National University of Computer & Emerging Sciences, Karachi

Fall-2023, School of Computing (BSCS, BSSE, BSAI)

Assignment # 3

Subject: Object Oriented Programming **Total Marks: 50**	**Post Date: 23rd November 2023** **Due Date: 10th December 2023**
Course Instructor: Ms. Abeeha Sattar, Ms. Bakhtawer	

Instructions to be strictly followed.

---

1. **Each student should submit these files:**
    a. **A zip of all source files named as "A3-Q#[StudentID]" where # is the question number and Student ID is your ID.**
    b. **A DOC file for diagrams, answers, code for each question and screen shot of the output. This document contains all the questions, answer codes and output in sequence. Name this document as "A3-[StudentID].docx".**
    c. **All the submissions will be made on Google Classroom.**
2. **Each output should have STUDENT ID and NAME of the student at the top.**
3. **It should be clear that your assignment would not get any credit if the assignment is submitted after the due date.**
4. **Zero grade for plagiarism (copy/ cheating) and late submissions.**

Task 1

Epic Electronics is a renowned electronics store that specializes in selling various gadgets, including smartphones, laptops, and smartwatches. Apart from individual gadgets, Epic Electronics offers exclusive bundles that comprise a smartphone, laptop, and a smartwatch. Each gadget has a unique serial number.

Assumptions:

- The design for smartphones can either be flagship, mid-range, or budget. The unit price for each of these types is $800, $500, and $200, respectively.
- Smartphones and laptops are taxable items, but smartwatches are not.
- The laptop models available are either ultrabook, gaming, or business. The unit price for a laptop is $1200 for ultrabook, $1500 for gaming, and $1000 for business.
- The smartwatch styles can either be fitness or luxury.
- The price for a fitness smartwatch is $150, and a luxury smartwatch is $500.
- The tax rate for smartphones is 8% of the price, and for laptops, it is 6% of the price.
- You may create or modify any variables and override any functions if needed to satisfy the requirements of the question.

Tasks to be performed:

a. Illustrate how different objects will interact with each other using a partial class diagram.

b. Identify the type(s) of inheritance present in the model and list different classes/interfaces, which are involved in that particular type of inheritance.

c. Declare variables and also provide suitable implementation for default and parameterized constructor(s) of each class.

d. Add a function calculateTotalPrice() in the class **Bundle** that calculates the total price of the bundle based on the items present in the bundle.

e. Overload the function calculateTotalPrice() in **Smartphone** class to accept discount vouchers (a string as a discount code) along with the smartphone object and return the discounted price.

f. For keeping track of the inventory, there must be a mechanism to find out how much of each individual item (smartphone, laptop, or smartwatch) is remaining in stock. Also, provide a mechanism to see the current revenue (overall profit). Keep in mind that each type of gadget has a different price.

h. Provide a copy constructor for copying objects of **Laptop** class.

Task 2

To promote environmental sustainability, a research institute is studying the ecological impact of various urban transportation methods, including electric scooters, hybrid cars, and eco-friendly bicycles. The goal is to evaluate their carbon footprints and contribute data to the ongoing efforts against climate change.

1. Create three small standalone classes:

    - **ElectricScooter**
    - **HybridCar**
    - **EcoFriendlyBicycle**

    Give each class some unique appropriate attributes and behaviors that it does not have in common with other classes. (This is entirely up to you—try to think of at least 3 attributes and behavior)

2. Provide suitable implementation for default and parameterized constructor(s) of each class.

3. Write an abstract class **EcoImpactAnalyzer** with only a public abstract **calculateEcoImpact** method. Have each of your classes inherit from that abstract class and implement the **calculateEcoImpact** method to evaluate an appropriate ecological impact for that class.

4. To calculate the **EcoImpact**, you must take input from the user about the energy source or fuel efficiency. Choose the primary energy source or fuel efficiency for each vehicle:

    - Electric Scooter: Battery charge percentage
    - Hybrid Car: Fuel efficiency in miles per gallon (MPG)
    - Eco-Friendly Bicycle: Human-powered or assisted

    Enter the relevant data for each vehicle type and use it to calculate the **EcoImpact**. The logic for this should be somewhat relevant, but you're free to think of an implementation yourself.

5. Write a program that creates objects of each of the three classes, **polymorphically** invoking each object's **calculateEcoImpact** method.

6. Write the results to the file **EcoImpactResults.txt** for future reference and analysis. (This file should be present in your submission)

Task 3

You are developing a ticket booking system for a theater. The system allows users to book tickets for various shows. However, there are exceptional situations that could occur during the ticket booking process, such as insufficient seats, invalid show selection, or payment errors.

Task: Implement exception handling in the ticket booking system to handle potential errors and ensure proper handling of exceptions.

Assume you have a Java program that handles ticket booking for a theater. Write a code showing how you would handle exceptions during the ticket booking process. Specifically, handle the following exceptions:

InsufficientSeatsException: If the requested number of seats is not available for the selected show.

InvalidShowException: If an invalid show is selected for booking.

PaymentErrorException: If there is any error during the payment process.

To demonstrate the exception, write a basic piece of code with a list/array of shows, and a list of Boolean arrays (**ArrayList< boolean[] >**) showing the availability of seats against each show. Perform different operations on these, and throw/catch the exceptions as needed.

Task 4

Write a C++ program that has a class named **Person**.

- The class has a default constructor that displays “I am a person”.
- The class has attributes name, age, nationality, address and CNIC.
- The class has an input function that prompts the user to enter all the details.
- For CNIC, the total number of digits should be exactly 13. If it’s less than 13 or greater display an error message.
- The class also has a display function that displays all the details.

Derive a class **Employee** from Person.

- The class Employee has a default constructor that invokes the base class’s constructor and displays “I am an employee”.
- The class has the attributes name of company, company’s location (city), no of years worked.
- The class has an input function that prompts the user to enter all the details. It also has a display function that displays all the details.

Derive a class **Manager** from Employee.

- The class Manager has a default constructor that invokes the base class’s constructor and displays “I am a manager”.
- The class has an array that contains the names of employee’s who are working under the manager’s supervision. Input at least five employees in the array from the user and display all these employees too.

In the main program, call all the functions and display the details.

Task 5

Imagine you are working on a complex e-commerce system that processes a large number of transactions daily. To maintain the system's stability and identify issues promptly, you decide to implement an error logging system using object-oriented principles.

Classes:

1. **ECommerceException:**
    - **Responsibility:** An exception class, responsible for using the ErrorLogger class to log errors to file. This exception will be thrown whenever something goes wrong in the e-commerce application. (You don't have to provide implementation for the app!)
    - **Properties:**
        - **errorMessage:** A string that stores information about the error that occurred.
    - **Methods:**
        - **Constructor:** A parameterized constructor that sets the error message as the message provided by the program whenever the exception is thrown.
        - **toString():** Override the toString() method to return the error after formatting it with the LogFormatter class.
2. **ErrorLogger:**
    - **Responsibility:** The main class responsible for error logging.
    - **Properties:**
        - **logFileName**: A string representing the name of the log file.
    - **Methods:**
        - **logError(errorMessage: String)**: Logs the error message along with a timestamp (with the help of the LogFormatter class) to the specified log file.
3. **LogFormatter:**
    - **Responsibility:** Formats the error messages with timestamps.
    - **Methods:**
        - **formatLogMessage(message: String): String**: Takes an error message and returns a formatted message with a timestamp.

In your main function, throw some ECommerceException exceptions in different try blocks, then handle them. In the catch block, utilize the ErrorLogger class to log the errors to a file. The generated file should be a part of the submission.