# Week 1 - Data Structures and Algorithms - Hands-On

## Exercise 2: E-commerce Platform Search Function

Code Snippets:

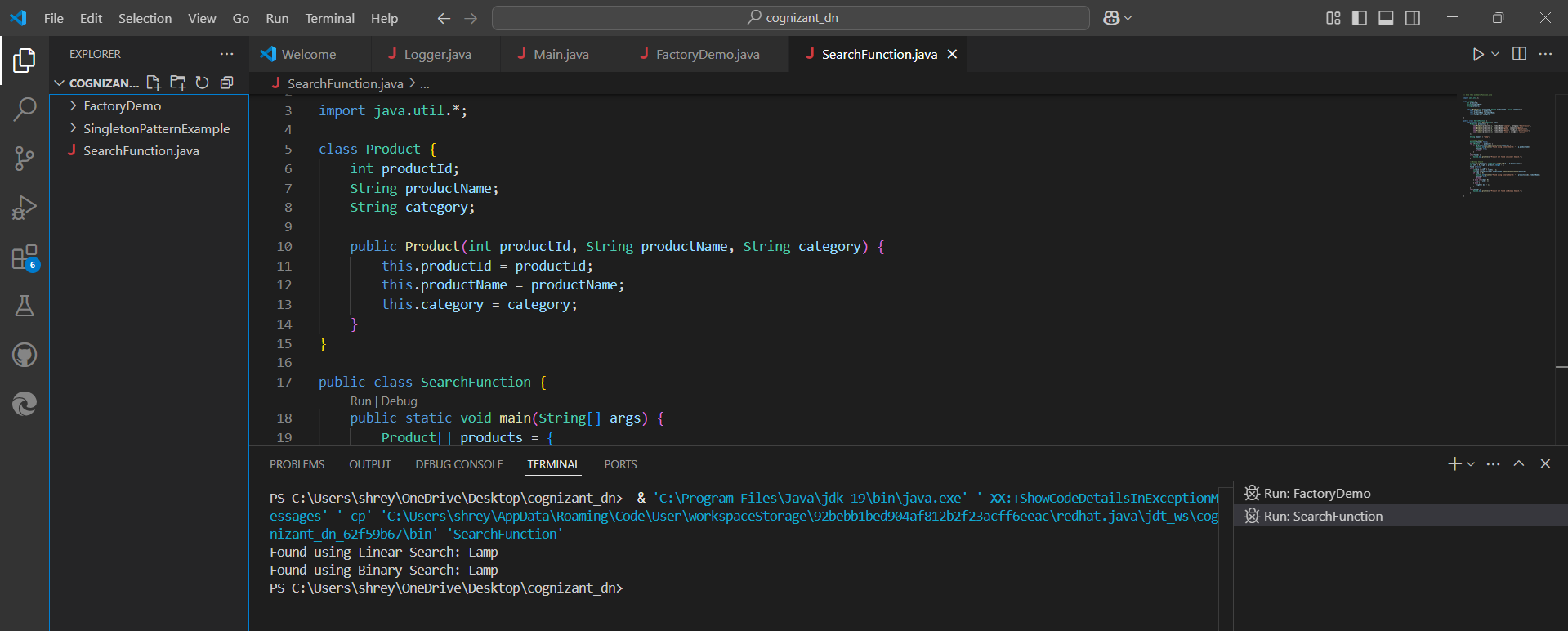
Product.java:

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;  
 }  
}

SearchFunction.java:

import java.util.\*;  
  
public class SearchFunction {  
 public static void main(String[] args) {  
 Product[] products = {  
 new Product(1, "Laptop", "Electronics"),  
 new Product(2, "Lamp", "Home"),  
 new Product(3, "Book", "Education"),  
 new Product(4, "Phone", "Electronics"),  
 new Product(5, "Shoes", "Fashion")  
 };  
  
 String keyword = "Lamp";  
  
 // Linear Search  
 boolean found = false;  
 for (Product p : products) {  
 if (p.productName.equalsIgnoreCase(keyword)) {  
 System.out.println("Found using Linear Search: " + p.productName);  
 found = true;  
 break;  
 }  
 }  
 if (!found) {  
 System.out.println("Product not found in Linear Search.");  
 }  
  
 // Binary Search (sorted by productName)  
 Arrays.sort(products, Comparator.comparing(p -> p.productName));  
 int left = 0, right = products.length - 1;  
 found = false;  
 while (left <= right) {  
 int mid = (left + right) / 2;  
 int cmp = products[mid].productName.compareToIgnoreCase(keyword);  
 if (cmp == 0) {  
 System.out.println("Found using Binary Search: " + products[mid].productName);  
 found = true;  
 break;  
 } else if (cmp < 0) {  
 left = mid + 1;  
 } else {  
 right = mid - 1;  
 }  
 }  
 if (!found) {  
 System.out.println("Product not found in Binary Search.");  
 }  
 }  
}

Output Screenshot:



# Week 1 - Data Structures and Algorithms - Hands-On

## Exercise 7: Financial Forecasting

Sample Code:

public class FinancialForecast {  
  
 // Recursive method to calculate future value  
 public static double predictValue(int years, double initialValue, double growthRate) {  
 if (years == 0) {  
 return initialValue;  
 }  
 return predictValue(years - 1, initialValue, growthRate) \* (1 + growthRate / 100);  
 }  
  
 public static void main(String[] args) {  
 int years = 5;  
 double initialValue = 10000;  
 double growthRate = 7.5;  
  
 double futureValue = predictValue(years, initialValue, growthRate);  
 System.out.printf("Predicted value after %d years: ₹%.2f\n", years, futureValue);  
 }  
}

Output Screenshot:

