**Data Structures And Design Patterns(Hands-on)**

**Exercise 1: Inventory Management System**

**Product.java**

package inventory;

public class Product {

private String productId;

private String productName;

private int quantity;

private double price;

public Product(String productId, String productName, int quantity, double price) {

this.productId = productId;

this.productName = productName;

this.quantity = quantity;

this.price = price;

}

public String getProductId() { return productId; }

public String getProductName() { return productName; }

public int getQuantity() { return quantity; }

public double getPrice() { return price; }

public void setQuantity(int quantity) { this.quantity = quantity; }

public void setPrice(double price) { this.price = price; }

@Override

public String toString() {

return "[" + productId + "] " + productName + " - Qty: " + quantity + ", Price: ₹" + price;

}

}

**InventoryManager.java**

package inventory;

import java.util.HashMap;

public class InventoryManager {

private HashMap<String, Product> inventory = new HashMap<>();

public void addProduct(Product product) {

inventory.put(product.getProductId(), product);

System.out.println("Product added: " + product);

}

public void updateProduct(String productId, int quantity, double price) {

Product product = inventory.get(productId);

if (product != null) {

product.setQuantity(quantity);

product.setPrice(price);

System.out.println("Product updated: " + product);

} else {

System.out.println("Product not found!");

}

}

public void deleteProduct(String productId) {

Product removed = inventory.remove(productId);

if (removed != null) {

System.out.println("Product deleted: " + removed);

} else {

System.out.println("Product not found!");

}

}

// View all products

public void showInventory() {

if (inventory.isEmpty()) {

System.out.println("Inventory is empty.");

return;

}

System.out.println("Current Inventory:");

for (Product p : inventory.values()) {

System.out.println(p);

}

}

}

**Main.java**

package inventory;

public class Main {

public static void main(String[] args) {

InventoryManager manager = new InventoryManager();

manager.addProduct(new Product("P001", "Laptop", 10, 50000));

manager.addProduct(new Product("P002", "Smartphone", 20, 15000));

manager.showInventory();

manager.updateProduct("P001", 8, 48000);

manager.deleteProduct("P002");

manager.showInventory();

}

}

A computer screen with white text

AI-generated content may be incorrect.

**Exercise 7: Financial Forecasting**

**Forecaster.java**

package forecast;

public class Forecaster {

public double forecastValue(double currentValue, double growthRate, int years) {

if (years == 0) {

return currentValue; // base case

}

double nextYearValue = forecastValue(currentValue, growthRate, years - 1);

return nextYearValue \* (1 + growthRate);

}

}

**Main.java**

package forecast;

public class Main {

public static void main(String[] args) {

Forecaster forecaster = new Forecaster();

double currentValue = 10000;

double growthRate = 0.08;

int years = 5;

double result = forecaster.forecastValue(currentValue, growthRate, years);

System.out.printf("Forecasted value after %d years: ₹%.2f\n", years, result);

}

}

A screen shot of a computer

AI-generated content may be incorrect.