National University of Computer and Emerging Sciences

Laboratory manual # 05
For
Software Design and Architecture

Course Instructor	Amir Iqbal
Lab Instructor	Muhammad Hashir Mohsineen, Syeda Aina Batool
Email	hashir.mohsineen@lhr.nu.edu.pk ainnie.batool275@gmail.com
Section	BSE-4D
Date	3-06-24 (MM/DD/YY)
Semester	Spring 24

Instructions for lab submission:
You have to submit source files along with a word document. In the word document you have to give the heading of each exercise/question, then paste your code. Save your word document in the following format: roll number-lab no-section i.e. 21I-0008-lab05-BSE4D.

Objective:

- □ Single Responsibility Principle
- □ Open/Closed Principle

Software for this lab:

- Java netbeans
- (You can also use other Java IDEs i.e. Eclipse)

1. **Exercise: Marks: 10**

Design a Java program that applies the Single Responsibility Principle (SRP) pattern to a simple shopping application. Here's the breakdown of what you need to accomplish:

a. Product Management:
    i. Create a class called **Product** to represent a product available for purchase.
    ii. Implement methods for managing product details such as name, price, and availability.
    iii. Ensure that the Product class has a single responsibility related to managing product information.

b. Cart Management:
    i. Develop a separate class called **ShoppingCart** responsible for managing the user's shopping cart.
    ii. Implement methods for adding and removing products from the cart, calculating the total price, and placing orders.
    iii. Ensure that the ShoppingCart class has a single responsibility related to managing the user's shopping cart.

c. Order Processing:
    i. Create a class called **OrderProcessor** responsible for processing orders placed by users.
    ii. Implement methods for validating orders, updating product availability, and generating order receipts.
    iii. Ensure that the OrderProcessor class has a single responsibility related to processing orders.

2. **Exercise:** **Marks: 10**

Create a Java program for a library management system that adheres to the Open/Closed Principle while ensuring correctness.

Book Management:

- Design an abstract class called **LibraryItem** representing a generic library item.
- Implement subclasses for specific types of library items such as **Book**, **Magazine**, and **DVD**.
- Each subclass should contain attributes relevant to the item (e.g., title, author, publication year) and override a method called displayDetails() to display item details.

Fine Calculation:

- Extend the program to include fine calculation for late returns of library items.
- Introduce a new interface named **FineCalculator** with methods for calculating fines.
- Implement classes such as **StandardFineCalculator** and **ReducedFineCalculator**, each providing a different fine calculation strategy based on item type and duration of late return.

Extension:

- Consider potential future requirements, such as introducing new types of library items or fine calculation strategies.
- Ensure that the program design allows for easy extension without modifying existing code.
- Discuss how the program's correctness is maintained even when extending its functionality.