**WEEK 1 : DATA STRUCTURES AND ALGORITHMS**

**EXERCISE 2: E-COMMERCE PLATFORM SEARCH FUNCTION**

Product.java:

package com.ecommerce.search;

public class Product {

int productId;

String productName;

String category;

public Product(int productId, String productName, String category) {

this.productId = productId;

this.productName = productName;

this.category = category;

}

public int getProductId() {

return productId;

}

public String getProductName() {

return productName;

}

public String getCategory() {

return category;

}

*@Override*

public String toString() {

return "Product ID: " + productId +

", Name: " + productName +

", Category: " + category;

}

}

**SearchUtils.java:**

package com.ecommerce.search;

import java.util.ArrayList;

import java.util.List;

public class SearchUtils {

public static Product linearSearchById(Product[] products, int targetId) {

for (Product product : products) {

if (product.getProductId() == targetId) {

return product;

}

}

return null;

}

public static List<Product> searchByName(Product[] products, String name) {

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

for (Product product : products) {

if (product.getProductName().equalsIgnoreCase(name)) {

matched.add(product);

}

}

return matched;

}

public static List<Product> searchByCategory(Product[] products, String category) {

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

for (Product product : products) {

if (product.getCategory().equalsIgnoreCase(category)) {

matched.add(product);

}

}

return matched;

}

}

**Main.java:**

package com.ecommerce.search;

import java.util.Scanner;

import java.util.List;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

Product[] products = {

new Product(105, "Laptop", "Electronics"),

new Product(102, "Shoes", "Fashion"),

new Product(101, "Book", "Stationery"),

new Product(104, "Mobile", "Electronics"),

new Product(103, "T-shirt", "Fashion"),

new Product(106, "Smartwatch", "Electronics")

};

System.out.println("==== E-Commerce Product Search ====");

System.out.println("Choose Search Criteria:");

System.out.println("1. Search by Product ID");

System.out.println("2. Search by Product Name");

System.out.println("3. Search by Category");

System.out.print("Enter choice: ");

int choice = scanner.nextInt();

scanner.nextLine(); // consume newline

switch (choice) {

case 1:

System.out.print("Enter Product ID to search: ");

int id = scanner.nextInt();

Product result = SearchUtils.linearSearchById(products, id);

if (result != null) {

System.out.println("Product Found: " + result);

} else {

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

}

break;

case 2:

System.out.print("Enter Product Name to search: ");

String name = scanner.nextLine();

List<Product> byName = SearchUtils.searchByName(products, name);

if (!byName.isEmpty()) {

System.out.println("Matching Products:");

byName.forEach(System.out::println);

} else {

System.out.println("No products found with name: " + name);

}

break;

case 3:

System.out.print("Enter Category to search: ");

String category = scanner.nextLine();

List<Product> byCategory = SearchUtils.searchByCategory(products, category);

if (!byCategory.isEmpty()) {

System.out.println("Matching Products:");

byCategory.forEach(System.out::println);

} else {

System.out.println("No products found in category: " + category);

}

break;

default:

System.out.println("Invalid choice.");

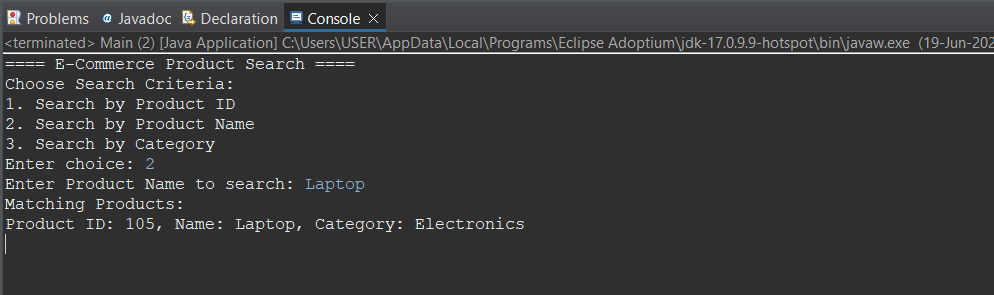
}

scanner.close();

}

}

**OUTPUT:**



**EXERCISE 7: FINANCIAL FORECASTING**

RevenuePredictor.java:

package com.company.forecast;

public class RevenuePredictor {

// Recursive method to calculate future revenue

public static double predictRevenue(double currentRevenue, double growthRate, int years) {

if (years == 0) {

return currentRevenue;

}

return *predictRevenue*(currentRevenue, growthRate, years - 1) \* (1 + growthRate);

}

// Optional: Iterative optimization to avoid recursion for large values

public static double predictRevenueIterative(double currentRevenue, double growthRate, int years) {

double revenue = currentRevenue;

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

revenue \*= (1 + growthRate);

}

return revenue;

}

}

Main.java:

package com.company.forecast;

import java.util.Scanner;

public class Main {

public static void main(String[] args) {

Scanner scanner = new Scanner(System.***in***);

System.***out***.println("==== Company Revenue Forecasting Tool ====");

System.***out***.print("Enter current annual revenue (in ₹): ");

double currentRevenue = scanner.nextDouble();

System.***out***.print("Enter annual growth rate (in %, e.g., 10 for 10%): ");

double growthRate = scanner.nextDouble() / 100;

System.***out***.print("Enter number of years to forecast: ");

int years = scanner.nextInt();

System.***out***.println("\n--- Forecasting using Recursive Algorithm ---");

double futureRevenue = RevenuePredictor.*predictRevenue*(currentRevenue, growthRate, years);

System.***out***.printf("Predicted revenue after %d years: ₹%.2f\n", years, futureRevenue);

System.***out***.println("\n--- Forecasting using Iterative Algorithm (Optimized) ---");

double optimizedRevenue = RevenuePredictor.*predictRevenueIterative*(currentRevenue, growthRate, years);

System.***out***.printf("Predicted revenue after %d years (optimized): ₹%.2f\n", years, optimizedRevenue);

scanner.close();

}

}

**OUTPUT:**

