Cognizant - DN 4.0 I Deep Skilling

Week-1

Singleton Pattern

Code:-

Public class Singleton{

private static Singleton instance;

private Singleton(){

System.out.println(“Singleton”);

}

Public static Singleton Instance(){

If(instance == null){

Instance=new Singleton();

}

}

}

}

Factory Method Pattern

Code:-

interface Animal {

void sound();

}

class Dog implements Animal {

public void sound() {

System.out.println("Woof!");

}

}

class Cat implements Animal {

public void sound() {

System.out.println("Meow!");

}

}

abstract class AnimalFactory {

public abstract Animal createAnimal();

}

class DogFactory extends AnimalFactory {

public Animal createAnimal() {

return new Dog();

}

}

class CatFactory extends AnimalFactory {

public Animal createAnimal() {

return new Cat();

}

}

public class Main {

public static void main(String[] args) {

AnimalFactory dogFactory = new DogFactory();

Animal dog = dogFactory.createAnimal();

dog.sound();

AnimalFactory catFactory = new CatFactory();

Animal cat = catFactory.createAnimal();

cat.sound();

}

}

E-commerce Platform Search Function

Code:-

import java.util.ArrayList;

import java.util.List;

import java.util.Scanner;

class Product {

    private String name;

    private String category;

    private double price;

    public Product(String name, String category, double price) {

        this.name = name;

        this.category = category;

        this.price = price;

    }

    public String getName() {

        return name;

    }

    public String getCategory() {

        return category;

    }

    public double getPrice() {

        return price;

    }

    public void display() {

        System.out.println("Product: " + name + ", Category: " + category + ", Price: ₹" + price);

    }

}

class ECommercePlatform {

    private List<Product> products;

    public ECommercePlatform() {

        products = new ArrayList<>();

        products.add(new Product("Samsung", "Electronics", 84999));

        products.add(new Product("Apple", "Electronics", 129999));

        products.add(new Product("Nike", "Footwear", 7499));

        products.add(new Product("Denim Jacket", "Clothing", 1899));

        products.add(new Product("Refrigerator", "Appliances", 45999));

    }

    public List<Product> search(String keyword) {

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

        for (Product product : products) {

            if (product.getName().toLowerCase().contains(keyword.toLowerCase()) ||

                product.getCategory().toLowerCase().contains(keyword.toLowerCase())) {

                results.add(product);

            }

        }

        return results;

    }

}

public class SearchFunctionDemo {

    public static void main(String[] args) {

        ECommercePlatform platform = new ECommercePlatform();

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter keyword to search for products: ");

        String keyword = scanner.nextLine();

        List<Product> searchResults = platform.search(keyword);

        if (searchResults.isEmpty()) {

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

        } else {

            System.out.println("Search Results:");

            for (Product product : searchResults) {

                product.display();

            }

        }

    }

}

Financial Forecasting

Code:-

import java.util.Scanner;

public class FinancialForecasting {

    public static double mean(double[] data) {

        double sum = 0;

        for (double val : data) {

            sum += val;

        }

        return sum / data.length;

    }

    public static double[] linearRegression(double[] x, double[] y) {

        int n = x.length;

        double xMean = mean(x);

        double yMean = mean(y);

        double numerator = 0;

        double denominator = 0;

        for (int i = 0; i < n; i++) {

            numerator += (x[i] - xMean) \* (y[i] - yMean);

            denominator += (x[i] - xMean) \* (x[i] - xMean);

        }

        double b = numerator / denominator;

        double a = yMean - b \* xMean;

        return new double[]{a, b};

    }

    public static double forecast(double a, double b, double futureX) {

        return a + b \* futureX;

    }

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter number of past months of data: ");

        int n = scanner.nextInt();

        double[] months = new double[n];

        double[] income = new double[n];

        System.out.println("Enter income data for each month:");

        for (int i = 0; i < n; i++) {

            months[i] = i + 1;

            System.out.print("Month " + (i + 1) + ": ");

            income[i] = scanner.nextDouble();

        }

        double[] coefficients = linearRegression(months, income);

        double a = coefficients[0];

        double b = coefficients[1];

        System.out.print("Enter future month number to predict income (e.g., 7): ");

        double futureMonth = scanner.nextDouble();

        double predictedIncome = forecast(a, b, futureMonth);

        System.out.printf("Predicted income for month %.0f is ₹%.2f\n", futureMonth, predictedIncome);

    }

}