Assignment – 2 Group Assignment: Object-Oriented : Transaction System

Objective:

Design and implement a simple system of your choice using Object-Oriented Programming (OOP) principles in Python. The system should allow users to create entities and perform transactions between them. Your implementation must demonstrate the following core OOP concepts:

- **Encapsulation**: Use private or protected attributes to protect data inside classes.
- **Inheritance**: Use at least one base class and one or more derived classes.
- **Polymorphism**: Use method overriding or method overloading to show polymorphic behavior.

Examples (you can choose):

- Banking system (accounts, transactions)
- Library system (members, borrowing/returning books)
- Online store (users, orders, payments)
- Game inventory system (players, items, trades)
- Any other system involving transactions or state changes

Requirements:

1. Classes and Inheritance
   Create a base class representing a general entity, and at least two subclasses that extend the base class.
2. Encapsulation
   - Use private or protected variables to store sensitive data (e.g., account balance, book status).
   - Provide getter and setter methods or use properties for controlled access.
3. Transactions
   - Implement a way to perform transactions between entities (e.g., transfer money, borrow/return items).
4. Polymorphism
   - Override methods in subclasses to demonstrate different behaviors for the same method name.
   - For example, a method called perform_transaction might behave differently in each subclass.
5. User Interaction
   - Implement simple input/output to allow the user to create objects and perform transactions.
   - Display appropriate messages and statuses after each transaction.

Evaluation Criteria:

- Correct usage of Classes & Objects
- Implementation of Encapsulation, Inheritance, and Polymorphism
- Functional transactions
- Code readability and documentation
- Additional features

Submission Requirements

Each group must submit:

- Video Recording of Presentation
- Python program with comments
- A group presentation explaining their approach

Grading Criteria (Total: 10 Marks)

1. OOP Concepts (Encapsulation, Inheritance, Polymorphism)	3 marks	- Proper use of private/protected variables (Encapsulation) - Base and derived classes implemented (Inheritance) - Method overriding or overloading shown (Polymorphism)
2. Class Design and Functionality	2 marks	- Logical structure and relationship between classes - Each class has a clear role - Proper use of methods and constructors
Transaction Implementation **User Interaction (Input/Output)**	1.5 marks	- Functional transaction system (e.g., transfer, borrow/return) - Reflects real-world logic and handles edge cases (e.g., insufficient funds, unavailable books) Accepts user input to create objects and perform actions - Provides clear prompts and messages
5. Code Quality and Documentation	1 mark	- Code is readable and well-commented - Uses meaningful variable/class names and consistent indentation
6. Group Presentation & Video	2.5 marks	- Clear explanation of the system, OOP usage, and features - Presentation is well-organized and within time - Video submitted and of good quality