E-commerce Platform Search Function

# 1. Understand Asymptotic Notation

Big O notation describes how fast or slow an algorithm performs relative to the size of the input (n). It helps compare algorithms regardless of hardware. It's especially useful for analyzing the efficiency of search and sort operations.

Search Operation Cases:

|  |  |  |  |
| --- | --- | --- | --- |
| Scenario | Linear Search | Binary Search |  |
| Best Case | O(1) — first match | O(1) — middle match |  |
| Average Case | O(n) | O(log n) |  |
| Worst Case | O(n) | O(log n) |  |

# 2. Setup: Product Class

**CODE**

class Product {

int productId;

String productName;

String category;

Product(int id, String name, String cat) {

productId = id;

productName = name;

category = cat;

}

public String toString() {

return productId + ": " + productName + " - " + category;

}

}

public class Main {

public static void main(String[] args) {

Product p1 = new Product(101, "Laptop", "Electronics");

Product p2 = new Product(102, "Shirt", "Clothing");

Product p3 = new Product(103, "Book", "Stationery");

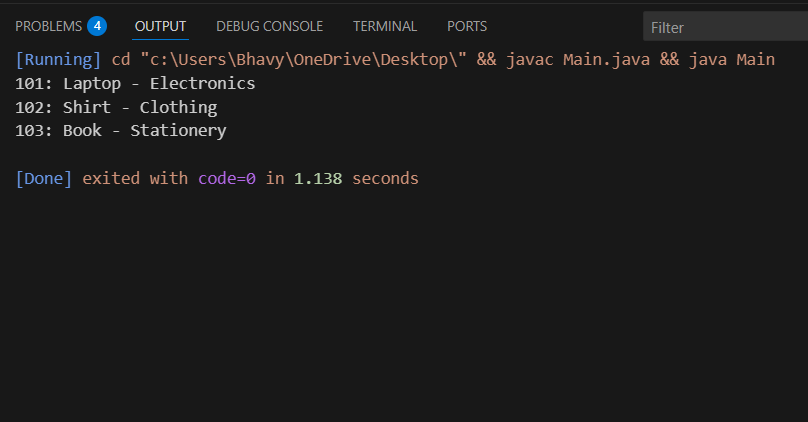
System.out.println(p1);

System.out.println(p2);

System.out.println(p3);

}

**OUTPUT**

}

# 3. Implementation: Linear Search

import java.util.Scanner;

class Product {

int id;

String name;

Product(int id, String name) {

this.id = id;

this.name = name;

} public String toString() {

return id + ": " + name;

}

}

public class Main {

public static void main(String[] args) {

Product[] products = {

new Product(1, "Laptop"),

new Product(2, "Book"),

new Product(3, "Phone")

};

Scanner sc = new Scanner(System.in);

System.out.print("Enter product ID to search: ");

int searchId = sc.nextInt();

boolean found = false;

for (Product p : products) {

if (p.id == searchId) {

System.out.println("Found: " + p);

found = true;

break;

}

} if (!found) {

System.out.println("Product not found");

}

sc.close();

}

}

## OUTPUT

## Binary Search (sorted array)

import java.util.\*;

class Product {

int id; String name;

Product(int id, String name) { this.id = id; this.name = name; }

public String toString() { return id + ": " + name; }

}

public class Main {

public static void main(String[] args) {

Product[] p = {

new Product(1, "Laptop"),

new Product(2, "Book"),

new Product(3, "Phone"),

new Product(4, "Mouse")

};

Scanner sc = new Scanner(System.in);

System.out.print("Enter ID: ");

int id = sc.nextInt();

int l = 0, r = p.length - 1;

boolean f = false;

while (l <= r) {

int m = (l + r) / 2;

if (p[m].id == id) { System.out.println("Found: " + p[m]); f = true; break; }

if (p[m].id < id) l = m + 1;

else r = m - 1;

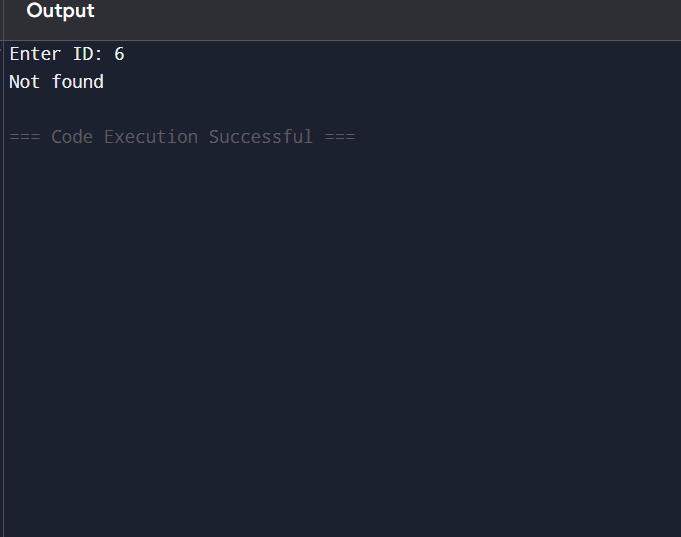
}

if (!f) System.out.println("Not found");

sc.close();

}

}

**OUTPUT**

Use Linear Search if:  
- Your product list is small or unsorted  
- You want a quick and easy solution  
  
Use Binary Search if:  
- Your product list is large  
- You can keep the data sorted by ID (or name)

**In real e-commerce platforms**, we use **binary search** for faster scalability.