**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 com.cognizant.dataStructures;

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;

}

public String toString() {

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

}

}

**SearchingProduct.java**

package com.cognizant.dataStructures;

import java.util.Arrays;

import java.util.Comparator;

public class SearchingProducts {

public static Product linearSearch(Product[] products, int targetId) {

for (Product p : products) {

if (p.productId == targetId) {

return p;

}

}

return null;

}

public static Product binarySearch(Product[] products, int targetId) {

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

while (left <= right) {

int mid = left + (right - left) / 2;

if (products[mid].productId == targetId)

return products[mid];

else if (products[mid].productId < targetId)

left = mid + 1;

else

right = mid - 1;

}

return null;

}

public static void main(String[] args) {

Product[] productList = {

new Product(101, "Shoes", "Footwear"),

new Product(203, "Laptop", "Electronics"),

new Product(150, "T-shirt", "Clothing"),

new Product(120, "Phone", "Electronics"),

new Product(180, "Book", "Stationery")

};

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

Product foundLinear = *linearSearch*(productList, 150);

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

Arrays.*sort*(productList, Comparator.*comparingInt*(p -> p.productId));

System.*out*.println("\nUsing Binary Search (after sorting by productId):");

Product foundBinary = *binarySearch*(productList, 150);

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

}

}

A computer screen shot of a program

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 com.cognizant.dataStructures;

public class FinancialForecast {

public static double calculateFutureValue(double amount, double rate, int years) {

if (years == 0) {

return amount;

}

return *calculateFutureValue*(amount \* (1 + rate), rate, years - 1);

}

public static void main(String[] args) {

double initialAmount = 10000.0;

double annualGrowthRate = 0.05;

int years = 5;

int years10 = 10;

double futureValue = *calculateFutureValue*(initialAmount, annualGrowthRate, years);

double futureValue10 = *calculateFutureValue*(initialAmount, annualGrowthRate, years10);

String fvFormatted = String.*format*("%.2f", futureValue);

String fv10Formatted = String.*format*("%.2f", futureValue10);

System.***out***.println("Future value after " + years + " years: ₹" + fvFormatted);

System.***out***.println("Future value after " + years10 + " years: ₹" + fv10Formatted); }

}

A computer screen shot of a program

AI-generated content may be incorrect.