**DATA STRUCTURES & ALGORITHMS**

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

## **Code**:

### ***Product.java:***

| package com.productsearch;  public class Product {  private int productId;  private String productName;  private String category;   public Product(int productId, String productName, String category) {  this.productId = productId;  this.productName = productName;  this.category = category;  }   public int getProductId() { return productId; }  public String getProductName() { return productName; }  public String getCategory() { return category; }  *// method @Override for toString can also be implemented here.*  public String getDetails() {  return "Product{id=" + productId + ", name='" + productName + "', category='" + category + "'}";  }  } |
| --- |

### ***LinearSearch.java:***

| package com.productsearch;  public class LinearSearch {  public static Product search(Product[] products, int targetId) {  for (Product product : products) {  if (product.getProductId() == targetId) {  return product;  }  }  return null;  } } |
| --- |

### ***BinarySearch.java:***

| package com.productsearch;  public class BinarySearch {  public static void sortProducts(Product[] products) {  for (int i = 0; i < products.length - 1; i++) {  int minIndex = i;  for (int j = i + 1; j < products.length; j++) {  if (products[j].getProductId() < products[minIndex].getProductId()) {  minIndex = j;  }  }  Product temp = products[minIndex];  products[minIndex] = products[i];  products[i] = temp;  }  }    public static Product search(Product[] products, int targetId) {  int low = 0;  int high = products.length - 1;  while (low <= high) {  int mid = low + (high - low) / 2;  int midId = products[mid].getProductId();  if (midId == targetId) {  return products[mid];  } else if (midId < targetId) {  low = mid + 1;  } else {  high = mid - 1;  }  }  return null;  } } |
| --- |

### ***ProductSearch.java:***

| package com.productsearch;  import java.util.Scanner;  public class ProductSearch {  public static void main(String[] args) {  Product[] products = {  new Product(1, "Detergent", "HouseholdItem"),  new Product(3, "Camera", "Electronics"),  new Product(8, "Mobile", "ElectronicGadget"),  new Product(4, "Milk", "Dairy"),  new Product(9, "Table", "Furniture")  };  Scanner scanner = new Scanner(System.in);    System.out.print("Enter product ID to search: ");  int searchId = scanner.nextInt();   *//Linear Search*  System.out.println("Using inear Search:");  Product found = LinearSearch.search(products, searchId);  if (found != null) {  System.out.println("Found: " + found.getDetails());  } else {  System.out.println("Not found");  }    *//Binary Search*  BinarySearch.sortProducts(products);  System.out.println("\nBy using Binary Search:");  found = BinarySearch.search(products, searchId);  if (found != null) {  System.out.println("Found: " + found.getDetails());  } else {  System.out.println("Not found");  }   } } |
| --- |

## 

## **Output Screenshots:**

