🧮 Assignment: Class in Python

🎯 **Objective:**
To help students understand how to define and use classes in Python, including constructors, attributes, instance methods, class methods, inheritance, and method overriding.

🟢 Level 1 – Easy

✅ Task 1: Basic Class Creation

Create a class `Person` with the following:

- Attributes: `name`, `age`
- Method: `display_info()` that prints name and age
  Create an object of the class and call the method.

✅ Task 2: Constructor Usage

Enhance the `Person` class to include a constructor (`__init__`) that initializes `name` and `age`.

✅ Task 3: Class vs Instance Variables

Create a class `Counter`:

- Class variable: `count = 0`
- Method: `increment()` increases `count` by 1
  Create multiple objects and observe how `count` behaves.

🟡 Level 2 – Medium

✅ Task 4: Bank Account Class

Create a class `BankAccount` with:

- Attributes: `account_holder`, `balance`
- Methods: `deposit()`, `withdraw()`, `check_balance()`
- Initialize balance via constructor Simulate a few transactions.
-

✅ Task 5: Student Management

Create a class `Student`:

- Attributes: `name`, `roll_no`, `marks` (list of 5 subjects)
- Method: `calculate_percentage()`
  Create objects and compute percentage for each student.

✅ Task 6: Class Method and Static Method

In a class `MathUtility`, create:

- A class variable: `pi = 3.14`
- Class method: `circle_area(radius)`
- Static method: `is_even(n)`

🔴 Level 3 – Hard

✅ Task 7: Inheritance Example

Create a base class `Vehicle`:

- Attributes: `brand`, `year`
- Method: `display_info()`

Create derived classes:

- `Car`: add attribute `num_doors`
- `Bike`: add attribute `engine_capacity`

Override the `display_info()` method in each subclass.

🧮 Assignment: Class in Python

✅ Task 8: Employee Hierarchy

Create a class `Employee` with:

- Attributes: `name`, `salary`
- Method: `display()`
  Create a subclass `Manager`:
- Add attribute: `department`
- Override the `display()` method to include department info.

✅ Task 9: Magic Methods Practice

Create a class `Product`:

- Attributes: `name`, `price`
- Override `__str__` to print the product nicely
- Override `__eq__` to compare products by price