**WEEK 1**

**DATA STRUCTURES AND ALGORITHMS**

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

**Scenario:**

You are developing an inventory management system for a warehouse. Efficient data storage and retrieval are crucial.

**Program**

**Product.java**

public class Product {

String productId;

String productName;

String category;

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

this.productId = productId;

this.productName = productName;

this.category = category;

}

@Override

public String toString() {

return productId + " - " + productName + " (" + category + ")";

}

}

**SearchEngine.java**

import java.util.Arrays;

import java.util.Comparator;

public class SearchEngine {

// Linear Search by product name

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

for (Product p : products) {

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

return p;

}

}

return null;

}

// Binary Search by product name

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

// Sort by productName before binary search

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

int low = 0, high = products.length - 1;

while (low <= high) {

int mid = (low + high) / 2;

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

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

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

else high = mid – 1; }

return null;

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

Product[] products = {

new Product("P001", "Mouse", "Electronics"),

new Product("P002", "Keyboard", "Electronics"),

new Product("P003", "Shoes", "Footwear"),

new Product("P004", "T-Shirt", "Clothing")

};

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

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

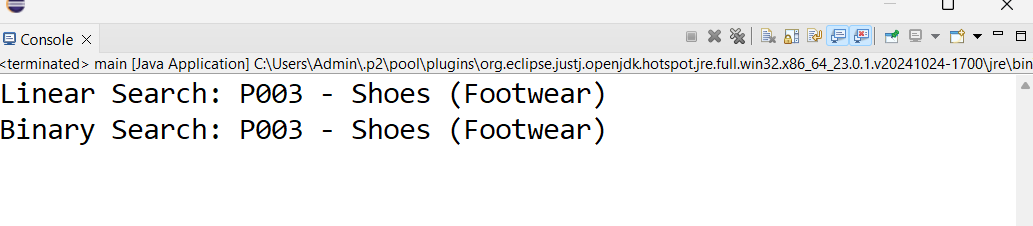
Product result2 = SearchEngine.binarySearch(products, "Shoes");

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

}

}

**Output**

****

**Exercise 7: Financial Forecasting**

**Scenario:**

You are developing a financial forecasting tool that predicts future values based on past data.

**Program**

public class Forecast {

public static double futureValue(double initial, double rate, int years) {

if (years == 0) {

return initial;

}

return futureValue(initial, rate, years - 1) \* (1 + rate);

}

}

**Main.java**

public class Main {

public static void main(String[] args) {

double initialValue = 10000; // Starting value

double annualGrowthRate = 0.10; // 10% growth

int years = 5;

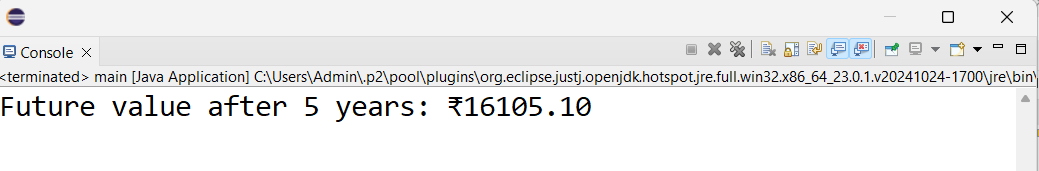
double result = Forecast.futureValue(initialValue, annualGrowthRate, years);

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

}

}

**Output**

****