**Exercise 2: 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.

**1.Understand Asymptotic Notation :**

**Big O Notation :**

• Big O notation is a mathematical representation used to describe the upper limit of the time complexity of an algorithm. It provides a way to express how the runtime of an algorithm grows relative to the input size.

• It helps in analyzing algorithms by providing a high-level understanding of their efficiency, especially for large inputs.

**Best, Average, and Worst-Case Scenarios :**

• Best Case: The minimum time required for an algorithm to complete. For example, in a linear search, the best case occurs when the target element is the first element in the array.

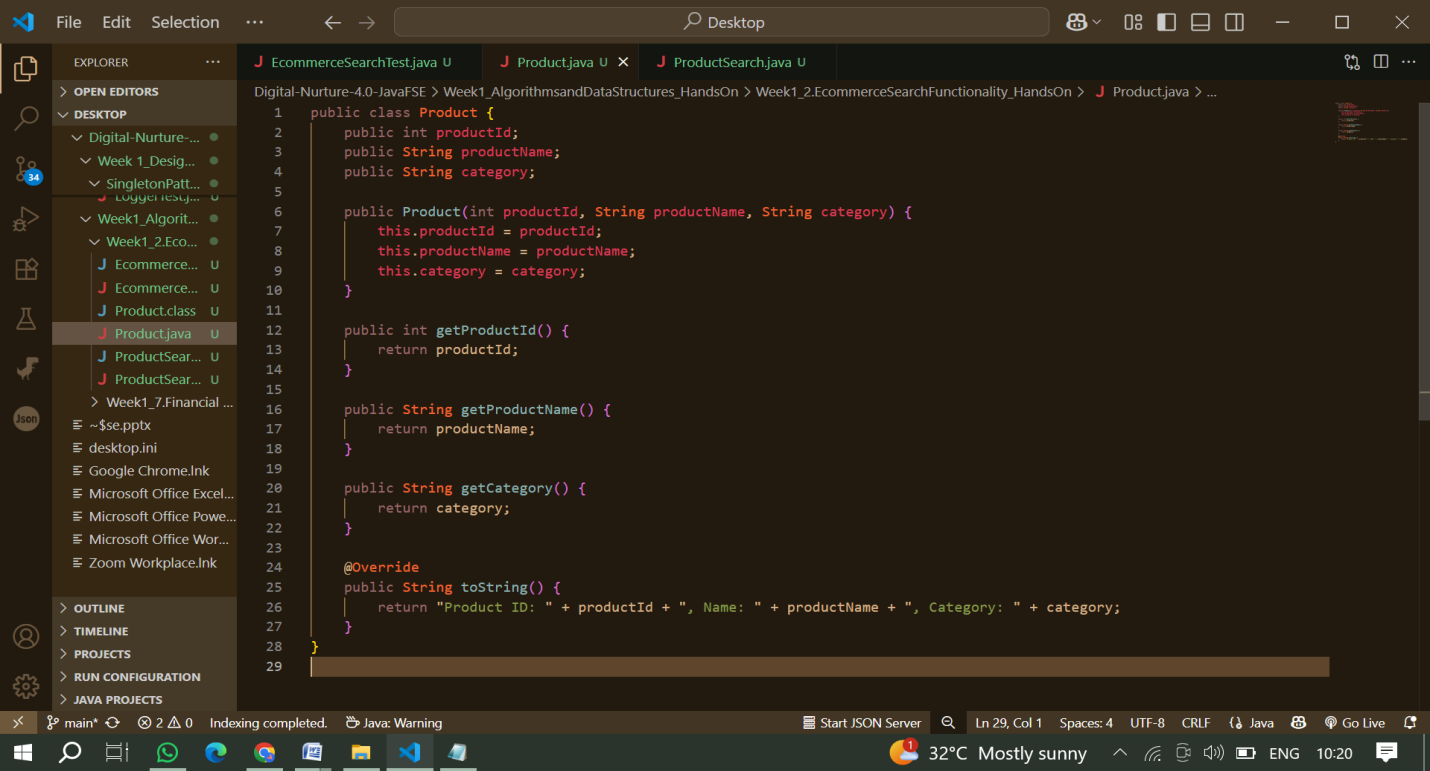
• Average Case: The expected time for an algorithm to complete, averaged over all possible inputs. For linear search, this is typically O(n/2), which simplifies to O(n).

• Worst Case: The maximum time required for an algorithm to complete. For linear search, this occurs when the target element is not present or is the last element, resulting in O(n). For binary search, the worst case is O(log n).

**2.SET-UP / PROCEDURE :**

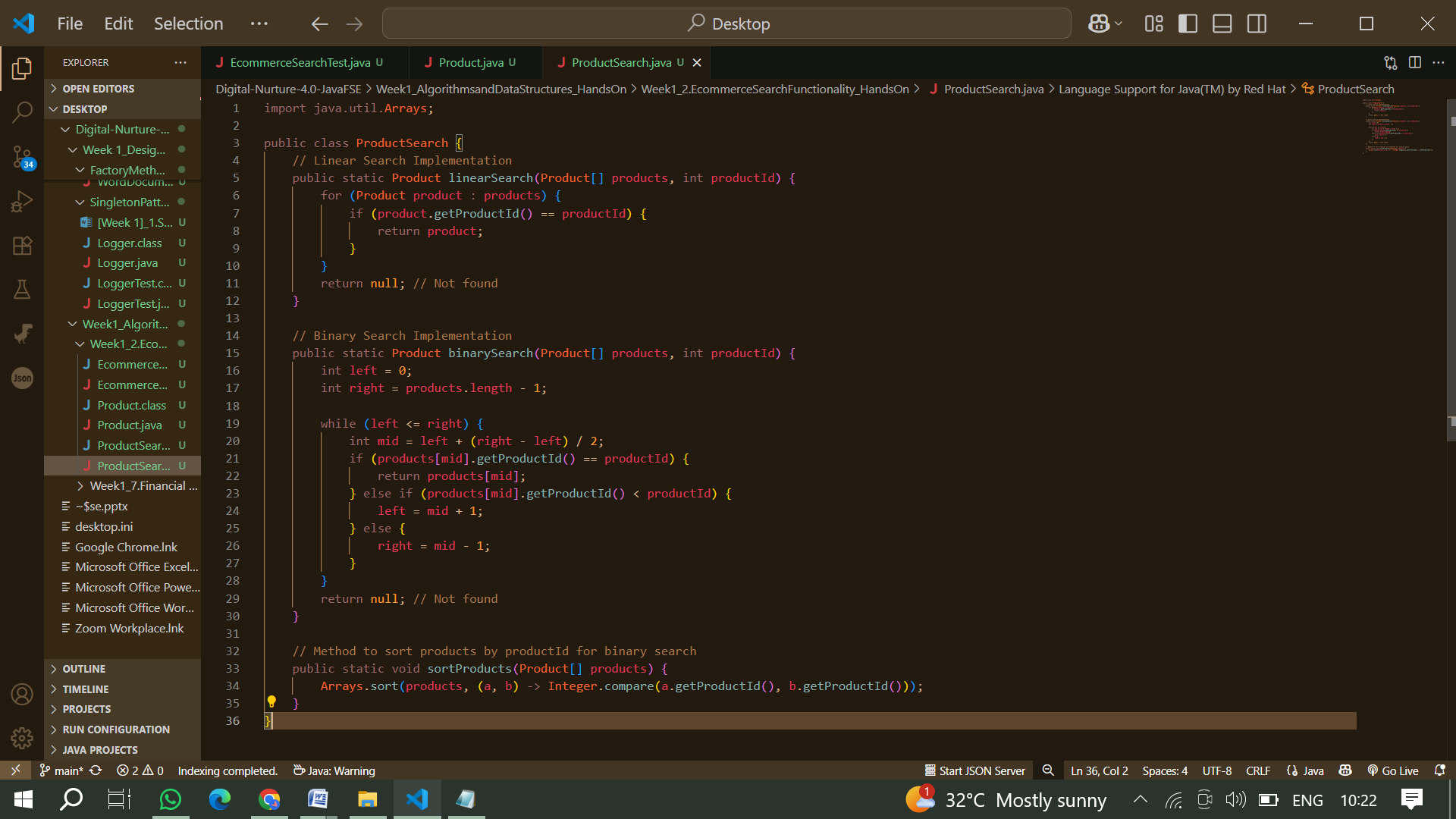
**1.Create the Product Class:**

a. Name the file Product.java and write the following code:



**2. Create a Search Class:**

a. Name the file ProductSearch.java and write the following code:

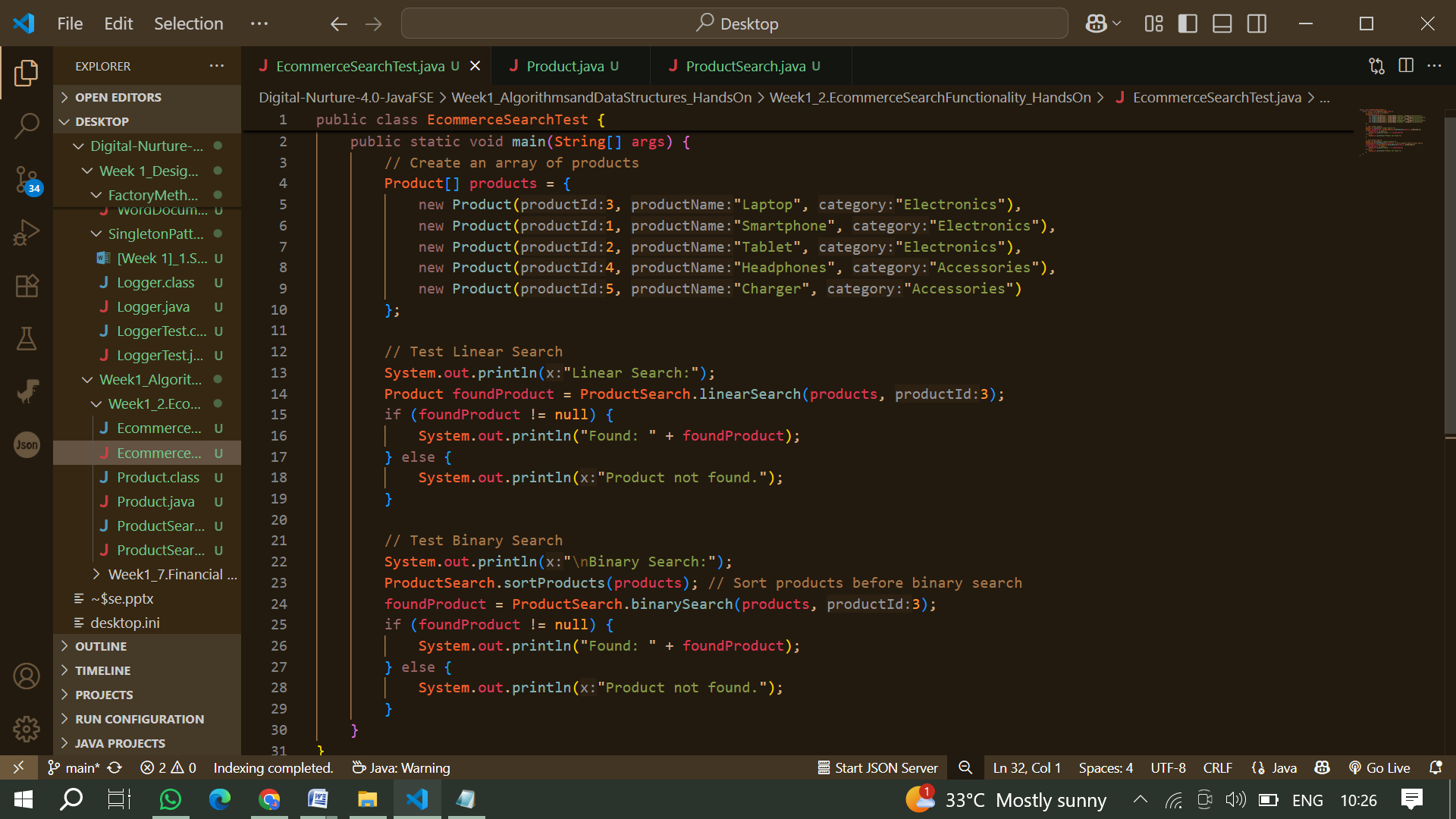


**3. IMPLEMENTATION**

**Create a Test Class**

a. Create a New Java Class for Testing, Name the file EcommerceSearchTest.java.

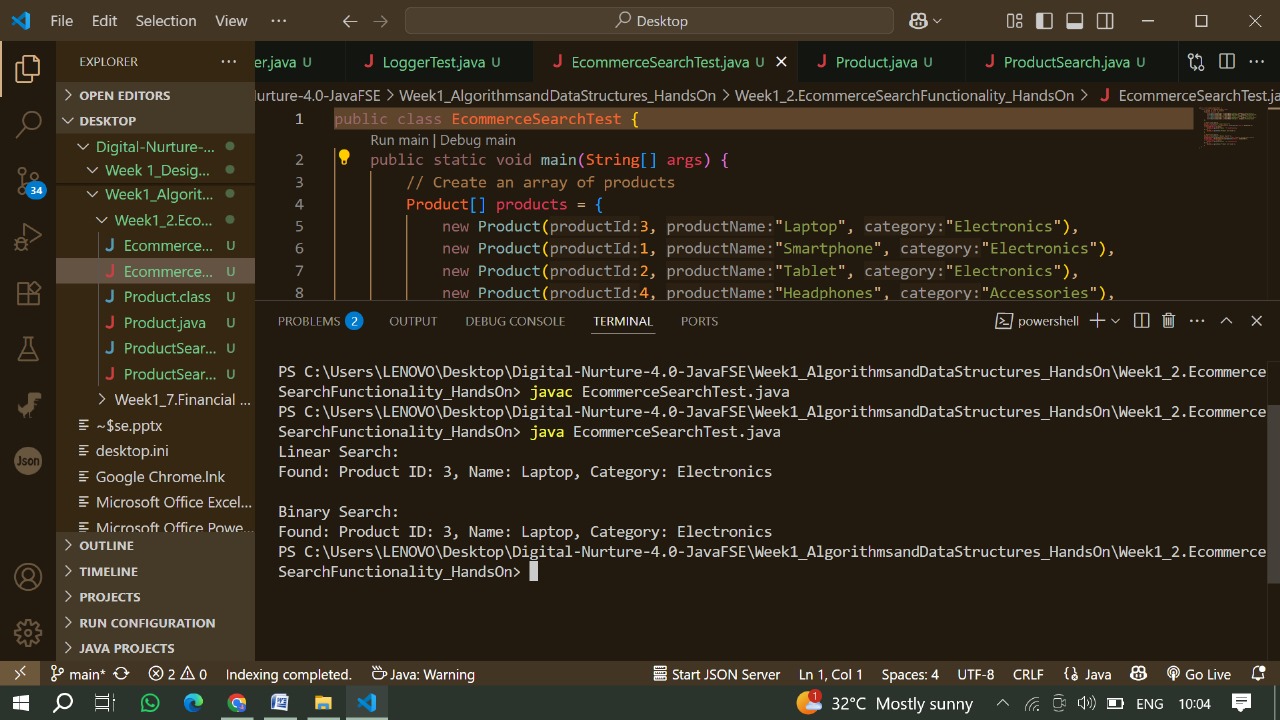
b. Write the following code:



**Compile the files and Run the test cases :**

Execute the EcommerceSearchTest class:

**OUTPUT :**



**4.Analysis**

**Time Complexity comparison:**

1. **Linear Search:**
2. Best Case: (element found at the first position)
3. Average Case:
4. Worst Case: (element not found or at the last position)
5. **Binary Search:**
6. Best Case: (element found at the middle)
7. Average Case:
8. Worst Case: (element not found)

**Which Algorithm is More Suitable?:**

1. Binary Search is more suitable for the e-commerce platform's search functionality because it has a significantly better time complexity of compared to linear search's. However, binary search requires the data to be sorted, which may involve additional overhead if the data changes frequently.
2. If the product list is static or changes infrequently, binary search is the preferred choice for faster search operations.