**ALGORITHMS\_DATA STRUCTURES**

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

**Code:**

import java.util.Arrays;

import java.util.Comparator;

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 String toString() {

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

}

}

class Search {

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

for (Product product : products) {

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

return product;

}

}

return null;

}

public static void sortProductsByName(Product[] products) {

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

}

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

int left = 0;

int 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;

}

}

public class Main {

public static void main(String[] args) {

Product[] products = {

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

new Product(2, "T-shirt", "Apparel"),

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

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

};

Product found = Search.linearSearch(products, "Shoes");

System.out.println("Linear Search Result: " + found);

Search.sortProductsByName(products);

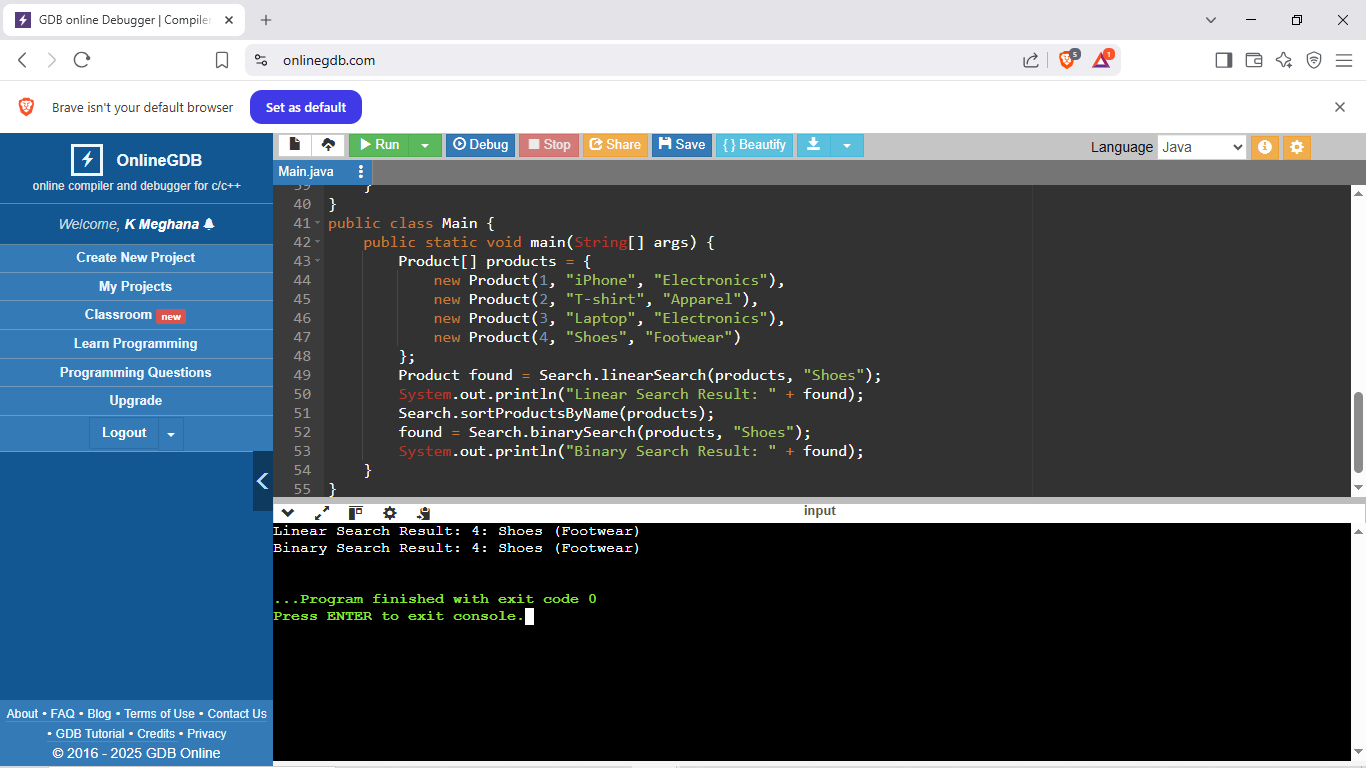
found = Search.binarySearch(products, "Shoes");

System.out.println("Binary Search Result: " + found);

}

}

**Output:**



**Exercise 7: Financial Forecasting**

**Code:**

public class Main {

public static double forecastValue(double presentValue, double rate, int years) {

if (years == 0) {

return presentValue;

} else {

return forecastValue(presentValue, rate, years - 1) \* (1 + rate);

}

}

public static void main(String[] args) {

double presentValue = 10000;

double growthRate = 0.05;

int years = 5;

double futureValue = forecastValue(presentValue, growthRate, years);

System.out.printf("Future Value after %d years = ₹%.2f\n", years, futureValue);

}

}

**Output:** 