

WEEK-01

Algorithms and Data Structures

- **Exercise 2: E-commerce Platform Search Function**

➔Solution:

Code:

BinarySearch.java

```
public class BinarySearch {  
    public static Product binSearch(Product[] pdts, int id) {  
        int s=0;  
        int e = pdts.length - 1;  
        while (s <= e) {  
            int mid = (s + e) / 2;  
            if (pdts[mid].productId == id) {  
                return pdts[mid];  
            } else if (pdts[mid].productId < id) {  
                s = mid + 1;  
            } else {  
                e = mid - 1;  
            }  
        }  
        return null;  
    }  
}
```

LinearSearch.java

```
public class LinearSearch {  
    public static Product linSearch(Product[] pdts, int id){  
        for (Product pdt : pdts) {  
            if (pdt.productId == id) {  
                return pdt;  
            }  
        }  
        return null;  
    }  
}
```

BubbleSort.java

```
public class BubbleSort {
    static void sort(Product[] arr) {
        int n = arr.length;
        for (int i = 0; i < n; i++) {
            for (int j = 0; j < n - i - 1; j++) {
                if (arr[j].productId > arr[j + 1].productId) {
                    Product p = (arr[j+1]);
                    arr[j+1] = arr[j];
                    arr[j] = p;
                }
            }
        }
    }

    static void printArr(Product[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.println(arr[i]);
        }
    }
}
```

Main.java

```
public class Main {
    public static void main(String[] args) {
        Product[] pdts = {
            new Product(1, "AC", "Appliances"),
            new Product(3, "Cards", "Toys"),
            new Product(2, "Deodrants", "Beauty"),
            new Product(4, "Smart watch", "Gadgets")
        };
        System.out.println("Linear Search: ");
        Product pdt = LinearSearch.linSearch(pdts, 4);
        System.out.println(pdt);
        System.out.println();

        BubbleSort.sort(pdts);
        System.out.println("products list after sorting id:");
        BubbleSort.printArr(pdts);
        System.out.println();

        System.out.println("Binary Search: ");
        Product pdt2 = BinarySearch.binSearch(pdts, 3);
    }
}
```

```
        System.out.println("found pdt: "+ pdt2);
    }
}
```

O/P:

```
PS C:\Users\schow\Desktop\cts dn 4.0\Deepskilling\solution\week _01> javac EcommerceSearch/Main.java
PS C:\Users\schow\Desktop\cts dn 4.0\Deepskilling\solution\week _01> java EcommerceSearch/Main
Linear Search:
productId=4, productName='Smart watch', category='Gadgets'

products list after sorting id:
productId=1, productName='AC', category='Appliances'
productId=2, productName='Deodrants', category='Beauty'
productId=3, productName='Cards', category='Toys'
productId=4, productName='Smart watch', category='Gadgets'

Binary Search:
found pdt: productId=3, productName='Cards', category='Toys'
```

- **Exercise 7: Financial Forecasting**

➔ **Solution:**

Code:

Finance.java

```
import java.util.Scanner;

public class Finance {
    public static double calculate(double val, double rate, int year) {
        if (year == 0) {
            return val;
        }
        return calculate((val + val * rate), rate, year - 1);
    }

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);

        System.out.print("Enter current value: ");
```

```
float currentVal = sc.nextFloat();

System.out.print("Enter growth rate: ");
float rate = sc.nextFloat();

System.out.print("Enter the number of years: ");
int year = sc.nextInt();

double val = calculate(currentVal, rate, year);
System.out.println(val);
}
}
```

O/P:

```
PS C:\Users\schow\Desktop\cts dn 4.0\Deepskilling\solution\week _01> javac FinancialForecasting/Finance.java
PS C:\Users\schow\Desktop\cts dn 4.0\Deepskilling\solution\week _01> java FinancialForecasting/Finance
Enter current value: 4000
Enter growth rate: 0.03
Enter the number of years: 20
7224.4448446122615
```