Assignment # 05: Class and Objects (Encapsulation and Inheritance)

Objective:

To help you understand:

- What are Classes, Object, Methods.
- How to implement Encapsulation and Inheritance in class.
- What are access modifiers and its types, and how to hide/restrict attributes and methods access by using access modifiers.
- How to implement single-level, multi-level, and multiple inheritance (Types of Inheritance).

Note: ***For each class in the tasks you have to make three objects and verify the logics implemented in the class***

.

Part 1: Encapsulation Practice Tasks:

Task 1: Bank Account

- Create a class **BankAccount** with private attributes **account_number** and **balance**.
- Add methods:
  - **deposit(amount)**
  - **withdraw(amount) (should not allow balance < 0)**
  - **get_balance()**

Task 2: Student Marks Validation

- Create a class **Student** with private attributes **name, roll_no, marks**.
- Add methods:
  - **Setter** method for marks (only accept values between 0–100).
  - **Getters** methods for all attributes.

Task 3: Password Manager

- Create a class **PasswordManager** with public attributes **username** and private attribute **password**.
- Add methods:
    - **set_password(old, new)** → update only if old matches.
    - **check_username(name)** → for checking the input user name. (Private method)
    - **check_password(input)** → return True/False. (Private Method)

Task 4: Employee Salary Protection

- Create a class **Employee** with private attributes **name**, **salary**.
- Add methods:
    - **Getter** for name.
    - **Setter** for salary (must be positive).
    - Method **show_details()**.

Task 5: Shopping Cart

- Class **ShoppingCart** with private list **items**.
- Add methods:
    - **add_item(item)**
    - **remove_item(item)**
    - **view_cart()** → return items of list.
- Ensure no duplicate items allowed.

Part 2: Inheritance Practice Tasks

Task 1: Single Level – Animal

- Parent class **Animal** with method **make_sound()**.
- Child class **Dog** overrides **make_sound()** with **"Bark!"**.
- Show method overriding.

Task 2: Single Level – Vehicle

- Parent class **Vehicle** with attributes **brand**, **model**.
- Child class **Car** with attribute **seats**.

- Create object of **Car** and display details.

Task 3: Multi-Level – Family Tree

- Create a class **GrandParent** with method **family_name()**.
- **Parent** class inherits **GrandParent**. Add method **occupation()** in **Parent** class.
- **Child** inherits **Parent**. Add method **hobby()** in **Child** class.
- Create **Child** object and call all methods.

Task 4: Multiple Inheritance – Skills

- Create **Father** class with method **skills()** returns **"Scientist"**.
- Create **Mother** class with method **skills()** returns **"Freelancer"**.
- Create **Child** class which inherits both **skills()** methods of Parents. Create method **skills()** in **Child** class which combines **skills** of both parent classes into "Scientist and Freelancer".