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

**Scenario:**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

**Code:  
  
Product.java**

package ecommerceSearch;

public 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;

}

@Override

public String toString() {

return "[" + productId + ", " + productName + ", " + category + "]";

}

}

**SearchDemo.java**

package ecommerceSearch;

import java.util.Arrays;

import java.util.Comparator;

public class SearchDemo {

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

for (Product product : products) {

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

return product;

}

}

return null;

}

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

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

if (result == 0) {

return products[mid];

} else if (result > 0) {

low = mid + 1;

} else {

high = mid - 1;

}

}

return null;

}

public static void main(String[] args) {

Product[] products = {

new Product(101, "Laptop", "Electronics"),

new Product(102, "Shoes", "Footwear"),

new Product(103, "Phone", "Electronics"),

new Product(104, "Watch", "Accessories"),

new Product(105, "Bag", "Bags")

};

String searchKey = "Phone";

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

Product result1 = linearSearch(products, searchKey);

System.out.println(result1 != null ? "Found: " + result1 : "Product not found.");

System.out.println("\n Binary Search:");

Product result2 = binarySearch(products, searchKey);

System.out.println(result2 != null ? "Found: " + result2 : "Product not found.");

}

}

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**Exercise 7: Financial Forecasting**

**Scenario:**

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

**Code:**

package financialForecast;

public class FinancialForecast {

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

if (years == 0) {

return presentValue;

}

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

}

public static void main(String[] args) {

double presentValue = 10000.0; // Initial investment

double growthRate = 0.08; // 8% annual growth

int years = 5; // Forecast for 5 years

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

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

}

}

**Output:**  
A screenshot of a computer

AI-generated content may be incorrect.