# Week1 - Algorithms & Data Structures - HandsOn

## Exercise: EcommerceSearch

EcommerceSearch.java

import java.util.\*;  
  
class Product {  
 int productId;  
 String productName;  
 String category;  
  
 Product(int productId, String productName, String category) {  
 this.productId = productId;  
 this.productName = productName;  
 this.category = category;  
 }  
}  
  
public class EcommerceSearch {  
 static int linearSearch(Product[] arr, String name) {  
 for (int i = 0; i < arr.length; i++) {  
 if (arr[i].productName.equalsIgnoreCase(name)) return i;  
 }  
 return -1;  
 }  
  
 static int binarySearch(Product[] arr, String name) {  
 int low = 0, high = arr.length - 1;  
 while (low <= high) {  
 int mid = (low + high) / 2;  
 int cmp = arr[mid].productName.compareToIgnoreCase(name);  
 if (cmp == 0) return mid;  
 else if (cmp < 0) low = mid + 1;  
 else high = mid - 1;  
 }  
 return -1;  
 }

public static void main(String[] args) {  
 Product[] products = {  
 new Product(1, "Charger", "Electronics"),  
 new Product(2, "Headphones", "Audio"),  
 new Product(3, "Laptop", "Computers")  
 };  
 Arrays.sort(products, Comparator.comparing(p -> p.productName));  
 System.out.println("Linear Search Index: " + linearSearch(products, "Laptop"));  
 System.out.println("Binary Search Index: " + binarySearch(products, "Laptop"));  
 }  
}

### Output: A screenshot of a computer AI-generated content may be incorrect.

## Exercise: Financial Forecasting

FinancialForecasting.java

public class FinancialForecasting {  
  
 static double forecast(int years, double currentValue, double growthRate) {  
 if (years == 0) return currentValue;  
 return forecast(years - 1, currentValue \* (1 + growthRate), growthRate);  
 }  
  
 public static void main(String[] args) {  
 double initial = 10000; // we can also take dynamic input here using Scanner  
 double growthRate = 0.05; // 5%  
 int years = 5; // we can also take dynamic inputs here using Scanner  
  
 double result = forecast(years, initial, growthRate);  
 System.out.printf("Forecast after %d years: %.2f\n", years, result);  
 }  
}

### Output: A screenshot of a computer AI-generated content may be incorrect.