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

import java.util.Arrays;

import java.util.Comparator;

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: " + productId + ", Name: " + productName + ", Category: " + category;

}

}

public class EcommerceSearch {

public static void main(String[] args) {

Product[] products = {

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

new Product(203, "Shoes", "Footwear"),

new Product(150, "Smartphone", "Electronics"),

new Product(305, "Watch", "Accessories")

};

Product foundLinear = linearSearch(products, 150);

System.out.println("Linear Search Found: " + (foundLinear != null ? foundLinear : "Not Found"));

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

Product foundBinary = binarySearch(products, 150);

System.out.println("Binary Search Found: " + (foundBinary != null ? foundBinary : "Not Found"));

}

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

for (Product product : products) {

if (product.productId == targetId) {

return product;

}

}

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;

}

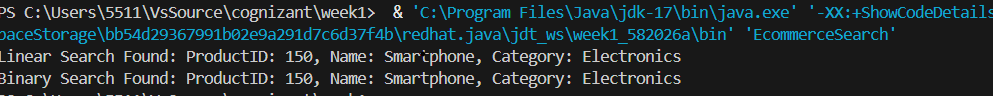
}

return null;

}

}

Output:



**Exercise 7: Financial Forecasting**

public class FinancialForecasting {

public static double futureValueRecursive(double presentValue, double growthRate, int years) {

if (years == 0) {

return presentValue;

}

return (1 + growthRate) \* futureValueRecursive(presentValue, growthRate, years - 1);

}

public static double futureValueIterative(double presentValue, double growthRate, int years) {

double value = presentValue;

for (int i = 0; i < years; i++) {

value \*= (1 + growthRate);

}

return value;

}

public static void main(String[] args) {

double presentValue = 1000.0;

double growthRate = 0.05;

int years = 5;

double resultRecursive = futureValueRecursive(presentValue, growthRate, years);

double resultIterative = futureValueIterative(presentValue, growthRate, years);

System.out.printf("Future Value (Recursive): %.2f%n", resultRecursive);

System.out.printf("Future Value (Iterative): %.2f%n", resultIterative);

}

}

Output:

