objects**
    - Classes act as blueprints for objects.
    - Objects are instances of classes containing attributes (data) and methods (functions).

Example: Car Class

```
class Car:
    def __init__(self, a, b, c=True):
        self.wheel = a
        self.steering = b
        self.airbags = c

    def drive(self):
        print("This car can be driven.")

    def reverse(self):
        print("This car can be reversed.")

    def music(self):
        print("Music can be played in this car.")
```

2. **Student Class Example**
    - Attributes: name, age, marks.
    - Methods: input marks and calculate results.
    - Result is based on performance in English and Maths.
3. **ATM Simulation using OOPs**
    - Created an **ATM system** using class and methods.
    - Functions: PIN creation, deposit, withdraw, check balance.
    - Error handling for invalid PIN and insufficient balance.

Summary:

- Understood the **concept of classes and objects**.
- Learned to use **methods for functionality** and **attributes for data storage**.
- Applied OOPs in practical programs like **Car**, **Student**, and **ATM system**.

Learning Outcomes:

✔ Ability to define and use **classes and objects** in Python.
✔ Improved understanding of **real-world problem modeling** using OOPs.
✔ Learned to build **modular and reusable code** with attributes and methods.
✔ Gained hands-on experience in developing practical OOP-based programs.
✔ Understood the role of OOPs in making programs more **organized, scalable, and maintainable**.