# E-commerce Platform Search Function :

SearchFunction.java:

import java.util.\*;

class Product {

int productId;

String productName;

String category;

Product(int id, String name, String cat) {

this.productId = id;

this.productName = name;

this.category = cat;

}

public String toString() {

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

}

}

public class SearchFunction {

public static void main(String[] args) {

Product[] products = {

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

new Product(205, "Shirt", "Apparel"),

new Product(150, "Shoes", "Footwear"),

new Product(330, "Headphones", "Electronics"),

new Product(120, "Book", "Stationery")

};

int searchId = 150;

// Linear Search

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

Product foundLinear = linearSearch(products, searchId);

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

// Sort for Binary Search

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

// Binary Search

System.out.println("\nBinary Search:");

Product foundBinary = binarySearch(products, searchId);

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

}

// Include linearSearch and binarySearch methods here

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

for (Product p : products) {

if (p.productId == id) return p;

}

return null;

}

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

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

while (low <= high) {

int mid = (low + high) / 2;

if (products[mid].productId == id) return products[mid];

else if (products[mid].productId < id) low = mid + 1;

else high = mid - 1;

}

return null;

}

}

