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

## **2. Setup: Creating a Product Class**

Product.java

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;

}

}

## **3. Implementation: Linear and Binary Search**

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

for (Product product : products) {

if (product.productName.equals(targetName)) {

return product;

}

}

return null;

}

### 

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

int low = 0;

int high = products.length - 1;

while (low <= high) {

int mid = (low + high) / 2;

int compare = products[mid].productName.compareTo(targetName);

if (compare == 0) {

return products[mid];

} else if (compare < 0) {

low = mid + 1;

} else {

high = mid - 1;

}

}

return null;

}

### **Sample Product List & Sorting for Binary Search:**

import java.util.Arrays;

import java.util.Comparator;

public class Main {

public static void main(String[] args) {

Product[] products = {

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

new Product(102, "Lipstick", "Cosmetics"),

new Product(103, "Shirt", "Clothing"),

new Product(104, "Shoes", "Footwear"),

new Product(105, "Book", "Stationery")

};

Arrays.sort(products, Comparator.comparing(p -> p.productName));

Product result1 = linearSearch(products, "Shirt");

if (result1 != null) {

System.out.println("Linear Search Found: " + result1.productName + " in " + result1.category);

} else {

System.out.println("Linear Search: Product not found.");

}

Product result2 = binarySearch(products, "Laptop");

if (result2 != null) {

System.out.println("Binary Search Found: " + result2.productName + " in " + result2.category);

} else {

System.out.println("Binary Search: Product not found.");

}

}

}

**Output**

