**Exercise 2: E-commerce Platform Search Function**

**Product.java**

public class Product {

    int productId;

    String productName;

    String category;

    public Product(int id, String name, String cat) {

        this.productId = id;

        this.productName = name;

        this.category = cat;

    }

    public String toString() {

        return "Product ID: " + productId + ", Name: " + productName + ", Category: " + category;

    }

}

**SearchUtil.java**

import java.util.Arrays;

import java.util.Comparator;

public class SearchUtil {

    public static Product linearSearch(Product[] products, String name) {

        for (Product p : products) {

            if (p.productName.equalsIgnoreCase(name)) {

                return p;

            }

        }

        return null;

    }

    public static Product binarySearch(Product[] products, String name) {

        Arrays.sort(products, Comparator.comparing(p -> p.productName.toLowerCase()));

        int left = 0, right = products.length - 1;

        while (left <= right) {

            int mid = (left + right) / 2;

            int cmp = products[mid].productName.compareToIgnoreCase(name);

            if (cmp == 0) return products[mid];

            else if (cmp < 0) left = mid + 1;

            else right = mid - 1;

        }

        return null;

    }

}

**SearchTest.java**

public class SearchTest {

    public static void main(String[] args) {

        Product[] products = {

            new Product(1, "Laptop", "Electronics"),

            new Product(2, "Shirt", "Clothing"),

            new Product(3, "Mobile", "Electronics"),

            new Product(4, "Shoes", "Footwear"),

            new Product(5, "Book", "Stationery")

        };

        Product result1 = SearchUtil.linearSearch(products, "Shoes");

        System.out.println("Linear Search Result: " + (result1 != null ? result1 : "Not Found"));

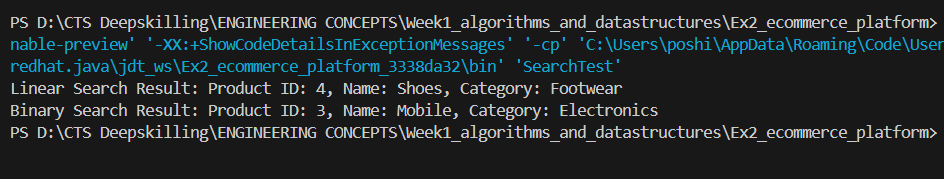
        Product result2 = SearchUtil.binarySearch(products, "Mobile");

        System.out.println("Binary Search Result: " + (result2 != null ? result2 : "Not Found"));

    }

}

**Output:**



**Exercise 7: Financial Forecasting**

**ForecastCalculator.java**

public class ForecastCalculator {

    public static double predictFutureValue(double currentValue, double growthRate, int years) {

        if (years == 0) {

            return currentValue;

        }

        return predictFutureValue(currentValue \* (1 + growthRate / 100), growthRate, years - 1);

    }

}

**ForecastApp.java**

import java.util.Scanner;

public class ForecastApp {

    public static void main(String[] args) {

        Scanner scanner = new Scanner(System.in);

        System.out.println(" Welcome to the Financial Forecast Tool!");

        System.out.print("Enter current value (in rupees): ");

        double currentValue = scanner.nextDouble();

        System.out.print("Enter annual growth rate (%): ");

        double growthRate = scanner.nextDouble();

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

        int years = scanner.nextInt();

        double predictedValue = ForecastCalculator.predictFutureValue(currentValue, growthRate, years);

        System.out.printf(" Predicted value after %d years: %.2f\n", years, predictedValue);

        scanner.close();

    }

}

**Output:**

