Day 13

What is OOP:

- Object oriented programming
- A programming Paradigm that uses Objects and classes:
  - class & object
  - Encapsulation
  - Abstraction
  - inheritance
  - Polymorphism.

1. class and Object:

- A class is a blue print for creating Objects. It defines attributes and methods.

An object is an instance of a class.

Syntax:

```
class Car:
    def __init__(self, brand, model):
        self.brand = brand
        self.model = model

    def display(self):
        print(f"Car: {self.brand}, model: {self.model}")
```

- __init__ is a special constructor method used to initialize object attributes.

2. Encapsulation:

Encapsulation is data hiding. It restricts direct access to some variables and allow controlled access using getter and setter methods.

3. Abstraction:

Abstraction hides complex implementation details and only exposes the necessary functionalities to the user.

4. Inheritance:

Inheritance allows a class (child) to inherit properties and behaviours from another class (parent). This promotes code reusability.

5. polymorphism:

polymorphism allows different classes to use the same method name but perform different behaviours.

Concept.

1. class & object - Blueprint & instance creation
   eg: car ("Toyota", "camry")

2. Encapsulation - Data hiding using private variables.
   eg. balance in Bank Account.

3. Abstraction - hiding details and exposing only functionality.
   eg. @ Abstract method in Animal class

4. Inheritance - child class inheriting parent class properties.
   eg. Dog (Animal)

5. Polymorphism - Same method, different behaviour
   fly() method in Eagle and penguin.