National University of Computer and Emerging Sciences

Laboratory manual # 11
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	05-08-24 (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-lab11-BSE4D.

Objective:

- ☐ Template Method Design Pattern

Software for this lab:

- Java netbeans
- Star UML

Write code (in java) and make class diagram for the following:

1. **Exercise:** **Marks: 10**

Implement a simple application that calculates the total cost of a customer's order in an online shopping system. Use template method design pattern for the implementation:

a. Create an abstract class named OrderTemplate that defines the skeleton of the algorithm to calculate the total cost of an order.
b. Implement two concrete subclasses of OrderTemplate: OnlineOrder and InStoreOrder. These subclasses will override the necessary methods to provide specific implementation details.
c. In the OrderTemplate class, define the following template methods:
    i. calculateTotalPrice() - This method should calculate the total cost of the order.
    ii. applyDiscount() - This method should apply any discounts applicable to the order.
    iii. addTaxes() - This method should add taxes to the total cost.
d. In the OnlineOrder subclass, implement the calculateTotalPrice() method to include shipping costs and override the applyDiscount() method to provide online-specific discounts.
e. In the InStoreOrder subclass, implement the calculateTotalPrice() method without shipping costs and override the applyDiscount() method to provide in-store-specific discounts.
f. Test your implementation by creating instances of OnlineOrder and InStoreOrder and calculating their total costs.

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

Implement the Template Method design pattern for sorting algorithms. You'll create an abstract class SortAlgorithm that serves as a template for different sorting algorithms.

1. Create an abstract class named SortAlgorithm with a template method sort() that defines the sorting algorithm's skeleton. This method should call two abstract methods:
    - compare() to compare elements during sorting.
    - swap() to swap elements if necessary.
2. Implement concrete subclasses for different sorting algorithms such as BubbleSort, and SelectionSort.
3. Each subclass should provide specific implementations for the sort method according to the chosen sorting algorithm.
4. Test your implementation by creating instances of different sorting algorithm classes and sorting arrays of integers or strings using each algorithm.