WEEK-1

MODULE-2

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

**package** EcommercePlatformSearchFunction;

**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;

}

@Override

**public** String toString() {

**return** "Product{id=" + productId + ", name='" + productName + "', category='" + category + "'}";

}

}

**package** EcommercePlatformSearchFunction;

**import** java.util.Arrays;

**import** java.util.Comparator;

**public** **class** ECommerceSearch {

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

**for** (Product p : products) {

**if** (p.getProductId() == id) {

**return** p;

}

}

**return** **null**;

}

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

**int** low = 0;

**int** high = products.length - 1;

**while** (low <= high) {

**int** mid = low + (high - low) / 2;

**if** (products[mid].getProductId() == id) {

**return** products[mid];

} **else** **if** (products[mid].getProductId() < id) {

low = mid + 1;

} **else** {

high = mid - 1;

}

}

**return** **null**;

}

**public** **static** **void** main(String[] args) {

// Sample Products

Product[] products = {

**new** Product(101, "Keyboard", "Electronics"),

**new** Product(105, "Mouse", "Electronics"),

**new** Product(102, "Headphones", "Electronics"),

**new** Product(103, "Laptop", "Electronics"),

**new** Product(104, "Monitor", "Electronics")

};

System.***out***.println("\nOriginal Products:");

**for** (Product p : products) {

System.***out***.println(p);

}

System.***out***.println("\nPerforming Linear Search for ID 103:");

Product result1 = *linearSearch*(products, 103);

System.***out***.println(result1 != **null** ? "Found: " + result1 : "Not Found");

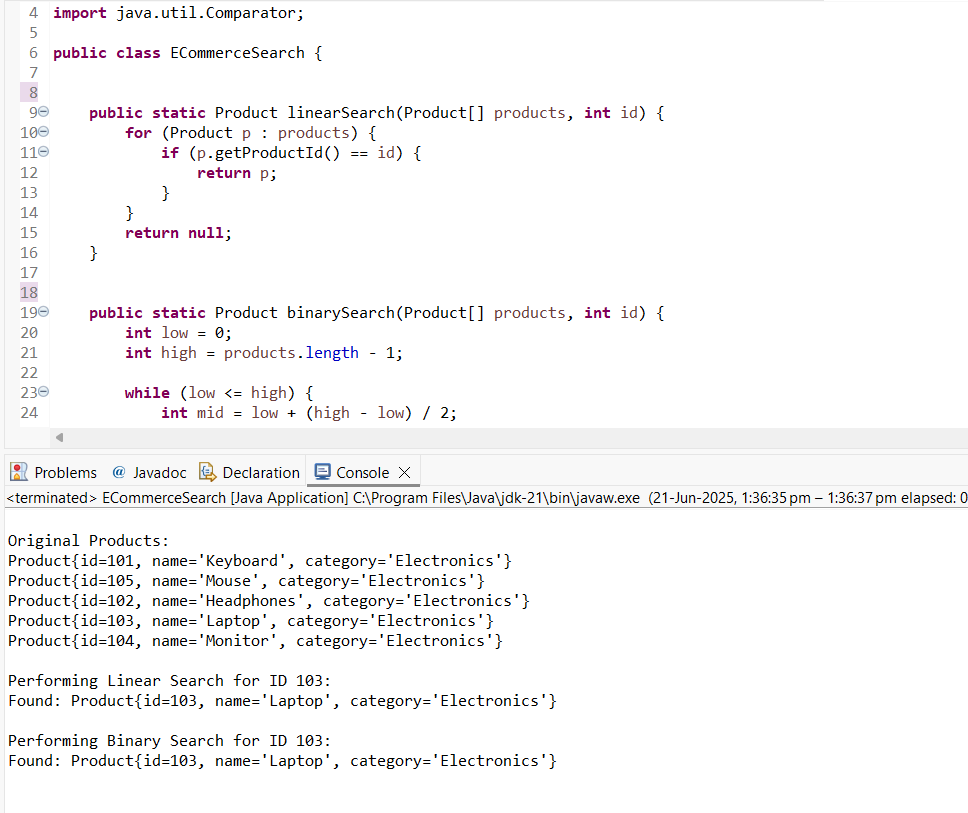
Arrays.*sort*(products, Comparator.*comparingInt*(Product::getProductId));

System.***out***.println("\nPerforming Binary Search for ID 103:");

Product result2 = *binarySearch*(products, 103);

System.***out***.println(result2 != **null** ? "Found: " + result2 : "Not Found");

}

}

**Exercise 7: Financial Forecasting**

**package** FinancialForecasting;

**public** **class** FinancialForecast {

**public** **static** **double** futureValue(**double** initialValue, **double** rate, **int** years) {

**if** (years == 0) {

**return** initialValue;

}

**return** *futureValue*(initialValue, rate, years - 1) \* (1 + rate);

}

**public** **static** **void** main(String[] args) {

**double** initialInvestment = 1000.0;

**double** annualGrowthRate = 0.05; // 5% annual growth

**int** years = 10;

**double** result = *futureValue*(initialInvestment, annualGrowthRate, years);

System.***out***.println("Future Value after " + years + " years: " + result);

}

}

