QUIZ- 6

35 points (7 x 5[questions])

Due Date: 02/25

All the programs should contain a main function and please use dummy values. Inside of function/class do not use pass (or …) at least write a print statement.

1. You are working on a codebase for managing online orders. Currently, the Order class has the following functionalities:

    a. Storing order details (customer info, items, shipping address)
    b. Calculating total order cost (including taxes and discounts)
    c. Validating order data (checking item availability, customer address etc.)
    d. Sending order confirmation emails to customers
    e. Updating inventory levels after order processing

    Question: Write a Program (WAP) on how you would refactor the Order class to follow the Single Responsibility Principle (SRP). Name your program **s.py**.
    - Each listed functionality represents a distinct responsibility.
    - SRP suggests separating each responsibility into its own class or module.
    - Think about how different functionalities interact and depend on each other.
    - Consider the impact on code readability, maintainability, and reusability.

2. Imagine you're developing a graphics application that requires calculating the areas of different shapes. Currently, you have a base Shape class with an abstract get_area method that each concrete shape class (e.g., Circle, Square, Rectangle) must implement.

    Question: WAP to design this system to adhere to the Open-Closed Principle (OCP). This means you should be able to add new shapes without modifying existing code. Name your program **o.py**.
3. You're developing a 2D geometry drawing application that uses various shapes. There are currently base classes for Shape and specific subclasses for Circle, Rectangle, and Triangle. Each subclass implements a get_area() method to calculate its respective area.

    Question:

    WAP to ensure compliance with the Liskov Substitution Principle (LSP). Consider the following challenges and Name your program **l.py**.:

    Circle and Rectangle have a set_width() and set_height() method, while Triangle doesn't. How can you handle this difference without violating LSP?
    The get_area() method implementation varies across shapes. Is this a violation of LSP. If so, propose solutions to maintain consistency while accommodating specific area calculations.

Imagine adding a new Polygon shape with multiple sides. How would you integrate it into the hierarchy without compromising LSP principles?

4. Imagine you're developing a library management system with a Library class. Currently, Library offers functionality for:

    - Adding and removing books from the catalog
    - Searching for books by title, author, or genre
    - Borrowing and returning books for registered users
    - Generating reports on borrowings, overdue books, and book popularity

    Question:

    WAP in which Library classes adhere to the Interface Segregation Principle (ISP). Consider the following issues and Name your program **i.py**:

    - Not all users need all functionalities. For example, a guest user might only be interested in searching for books, while a librarian needs broader management features.
    - Large interface can lead to tight coupling and inflexibility. Adding new functionalities can potentially affect all users, even those who don't need them.

5. Imagine you're developing a web application and need a robust logging system. Currently, your code directly utilizes a specific logging library like **logging**, **loguru**, **Google_auth**, throughout your application.
    Question:

    WAP to adhere to the Dependency Inversion Principle (DIP) in regards to logging. Focus on the following aspects and Name your program **d.py**.:

    - Tight coupling with a specific logging library: Can you decouple your application logic from the chosen library, making it easier to switch or extend logging capabilities later?
    - Testing concerns: How can DIP improve the testability of your application's logging functionality?
    - Enhancing flexibility: Can you introduce mechanisms to dynamically configure logging behavior based on different environments or user preferences?

Bonus Question: 25 points out of your minterm. That means if the midterm exam is of 100 points you have to attempt only 75 points. Plus, I will put down your name on the list of students who have attempted bonus questions. It should be a full fledged system with working dummy data and all. I cannot give you all the constraints so assume things on your own. You should assume more rather than less.

Develop a system for a fitness tracker that collects, stores, and displays user activity data (steps, distance, calories burned). This system should allow for easy addition of new activity types (e.g., swimming, cycling) in the future.

Question:

Implement the data collection, storage, and display functionality, focusing on the Observer pattern and applying SOLID principles:

SRP: Create separate classes for User, Activity, ActivityMonitor, DataStorage, and Display.
OCP: Leverage the Observer pattern to notify the Display whenever the ActivityMonitor collects new data about the user's activity. This allows you to add new activity types without modifying existing classes.
LSP: Ensure the Activity class and its subclasses adhere to the observer pattern's contracts, making them compatible with the notification mechanism.
ISP: Define separate interfaces for data collection and display if they have distinct concerns.
DIP: Inject dependencies like the DataStorage and Display into the ActivityMonitor constructor for loose coupling and easier testing.

Key Considerations:

Modularity: Break down the problem into well-defined modules or components.
Cohesion: Each module should focus on a single aspect of the functionality.
Extensibility: Design for future flexibility and growth.
Testability: Make the code easier to test and maintain.
Documentation: Clearly document your design decisions and rationale. This is important, write comments on top of the program explaining how SOLID is satisfied in your program. Write in details.

Push it to your github and and share the link with the TA ONLY. In the email the subject should be "CS325:quiz-6". If bonus attempted "CS325:quiz6_BONUS".

ALSO ZIP the 5-files (plus bonus if attempted) into a single zip file and submit it to Moodle.