Python_Activity43

👩‍💻 Group Code Activities (Design Principles in Practice)

Each group receives a different real-world system **scenario** and must implement a small modular Python solution using an **assigned design pattern**.

✅ Group 1: Fitness Center Management System

Design Focus: MVC (Model-View-Controller)

Instructions:

- Create a booking system where a **member** can book a **fitness class**.
- Use:
  - A **Model** to represent the member
  - A **Controller** to handle the booking logic
  - A **View** to display confirmation messages

✅ Group 2: Barangay Document Request System

Design Focus: Factory Pattern

Instructions:

- Create a system where residents can request different types of **barangay documents**.
- Use a **Factory** to create the correct document object (e.g., Clearance, Indigency).
- Each document should have a method like `prepare()` to simulate document processing.

✅ Group 3: School Clinic Appointment System

Design Focus: Singleton Pattern

Instructions:

- Create an **Appointment Manager** that only exists once.
- Ensure that all bookings use the same instance of the manager.
- Add a method like `book(name)` and show that all bookings are stored in the same object.

✅ Group 4: Online Book Borrowing System

Design Focus: Decorator Pattern

Instructions:

- Create a basic `Book` class.
- Allow additional features like `WithLateFee` or `WithReminderNotification` to be added without changing the core `Book` class.
- Use **decorators** to add these features dynamically.

✅ Group 5: Student Event Registration System

Design Focus: Observer Pattern

Instructions:

- Create a registration system where students are notified when new events are announced.
- Students must be able to **subscribe** and receive messages when a new event is posted.
- Implement `subscribe()` and `notify()` functionality.