**Algorithms and Data Structures**

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

**SOLUTION**

**Product.java**

package com.fse.ecom;

public class Product implements Comparable<Product> {

    int productId;

    String productName;

    String category;

    Product(int productId, String productName, String category) {

        this.productId = productId;

        this.productName = productName;

        this.category = category;

    }

    public String toString() {

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

    }

    @Override

    public int compareTo(Product other) {

        return Integer.compare(this.productId, other.productId); // for binary search

    }

}

**Search.java**

package com.fse.ecom;

import java.util.Arrays;

public class Search {

 // Linear Search

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

     for (Product product : products) {

         if (product.productId == targetId) {

             return product;

         }

     }

     return null;

 }

 // Binary Search

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

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

     while (left <= right) {

         int mid = (left + right) / 2;

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

             return products[mid];

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

             left = mid + 1;

         } else {

             right = mid - 1;

         }

     }

     return null;

 }

 // sort products

 public static void sortProducts(Product[] products) {

     Arrays.sort(products);

 }

}

**ECommerce.java**

package com.fse.ecom;

public class ECommerce {

 public static void main(String[] args) {

     Product[] products = {

         new Product(104, "Pen", "Education"),

         new Product(101, "Shirt", "Clothing"),

         new Product(108, "Book", "Education"),

         new Product(102, "Phone", "Electronics"),

         new Product(106, "Dress", "Clothing"),

         new Product(112, "TV", "Electronics")

     };

     int searchId = 106;

     // LINEAR SEARCH

     Product result1 = Search.linearSearch(products, searchId);

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

     if (result1 != null) {

         System.out.println(result1);

     } else {

         System.out.println("Product not found.");

     }

     // BINARY SEARCH

     Search.sortProducts(products);

     Product result2 = Search.binarySearch(products, searchId);

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

     if (result2 != null) {

         System.out.println(result2);

     } else {

         System.out.println("Product not found.");

     }

 }

}

**OUTPUT:**

