

## Lab#2

### Lab Tasks

1. Write a program that initializes Vector with 10 integers in it. Display all the integers and sum of these integers.

```
public class VectorSum {  
    public static void main(String[] args) {  
  
        Vector<Integer> numbers = new Vector<>();  
        numbers.add(5);  
        numbers.add(10);  
        numbers.add(15);  
        numbers.add(20);  
        numbers.add(25);  
        numbers.add(30);  
        numbers.add(35);  
        numbers.add(40);  
        numbers.add(45);  
        numbers.add(50);  
        int sum = 0;  
        System.out.println("NUMBERS are =" + numbers);  
        for (int num: numbers){  
            sum += num;  
        }  
        System.out.println("Sum of Numbers = " + sum);  
    }  
}
```

Output:

```
NUMBERS are =[5, 10, 15, 20, 25, 30, 35, 40, 45, 50]  
Sum of Numbers = 275
```

## 2. Create a ArrayList of string. Write a menu driven program which: a. Displays all the elements b. Displays the largest String

```
import java.util.ArrayList;
import java.util.Scanner;

public class StringArray {
    public static void main(String[] args) {
        ArrayList<String> strings = new ArrayList<>();
        Scanner scanner = new Scanner(System.in);
        strings.add("apple");
        strings.add("banana");
        strings.add("watermelon");
        strings.add("kiwi");
        strings.add("strawberry");
        while (true) {
            System.out.println("Menu:");
            System.out.println("1. Display all elements");
            System.out.println("2. Display the largest string");
            System.out.print("Choose an option: ");
            int choice = scanner.nextInt();
            scanner.nextLine();

            switch (choice) {
                case 1:
                    System.out.println("Elements in the ArrayList:");
                    for (String str : strings) {
                        System.out.println(str);
                    }
                    break;

                case 2:
                    String largestString = "";
                    for (String str : strings) {
                        if (str.length() > largestString.length()) {
                            largestString = str;
                        }
                    }
                    System.out.println("Largest string: " + largestString);
                    break;

                default:
                    System.out.println("Invalid option. Please choose again.");
            }
            System.out.println();
        }
    }
}
```

Menu:

1. Display all elements

2. Display the largest string

Choose an option: Largest string: watermelon

3. Create a ArrayList storing Employee details including Emp\_id, Emp\_Name, Emp\_gender, Year\_of\_Joining .Then sort the employees according to their joining year using Comparator and Comparable interfaces.

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.List;

class Employee implements Comparable<Employee> {
    private int empId;
    private String empName;
    private String empGender;
    private int yearOfJoining;

    public Employee(int empId, String empName, String empGender, int yearOfJoining) {
        this.empId = empId;
        this.empName = empName;
        this.empGender = empGender;
        this.yearOfJoining = yearOfJoining;
    }

    public int getYearOfJoining() {
        return yearOfJoining;
    }

    @Override
    public String toString() {
        return "Employee{" +
            "Emp ID=" + empId +
            ", Name=" + empName + '\'' +
            ", Gender=" + empGender + '\'' +
            ", Year of Joining=" + yearOfJoining +
            '}';
    }

    @Override
    public int compareTo(Employee other) {
        return Integer.compare(this.yearOfJoining, other.yearOfJoining);
    }
}

public class EmployeesSorting {
    public static void main(String[] args) {
        List<Employee> employees = new ArrayList<>();

        employees.add(new Employee(1, "Alice", "Female", 2019));
        employees.add(new Employee(2, "Bob", "Male", 2017));
        employees.add(new Employee(3, "Charlie", "Male", 2020));
        employees.add(new Employee(4, "Diana", "Female", 2018));
        Collections.sort(employees);
        System.out.println("Employees sorted by year of joining (using Comparable):");
        for (Employee emp : employees) {
            System.out.println(emp);
        }

        Comparator<Employee> byYearOfJoining = new Comparator<Employee>() {
            @Override
            public int compare(Employee e1, Employee e2) {
                return Integer.compare(e1.getYearOfJoining(), e2.getYearOfJoining());
            }
        };

        Collections.sort(employees, byYearOfJoining);
        System.out.println("\nEmployees sorted by year of joining (using Comparator):");
        for (Employee emp : employees) {
            System.out.println(emp);
        }
    }
}
```

Employees sorted by year of joining (using Comparable):  
 Employee{Emp ID=2, Name='Bob', Gender='Male', Year of Joining=2017}  
 Employee{Emp ID=4, Name='Diana', Gender='Female', Year of Joining=2018}  
 Employee{Emp ID=1, Name='Alice', Gender='Female', Year of Joining=2019}  
 Employee{Emp ID=3, Name='Charlie', Gender='Male', Year of Joining=2020}

Employees sorted by year of joining (using Comparator):  
 Employee{Emp ID=2, Name='Bob', Gender='Male', Year of Joining=2017}  
 Employee{Emp ID=4, Name='Diana', Gender='Female', Year of Joining=2018}  
 Employee{Emp ID=1, Name='Alice', Gender='Female', Year of Joining=2019}  
 Employee{Emp ID=3, Name='Charlie', Gender='Male', Year of Joining=2020}

4. Write a program that initializes Vector with 10 integers in it. Display all the integers ,Sum of these integers, Find Maximum Element in Vector .

```
import java.util.Vector;

public class VectorOperations {
    public static void main(String[] args) {
        Vector<Integer> numbers = new Vector<>();
        numbers.add(5);
        numbers.add(10);
        numbers.add(15);
        numbers.add(20);
        numbers.add(25);
        numbers.add(30);
        numbers.add(35);
        numbers.add(40);
        numbers.add(45);
        numbers.add(50);
        System.out.println("Integers in the Vector:");
        for (int number : numbers) {
            System.out.print(number + " ");
        }
        System.out.println();

        int sum = 0;
        for (int number : numbers) {
            sum += number;
        }
        System.out.println("Sum of integers: " + sum);
        int max = numbers.get(0);
        for (int number : numbers) {
            if (number > max) {
                max = number;
            }
        }
        System.out.println("Maximum element: " + max);
    }
}
```

Integers in the Vector:

5 10 15 20 25 30 35 40 45 50

Sum of integers: 275

Maximum element: 50

## 5. Find the k-th smallest element in a sorted ArrayList

```
import java.util.ArrayList;
import java.util.Collections;

public class KthSmallestElement {
    public static void main(String[] args) {
        ArrayList<Integer> sortedList = new ArrayList<>();
        Collections.addAll(sortedList, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19);
        int k = 5;
        if (k > 0 && k <= sortedList.size()) {
            int kthSmallest = sortedList.get(k - 1);
            System.out.println("The " + k + "-th smallest element is: " + kthSmallest);
        } else {
            System.out.println("Invalid value of k: " + k);
        }
    }
}
```

Output:

The 5-th smallest element is: 9

## 6. Write a program to merge two ArrayLists into one.

```
import java.util.ArrayList;

public class MergeArrayLists {
    public static void main(String[] args) {
        ArrayList<String> list1 = new ArrayList<>();
        list1.add("Apple");
        list1.add("Banana");
        list1.add("Cherry");
        ArrayList<String> list2 = new ArrayList<>();
        list2.add("Date");
        list2.add("Elderberry");
        list2.add("Fig");
        list1.addAll(list2);
        System.out.println("Merged ArrayList:");
        for (String fruit : list1) {
            System.out.println(fruit);
        }
    }
}
```

Output:

Merged ArrayList:

Apple  
Banana  
Cherry  
Date  
Elderberry  
Fig

## Home Tasks

1. Create a Vector storing integer objects as an input.

- a. Sort the vector
- b. Display largest number
- c. Display smallest number

```
import java.util.Collections;
import java.util.Scanner;
import java.util.Vector;

public class InputVectorOperations {
    public static void main(String[] args) {
        Vector<Integer> numbers = new Vector<>();
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter integers (type 'done' to finish):");
        while (scanner.hasNext()) {
            if (scanner.hasNextInt()) {
                numbers.add(scanner.nextInt());
            } else if (scanner.next().equalsIgnoreCase("done")) {
                break;
            } else {
                System.out.println("Please enter a valid integer or 'done' to finish.");
            }
        }
        if (numbers.isEmpty()) {
            System.out.println("No integers were entered.");
        } else {
            Collections.sort(numbers);
            System.out.println("Sorted Vector: " + numbers);
            System.out.println("Largest number: " + numbers.lastElement());
            System.out.println("Smallest number: " + numbers.firstElement());
        }
        scanner.close();
    }
}
```

Output:

Enter integers (type 'done' to finish):

Sorted Vector: [3, 4, 5, 6, 7, 8]

Largest number: 8

Smallest number: 3

**2. Write a java program which takes user input and gives hashCode value of those inputs using hashCode () method.**

```
import java.util.Scanner;

public class HashCodeExample {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        System.out.println("Enter a string (type 'exit' to finish):");

        while (true) {
            String input = scanner.nextLine();
            if (input.equalsIgnoreCase("exit")) {
                break;
            }
            int hashCode = input.hashCode();
            System.out.println("Input: \"\" + input + \"\" has hash code: \"\" + hashCode);
        }

        scanner.close();
        System.out.println("Program ended.");
    }
}
```

Output:

Enter a string (type 'exit' to finish):

Input: "a" has hash code: 97

Input: "f" has hash code: 102

Input: "g" has hash code: 103

Input: "h" has hash code: 104

Input: "r" has hash code: 114

4. Create a Color class that has red, green, and blue values. Two colors are considered equal if their RGB values are the same

```
import java.util.Objects;

public class Color {
    private int red;
    private int green;
    private int blue;

    public Color(int red, int green, int blue) {
        this.red = red;
        this.green = green;
        this.blue = blue;
    }

    public int getRed() {
        return red;
    }

    public int getGreen() {
        return green;
    }

    public int getBlue() {
        return blue;
    }

    @Override
    public boolean equals(Object obj) {
        if (this == obj) return true;
        if (obj == null || getClass() != obj.getClass()) return false;
        Color color = (Color) obj;
        return red == color.red && green == color.green && blue == color.blue;
    }

    @Override
    public int hashCode() {
        return Objects.hash(red, green, blue);
    }

    @Override
    public String toString() {
        return "Color{" +
            "red=" + red +
            ", green=" + green +
            ", blue=" + blue +
            "}";
    }
}
```

```
public static void main(String[] args) {
    Color color1 = new Color(255, 0, 0);
    Color color2 = new Color(255, 0, 0);
    Color color3 = new Color(0, 255, 0);

    System.out.println("color1: " + color1);
    System.out.println("color2: " + color2);
    System.out.println("color3: " + color3);

    System.out.println("color1 equals color2: " + color1.equals(color2));
    System.out.println("color1 equals color3: " + color1.equals(color3));

    System.out.println("color1 hashCode: " + color1.hashCode());
    System.out.println("color2 hashCode: " + color2.hashCode());
    System.out.println("color3 hashCode: " + color3.hashCode());
}
```



```
color1: Color{red=255, green=0, blue=0}
color2: Color{red=255, green=0, blue=0}
color3: Color{red=0, green=255, blue=0}
color1 equals color2: true
color1 equals color3: false
color1 hashCode: 274846
color2 hashCode: 274846
color3 hashCode: 37696
```