**1. Product Class**

Attributes: productId, name, category, price, stockQuantity

Getters, setters, constructors

**2. ProductUtility Class**

retrieveProductsByCategory

retrieveProductsByPriceBelow

retrieveLowStockProducts

**3. UserInterface Class**

Input product details

Menu to query inventory

Display results

**Boiler plate**

public class Product {

private String productId;

private String name;

private String category;

private double price;

private int stockQuantity;

public Product() {}

public Product(String productId, String name, String category, double price, int stockQuantity) {

this.productId = productId;

this.name = name;

this.category = category;

this.price = price;

this.stockQuantity = stockQuantity;

}

public String getProductId() {

return productId;

}

public void setProductId(String productId) {

this.productId = productId;

}

public String getName() {

return name;

}

public void setName(String name) {

this.name = name;

}

public String getCategory() {

return category;

}

public void setCategory(String category) {

this.category = category;

}

public double getPrice() {

return price;

}

public void setPrice(double price) {

this.price = price;

}

public int getStockQuantity() {

return stockQuantity;

}

public void setStockQuantity(int stockQuantity) {

this.stockQuantity = stockQuantity;

}

}

import java.util.List;

import java.util.stream.Collectors;

import java.util.stream.Stream;

public class ProductUtility {

public List<Product> retrieveProductsByCategory(Stream<Product> productStream, String category) {

//fill code

}

public List<Product> retrieveProductsByPriceBelow(Stream<Product> productStream, double maxPrice) {

fill the code

}

public List<Product> retrieveLowStockProducts(Stream<Product> productStream, int threshold) {

//fill the cod

}}

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

public class UserInterface {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

List<Product> productList = new ArrayList<>();

ProductUtility utility = new ProductUtility();

System.out.println("Enter the number of products:");

int numProducts = Integer.parseInt(scanner.nextLine());

System.out.println("Enter product details (productId, name, category, price, stockQuantity) separated by commas:");

for (int i = 0; i < numProducts; i++) {

String[] details = scanner.nextLine().split(",");

Product product = new Product(

details[0].trim(),

details[1].trim(),

details[2].trim(),

Double.parseDouble(details[3].trim()),

Integer.parseInt(details[4].trim())

);

productList.add(product);

}

//fill the code

}

**2) order class**

**//boiler palte**

public class Order {

    private String orderId;

    private String customerName;

    private double amountSpent;

    public Order(String orderId, String customerName, double amountSpent) {

        this.orderId = orderId;

        this.customerName = customerName;

        this.amountSpent = amountSpent;

    }

    public String getOrderId() {

        return orderId;

    }

    public String getCustomerName() {

        return customerName;

    }

    public double getAmountSpent() {

        return amountSpent;

    }

    public String toString() {

        return "Order{" +"orderId='" + orderId + '\'' +", customerName='" + customerName + '\'' + ", amountSpent=" + amountSpent +'}';

    }

}

**OrderGroupingExample class**

**Perform following operation**

1)Group orders by customer and calculate the total amount spent by each customer

2)   Group orders by customer and count the number of orders for each customer

3) collect order IDs and their price as Map

class OrderGroupingExample {

    public static void main(String[] args) {

        List<Order> orders = Arrays.asList(

            new Order("O001", "ABC", 150.75),

            new Order("O002", "PQR", 200.50),

            new Order("O003", "ABC", 99.99),

            new Order("O004", "XYZ", 125.00),

            new Order("O005", "PQR", 75.25),

            new Order("O006", "XYZ", 180.00)

        );

//fill the code

}

}