RAJIV GANDHI UNIVERSITY OF KNOWLEDGE TECHNOLOGIES, BASAR
Basar, Nirmal (Dist), Telangana - 504107

(A.Y. 2023-2024)

Branch: Computer Science and Engineering

R19

Subject Name:	**Object Oriented Programming Lab**	**Subject Code:**	
Date & Class :	**14/12/23- C1**	**Time:**	**180 Minutes**
Exam:	**Lab Internal** **SET NO : II**	**Max. Marks:**	**40M**

Note:

i. Answer any two questions, Each question carries **20 marks.**

1. **Programming Question: Mail-Order House Product Sales and Billing System**

Consider a mail-order house that offers five products with the following details:

Product_ID	Product_Name	Initial_Quantity	Product_Price (in Rs.)
1	Widget A	100	99.90
2	Gizmo B	150	20.20
3	Doodad C	200	6.87
4	Thingamajig D	120	45.50
5	Whatchamacallit E	80	40.49

Each product is identified by its Product_ID and includes details such as Product_Name, Initial_Quantity, and Product_Price.

Develop a Java application that handles a series of product sales by prompting users to input pairs of numbers as follows:

1. Product_ID (1-5) representing the product being sold.
2. Quantity of the product being purchased.

The application should generate a bill for the purchase made by a customer. The customer can purchase any product in any quantity as long as it does not exceed the available quantity, if it exceeds prompt a message that " no products to buy". After each purchase, display the products purchased by the customer along with the bill. Additionally, display the table with the remaining products available in stock.

Ensure that the program provides appropriate prompts for user input and displays the purchased products, bill, and the updated table showing the remaining available products after each purchase.

2. Imagine a scenario where a journey to Warangal is planned, aiming for the minimum possible fuel consumption and the shortest travel time. There are three available cars: `Ford`, `Toyota`, and `Volkswagen`, each possessing unique characteristics.

Detailed information about the cars available for the journey:

	• `Ford`:
	  o Company: Ford
	  o Mileage: (mileage value) km/L
	  o Speed: (speed value) km/h
	  o Color: (color)
	• `Toyota`:
	  o Company: Toyota
	  o Mileage: (mileage value) km/L
	  o Speed: (speed value) km/h
	  o Color: (color)
	• `Volkswagen`:
	  o Company: Volkswagen
	  o Mileage: (mileage value) km/L
	  o Speed: (speed value) km/h
	  o Color: (color)
	
	Develop a Java program that assists in recommending the most suitable car for the journey to Warangal, emphasizing minimal fuel consumption and the least travel time.
	
	Class Name: `Car` Data Fields:
	
	• `company`: String
	• `mileage`: double
	• `speed`: double
	• `color`: String Constructors:
	• `Car()` Methods:
	• `getMileage()`
	• `getSpeed()`
	
	Implement the program logic to compare the mileage and speed attributes of the cars and suggest the optimal car for the journey to Warangal.
	
	Test the program by displaying the recommendation of the most suitable car for the journey based on the specified criteria.

3.	Create an Online Store Inventory Management system in Java using interfaces and OOP principles.
	
	1. Define an interface named `InventoryItem` with methods:
	  o `getItemName()`: Retrieves the name of the inventory item.
	  o `getQuantity()`: Retrieves the quantity of the inventory item available in stock.
	2. Implement classes `ClothingItem` and `ElectronicsItem` that implement the `InventoryItem` interface:
	  o `ClothingItem` and `ElectronicsItem` should contain attributes such as `itemName`, `quantity`, `size` (for ClothingItem), `brand` (for ElectronicsItem), etc.
	  o Implement methods to return the name and quantity of each inventory item.
	3. Create a class named `InventoryManager` with functionality to:
	  o Maintain a list of available inventory items.

	o Add new items to the inventory. o Update the quantity of existing items. o Display the list of items and their quantities in stock. 4. Demonstrate the program by performing inventory management operations: o Add various clothing items and electronics to the inventory. o Update quantities of existing items. o Display the inventory status. Sample output : Shirt added to inventory. Laptop added to inventory. Dress added to inventory. Shirt quantity updated to 60 Inventory Status: Item: Shirt, Quantity: 60 Item: Laptop, Quantity: 20 Item: Dress, Quantity: 30

* * * * *