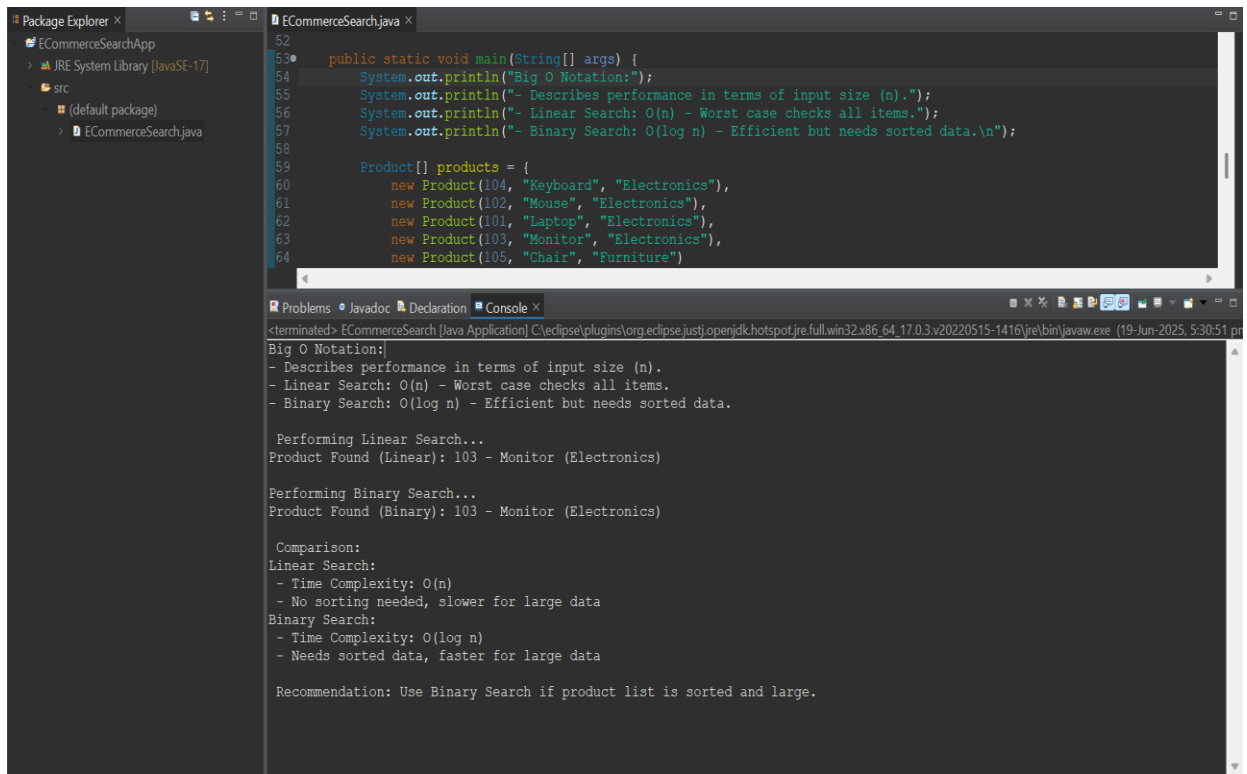


## Output:-



The screenshot shows the Eclipse IDE with the `ECommerceSearch.java` file open. The code defines a `Product` class and a `main` method. The `main` method prints "Big O Notation:" and then performs both Linear and Binary searches on a list of products. The console output shows the results of these searches and a comparison of their time complexities.

```
52
53 public static void main(String[] args) {
54     System.out.println("Big O Notation:");
55     System.out.println("- Describes performance in terms of input size (n).");
56     System.out.println("- Linear Search: O(n) - Worst case checks all items.");
57     System.out.println("- Binary Search: O(log n) - Efficient but needs sorted data.\n");
58
59     Product[] products = {
60         new Product(104, "Keyboard", "Electronics"),
61         new Product(102, "Mouse", "Electronics"),
62         new Product(101, "Laptop", "Electronics"),
63         new Product(103, "Monitor", "Electronics"),
64         new Product(105, "Chair", "Furniture")
65     };
66 }
```

Console Output:

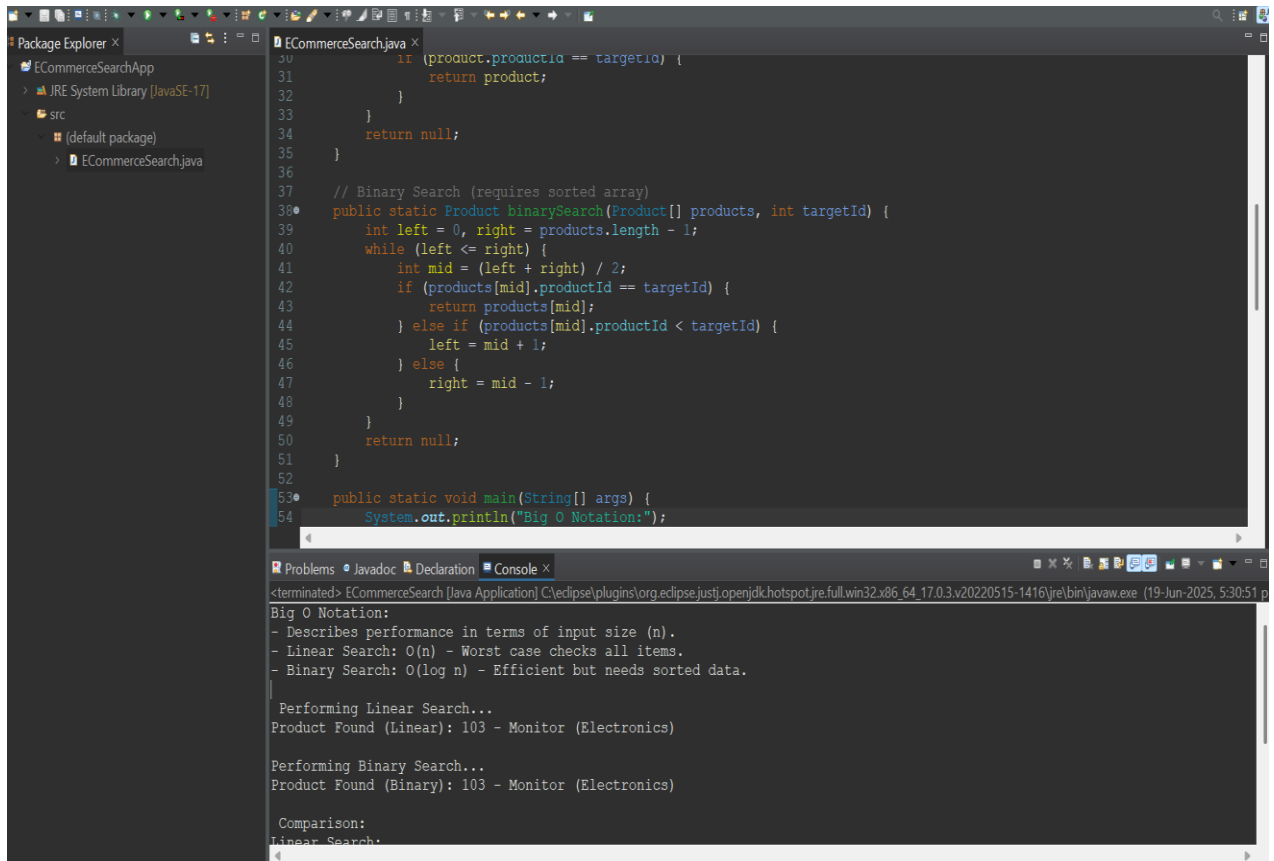
```
<terminated> ECommerceSearch [Java Application] C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.3.v20220515-1416\jre\bin\javaw.exe (19-Jun-2025, 5:30:51 PM)
Big O Notation:
- Describes performance in terms of input size (n).
- Linear Search: O(n) - Worst case checks all items.
- Binary Search: O(log n) - Efficient but needs sorted data.

Performing Linear Search...
Product Found (Linear): 103 - Monitor (Electronics)

Performing Binary Search...
Product Found (Binary): 103 - Monitor (Electronics)

Comparison:
Linear Search:
- Time Complexity: O(n)
- No sorting needed, slower for large data
Binary Search:
- Time Complexity: O(log n)
- Needs sorted data, faster for large data

Recommendation: Use Binary Search if product list is sorted and large.
```



The screenshot shows the Eclipse IDE with the `ECommerceSearch.java` file open. The code defines a `Product` class and a `main` method. The `main` method prints "Big O Notation:" and then performs both Linear and Binary searches on a list of products. The console output shows the results of these searches and a comparison of their time complexities.

```
30     if (product.productId == targetId) {
31         return product;
32     }
33 }
34 return null;
35 }
36
37 // Binary Search (requires sorted array)
38 public static Product binarySearch(Product[] products, int targetId) {
39     int left = 0, right = products.length - 1;
40     while (left <= right) {
41         int mid = (left + right) / 2;
42         if (products[mid].productId == targetId) {
43             return products[mid];
44         } else if (products[mid].productId < targetId) {
45             left = mid + 1;
46         } else {
47             right = mid - 1;
48         }
49     }
50     return null;
51 }
52
53 public static void main(String[] args) {
54     System.out.println("Big O Notation:");
55     System.out.println("- Describes performance in terms of input size (n).");
56     System.out.println("- Linear Search: O(n) - Worst case checks all items.");
57     System.out.println("- Binary Search: O(log n) - Efficient but needs sorted data.\n");
58
59     Product[] products = {
60         new Product(104, "Keyboard", "Electronics"),
61         new Product(102, "Mouse", "Electronics"),
62         new Product(101, "Laptop", "Electronics"),
63         new Product(103, "Monitor", "Electronics"),
64         new Product(105, "Chair", "Furniture")
65     };
66 }
```

Console Output:

```
<terminated> ECommerceSearch [Java Application] C:\eclipse\plugins\org.eclipse.justi.openjdk.hotspot.jre.full.win32.x86_64.17.0.3.v20220515-1416\jre\bin\javaw.exe (19-Jun-2025, 5:30:51 PM)
Big O Notation:
- Describes performance in terms of input size (n).
- Linear Search: O(n) - Worst case checks all items.
- Binary Search: O(log n) - Efficient but needs sorted data.

Performing Linear Search...
Product Found (Linear): 103 - Monitor (Electronics)

Performing Binary Search...
Product Found (Binary): 103 - Monitor (Electronics)

Comparison:
Linear Search:
- Time Complexity: O(n)
- No sorting needed, slower for large data
Binary Search:
- Time Complexity: O(log n)
- Needs sorted data, faster for large data

Recommendation: Use Binary Search if product list is sorted and large.
```