**DIGITAL NURTURE 4.0 DEEP SKILLING JAVA FSE-WEEK1**

**NAME: SIVITHA GUNASEKARAN   
SUPERSET ID: 6413354**

**WEEK 1: Algorithms Data Structures  
TASK 1:**

**E-commerce Platform Search Function**

**Scenario:**

You are working on the search functionality of an e-commerce platform. The search needs to be optimized for fast performance.

**Steps:**

1. **Understand Asymptotic Notation:**
   * Explain Big O notation and how it helps in analyzing algorithms.
   * Describe the best, average, and worst-case scenarios for search operations.
2. **Setup:**
   * Create a class **Product** with attributes for searching, such as **productId, productName**, and **category**.
3. **Implementation:**
   * Implement linear search and binary search algorithms.
   * Store products in an array for linear search and a sorted array for binary search.
4. **Analysis:**
   * Compare the time complexity of linear and binary search algorithms.
   * Discuss which algorithm is more suitable for your platform and why.

**MY CODE FILES:**

**Product.java**

**package** ecommerceproject;

**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;

}

}

**SearchFunction.java**

**package** ecommerceproject;

**import** java.util.\*;

**public** **class** SearchFunction {

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

**for** (**int** i = 0; i < products.length; i++) {

**if** (products[i].productName.equalsIgnoreCase(targetName)) {

**return** i;

}

}

**return** -1;

}

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

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

**while** (left <= right) {

**int** mid = (left + right) / 2;

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

**if** (compare == 0)

**return** mid;

**else** **if** (compare < 0)

left = mid + 1;

**else**

right = mid - 1;

}

**return** -1;

}

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

Product[] products = {

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

**new** Product(102, "Shirt", "Clothing"),

**new** Product(103, "Watch", "Accessories"),

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

};

**int** index1 = *linearSearch*(products, "Watch");

System.***out***.println("Linear Search: Found at index " + index1);

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

**int** index2 = *binarySearch*(products, "Watch");

System.***out***.println("Binary Search: Found at index " + index2);

}

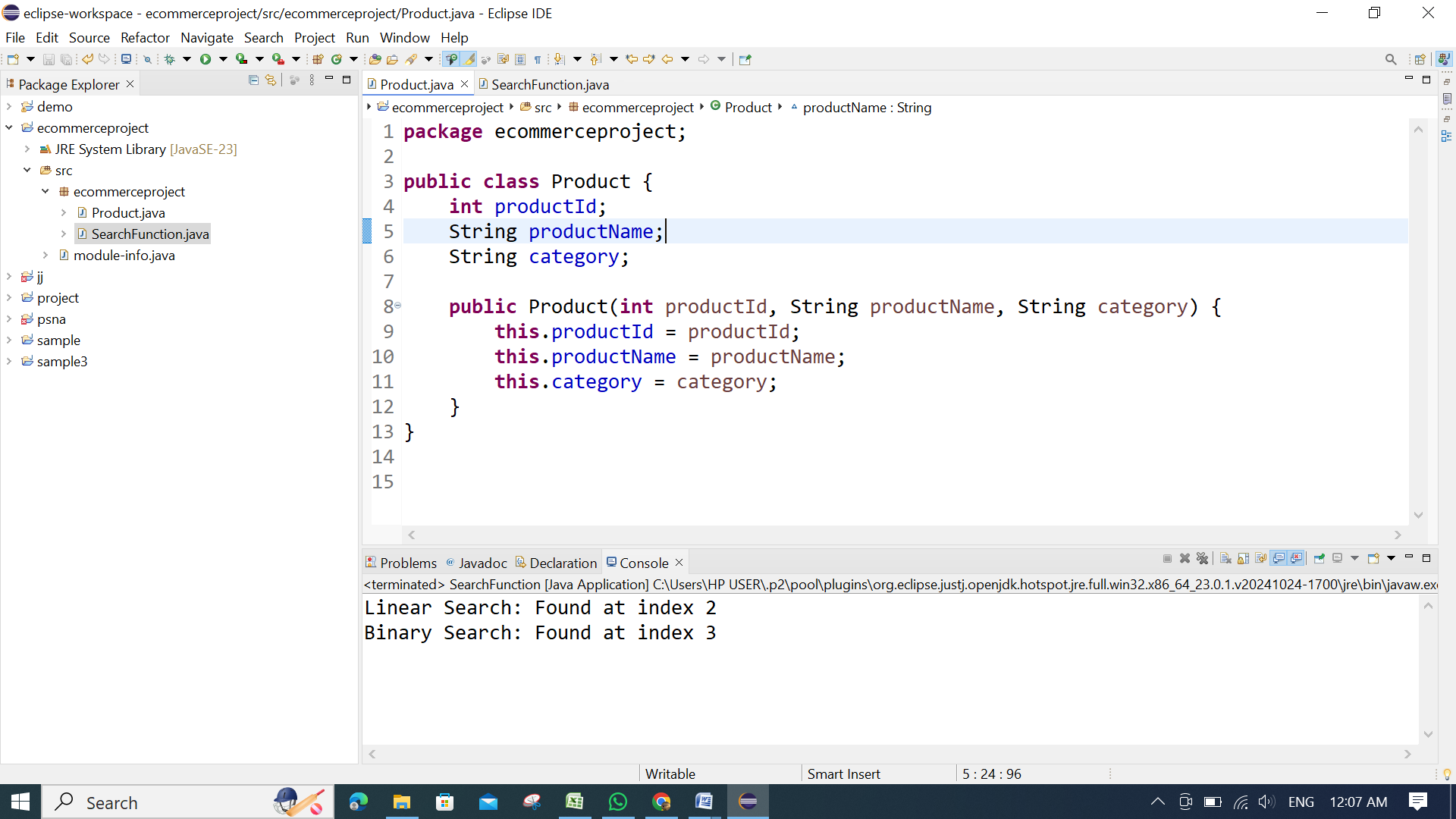
}

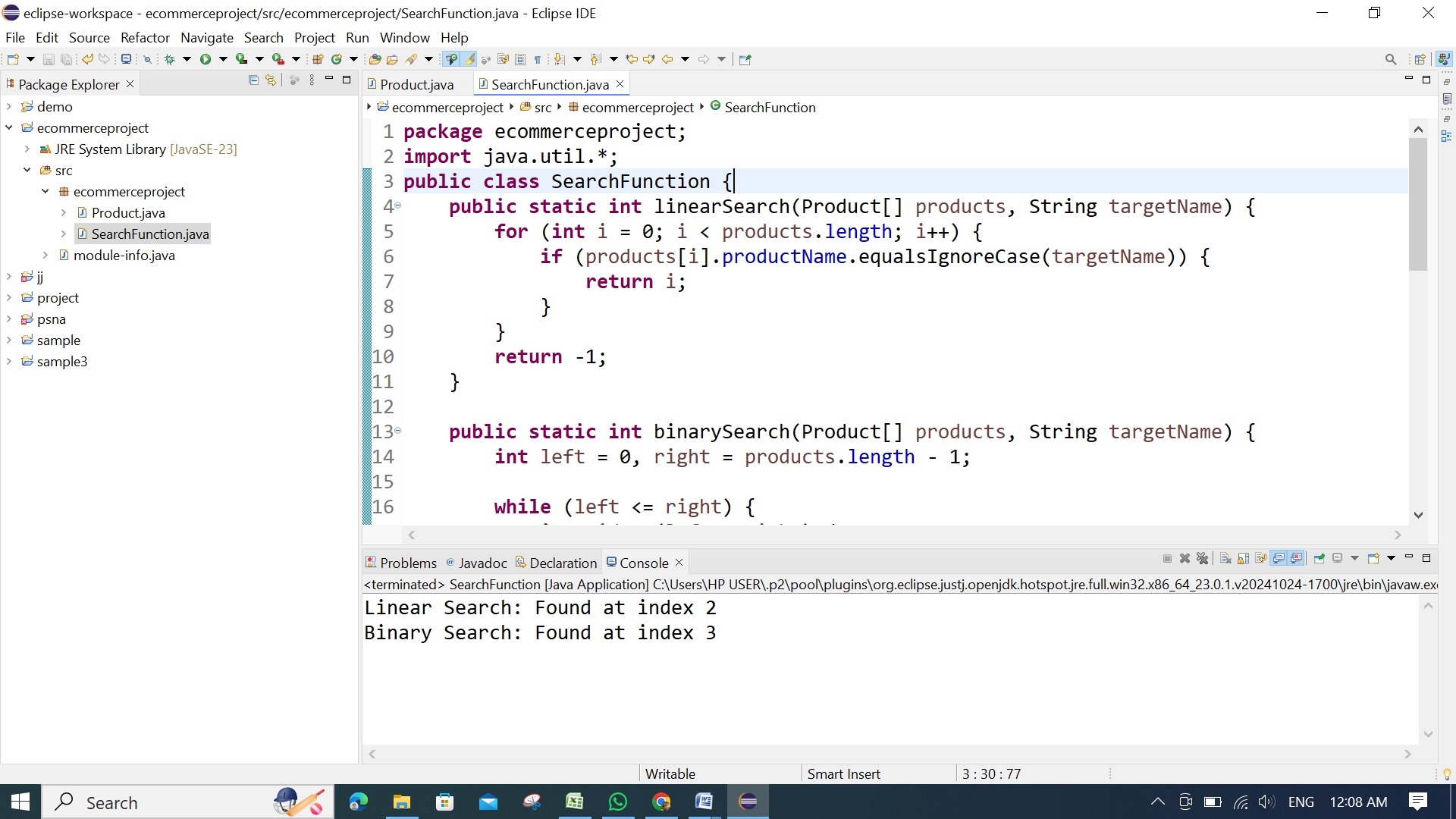
**Output:**

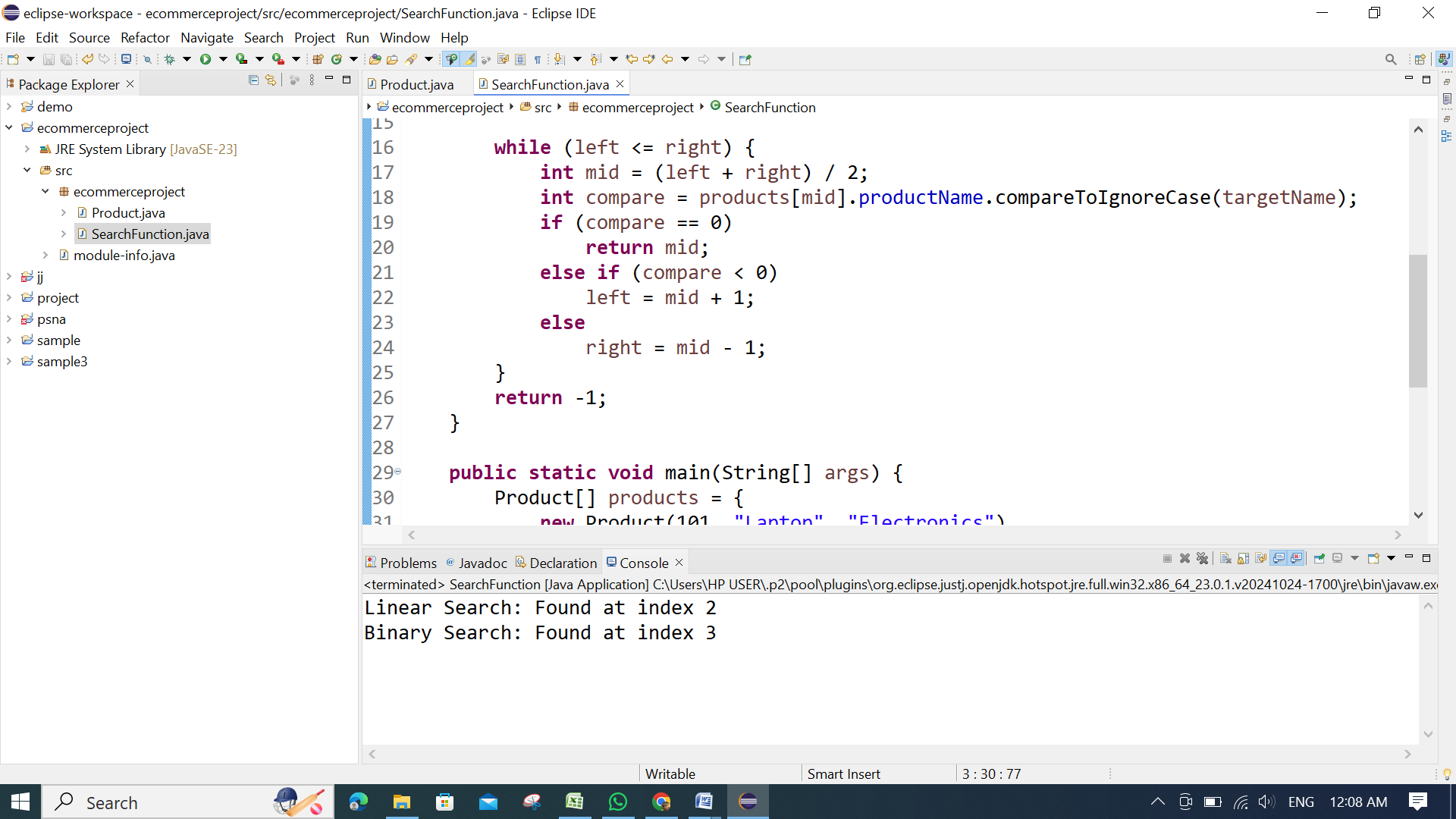
Linear Search: Found at index 2

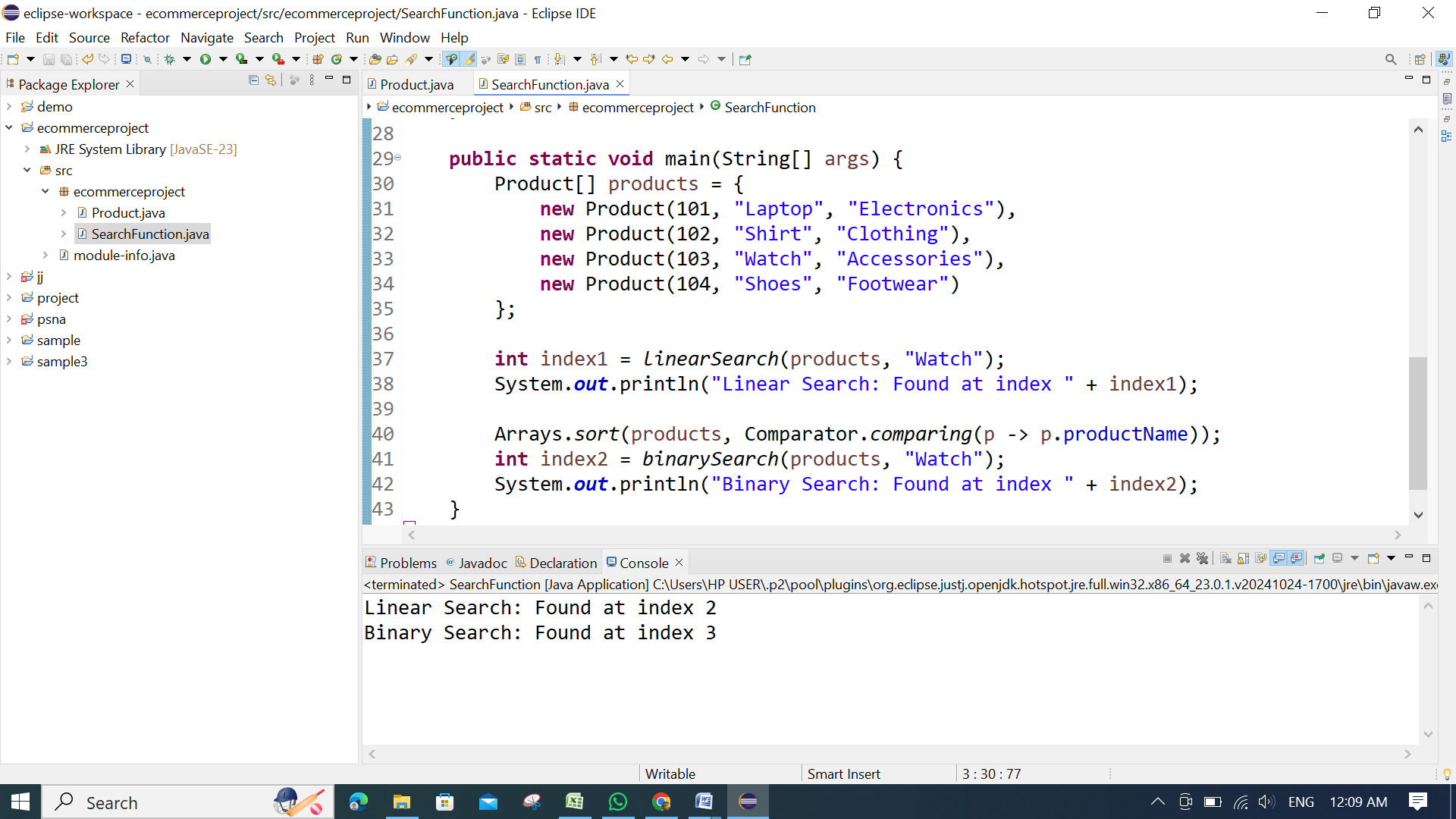
Binary Search: Found at index 3

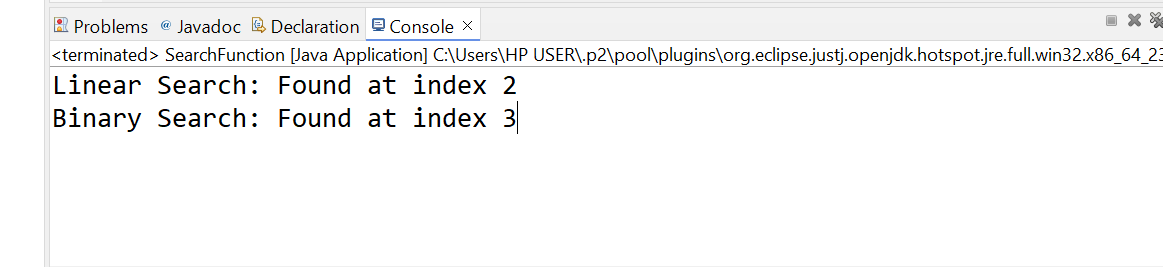
**MY SCREENSHOT PROOFS BY RUNNING THE ABOVE SAMPLE CODE:**

**Product.java**

**SearchFunction.java**





**OUTPUT:**