Week – 1 - Engineering concepts:

Design Patterns and Principles:

1.Implementing the Singleton Pattern:

Singleton.java

class Singleton {

private static Singleton instance;

private Singleton() {

System.out.println("Singleton Instance Created");

}

public static Singleton getInstance() {

if (instance == null) {

instance = new Singleton();

}

return instance;

}

public void displayMessage() {

System.out.println("Hello! I am a Singleton object.");

}

}

SingletonDemo.java

public class SingletonDemo {

public static void main(String[] args) {

Singleton obj1 = Singleton.getInstance();

obj1.displayMessage();

Singleton obj2 = Singleton.getInstance();

obj2.displayMessage();

if (obj1 == obj2) {

System.out.println("Both objects are the same (singleton confirmed).");

} else {

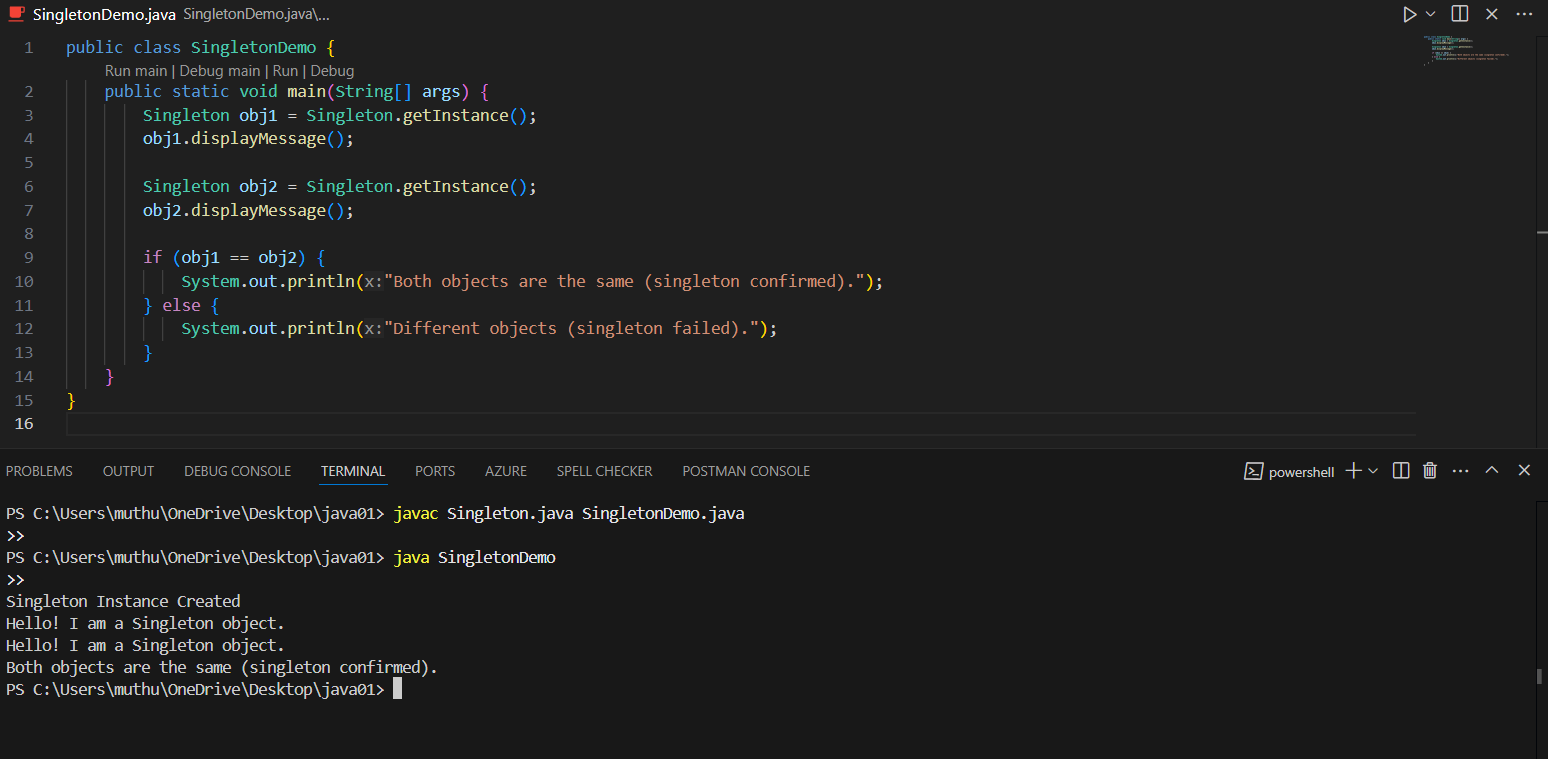
System.out.println("Different objects (singleton failed).");

}

}

}

Output:



2. Implementing the Factory Method Pattern:

FactoryDemo.java

interface Shape {

void draw();

}

class Circle implements Shape {

public void draw() {

System.out.println("Drawing a Circle");

}

}

class Rectangle implements Shape {

public void draw() {

System.out.println("Drawing a Rectangle");

}

}

class ShapeFactory {

public Shape getShape(String shapeType) {

if (shapeType == null) {

return null;

}

if (shapeType.equalsIgnoreCase("CIRCLE")) {

return new Circle();

} else if (shapeType.equalsIgnoreCase("RECTANGLE")) {

return new Rectangle();

}

return null;

}

}

public class FactoryDemo {

public static void main(String[] args) {

ShapeFactory shapeFactory = new ShapeFactory();

Shape shape1 = shapeFactory.getShape("CIRCLE");

shape1.draw();

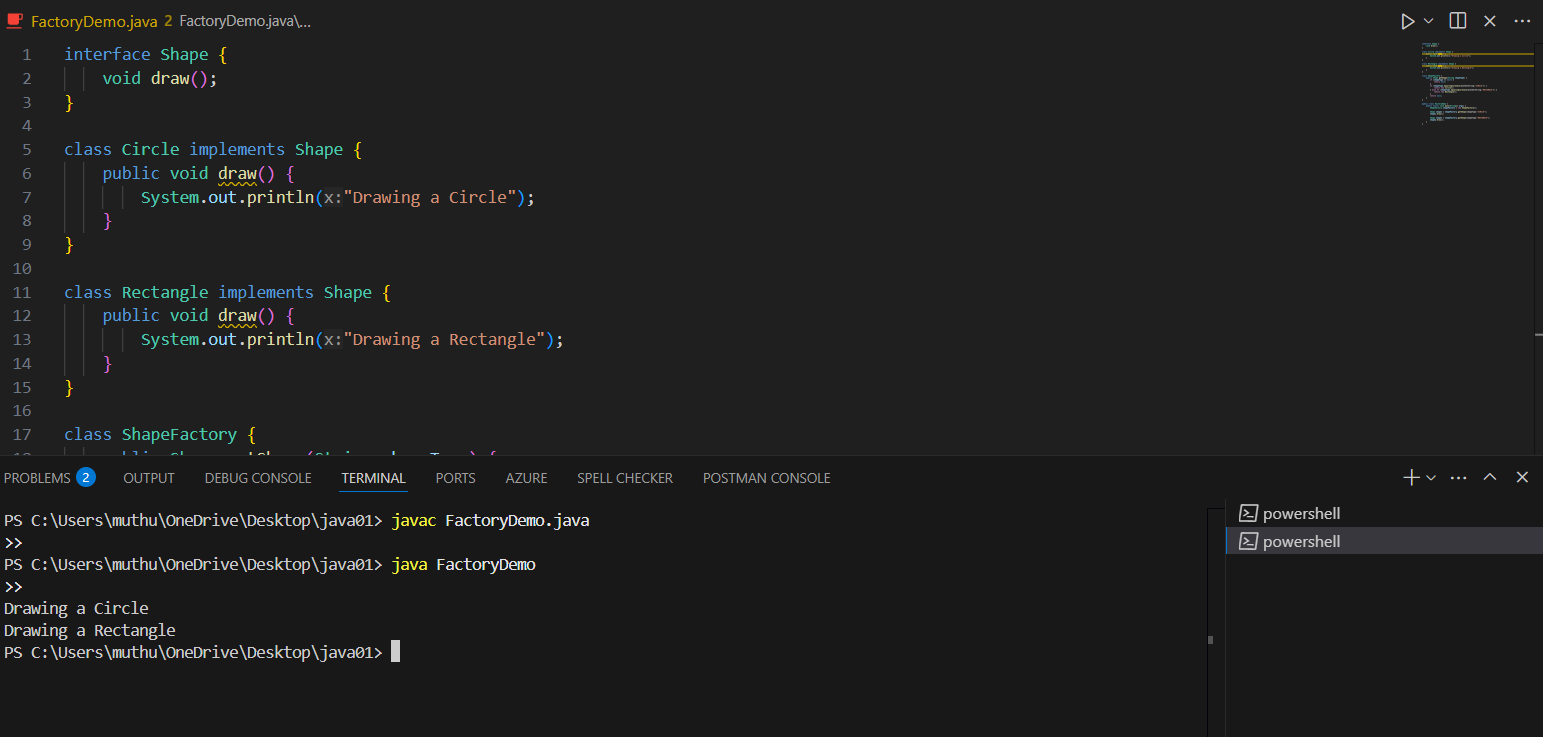
Shape shape2 = shapeFactory.getShape("RECTANGLE");

shape2.draw();

}

}

Output:



Data Structures and Algorithms:

1.E-commerce Platform Search Function:

ECommerceSearch.java

import java.util.ArrayList;

import java.util.List;

class Product {

private String name;

private String category;

public Product(String name, String category) {

this.name = name;

this.category = category;

}

public String getName() {

return name;

}

public void display() {

System.out.println("Product: " + name + " | Category: " + category);

}

}

class SearchService {

private List<Product> products;

public SearchService() {

products = new ArrayList<>();

products.add(new Product("Laptop", "Electronics"));

products.add(new Product("Smartphone", "Electronics"));

products.add(new Product("T-Shirt", "Clothing"));

products.add(new Product("Shoes", "Footwear"));

products.add(new Product("Laptop Bag", "Accessories"));

}

public void search(String keyword) {

System.out.println("Search results for: \"" + keyword + "\"");

boolean found = false;

for (Product p : products) {

if (p.getName().toLowerCase().contains(keyword.toLowerCase())) {

p.display();

found = true;

}

}

if (!found) {

System.out.println("No matching products found.");

}

System.out.println(); // just for clean spacing

}

}

public class ECommerceSearch {

public static void main(String[] args) {

SearchService searchService = new SearchService();

searchService.search("laptop");

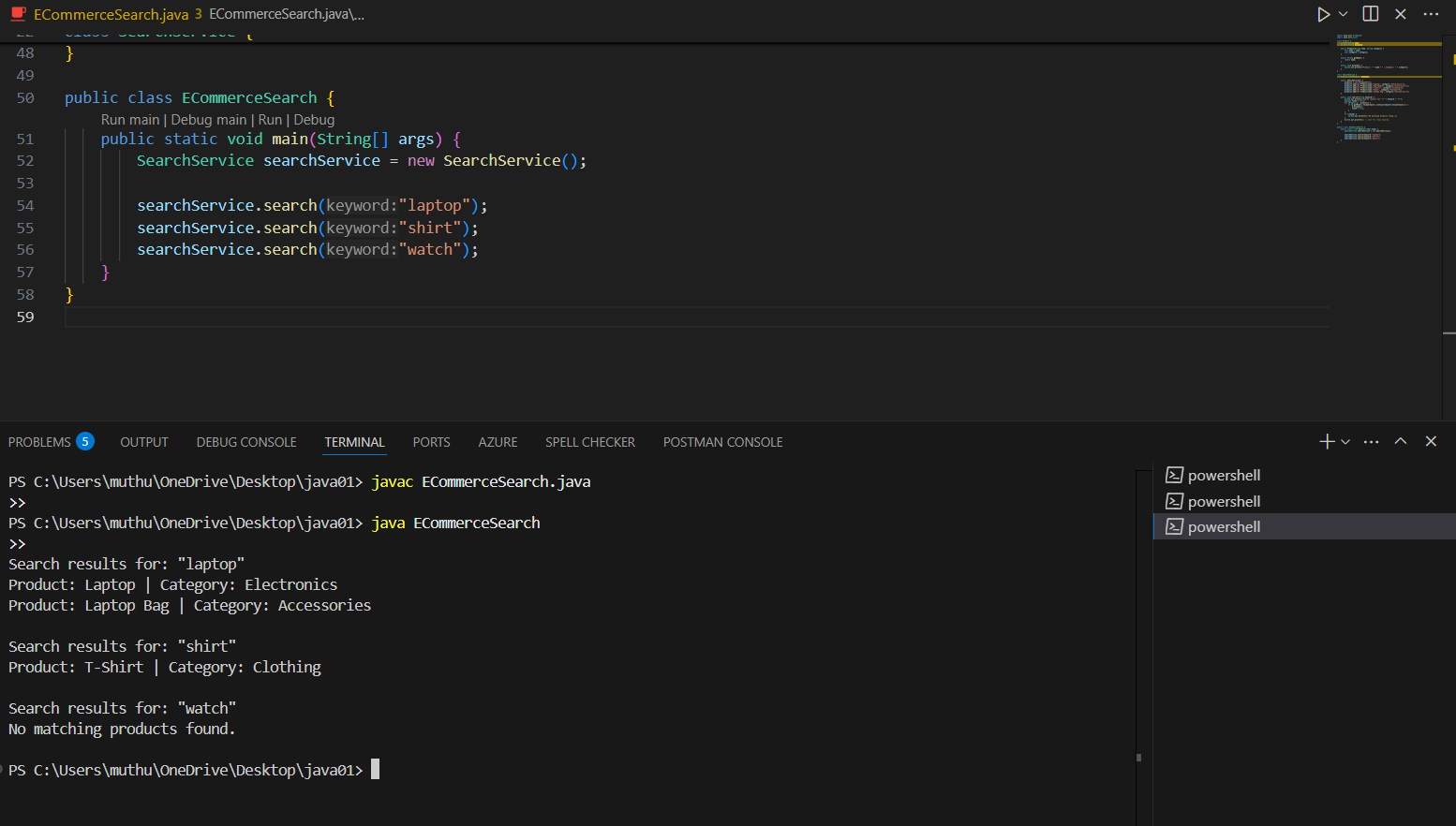
searchService.search("shirt");

searchService.search("watch");

}

}

Output:



2. Financial Forecasting

FinancialForecast.java

public class FinancialForecast {

public static void main(String[] args) {

double initialAmount = 10000.0; // starting amount in ₹

double monthlyGrowthRate = 0.05; // 5% growth per month

int months = 12;

System.out.println("📊 Financial Forecast for 12 Months");

System.out.println("----------------------------------");

double amount = initialAmount;

for (int i = 1; i <= months; i++) {

amount += amount \* monthlyGrowthRate;

System.out.printf("Month %2d: ₹%.2f%n", i, amount);

}

System.out.println("----------------------------------");

System.out.printf("Projected Final Amount: ₹%.2f%n", amount);

}

}

Output:

