



Course Title: Object - Oriented Programming II LAB

Course Code: CSE 2110

<u>Submitted by</u>	<u>Submitted to</u>
Name : Md Shabab Sayem	Name : SHOVON MANDAL
ID : 11230121166	Designation :
Department : CSE	Lecturer
Section : 3j	Dept. of Computer Science Engineering
	Northern University of Business & Technology, Khulna

Submission Date: 16/11/24

Problem 1: Integer array.

Question:

Write a Java program to find the sum of all elements in an integer array.

Objective:

We can learn how to create an array and use the scanner function.

Lab Work:

```
import java.util.Scanner;

public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = sc.nextInt();
        int[] numbers = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            numbers[i] = sc.nextInt();
        }
        int sum = 0;
        for (int number : numbers) {
            sum += number;
        }
        System.out.println("The sum of all elements in the array is: " + sum);
    }
}
```

Output:

```
Enter the number of elements in the array: 5
Enter the elements of the array:
1 2 3 4 5 4
The sum of all elements in the array is: 15
```

Result analysis:

This code successfully scans the array limit and takes input of the array elements and prints the sum of arrays.

Problem 2:

Question:

Write a Java program to find the largest element in an integer array.

Objective:

Can learn about the algorithm that determines the maximum number in an array.

Lab work:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = sc.nextInt();

        int[] numbers = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            numbers[i] = sc.nextInt();
        }
        int largest = numbers[0];
        for (int i = 1; i < n; i++) {
            if (numbers[i] > largest) {
                largest = numbers[i];
            }
        }
        System.out.println("The largest element in the array is: " + largest);
    }
}
```

Output:

```
Enter the number of elements in the array: 4
Enter the elements of the array:
3 5 9 2
The largest element in the array is: 9
```

Result analysis: This code successfully scans the elements of array and performs algorithm to print the largest of them.

Problem 3:

Question:

Write a Java program to count the occurrences of a specific element in an array.

Object:

Can learn how to print the number of occurrence by using if condition inside a for loop.

Lab work:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = sc.nextInt();
        int[] numbers = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            numbers[i] = sc.nextInt();
        }
        System.out.print("Enter the element to count occurrences of: ");
        int target = sc.nextInt();
        int count = 0;
        for (int num : numbers) {
            if (num == target) {
                count++;
            }
        }
        System.out.println("The element " + target + " occurs " + count + " times in the array.");
    }
}
```

Output:

```
Enter the number of elements in the array: 6
Enter the elements of the array:
1 2 2 3 2 4
Enter the element to count occurrences of: 2
The element 2 occurs 3 times in the array.
```

Result analysis:

This program successfully performs the job of counting same number.

Problem 4:

Question:

Write a Java program to reverse the elements in an array.

Objective:

We can learn the algorithm to reverse the elements of array.

Lab work:

```
import java.util.Scanner;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = sc.nextInt();
        int[] numbers = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            numbers[i] = sc.nextInt();
        }
        for (int i = 0; i < n / 2; i++) {
            int temp = numbers[i];
            numbers[i] = numbers[n - i - 1];
            numbers[n - i - 1] = temp;
        }
        System.out.println("Reversed array:");
        for (int number : numbers) {
            System.out.print(number + " ");
        }
        System.out.println();
    }
}
```

Output:

```
Enter the number of elements in the array: 5
Enter the elements of the array:
10 20 30 40 50
Reversed array:
50 40 30 20 10
```

Result analysis:

This program allows us to input an array of integers and then reverses and print the array.

Problem 5:

Question:

Write a Java program to check if an array is sorted in ascending order.

Objective:

We can learn how to check if a array is sorted or not and learn about Boolean data type

Lab work:

```
import java.util.Scanner;

public class Main{
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter the number of elements in the array: ");
        int n = sc.nextInt();
        int[] numbers = new int[n];
        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < n; i++) {
            numbers[i] = sc.nextInt();
        }
        boolean isSorted = true;
        for (int i = 0; i < n - 1; i++) {
            if (numbers[i] > numbers[i + 1]) {
                isSorted = false;
                break;
            }
        }
        if (isSorted) {
            System.out.println("The array is sorted in ascending order.");
        } else {
            System.out.println("The array is not sorted in ascending order.");
        }
    }
}
```


Output:

```
Enter the number of elements in the array: 5
Enter the elements of the array:
1 2 3 4 5
The array is sorted in ascending order.
```

Result discussion:

This program successfully checks if the array is in ascending order or not.

Problem 6:

Question:

Write a Java program to merge two integer arrays into a single array.

Objective:

We can learn about algorithms to merge arrays.

Lab work:

```
import java.util.Scanner;

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

        System.out.print("Enter number of elements in the first array: ");
        int n1 = sc.nextInt();
        int[] arr1 = new int[n1];
        System.out.println("Enter elements of the first array:");
        for (int i = 0; i < n1; i++) {
            arr1[i] = sc.nextInt();
        }

        System.out.print("Enter number of elements in the second array: ");
        int n2 = sc.nextInt();
        int[] arr2 = new int[n2];
        System.out.println("Enter elements of the second array:");
        for (int i = 0; i < n2; i++) {
            arr2[i] = sc.nextInt();
        }

        int[] mrgArr = new int[n1 + n2];
        for (int i = 0; i < n1; i++) {
            mrgArr[i] = arr1[i];
        }
        for (int i = 0; i < n2; i++) {
            mrgArr[n1 + i] = arr2[i];
        }
        System.out.println("Merged array:");
        for (int i : mrgArr) {
            System.out.print(i + " ");
        }
        System.out.println();
    }
}
```

Output:

```
Enter number of elements in the first array: 3
Enter elements of the first array:
1 3 5
Enter 3number of elements in the second array: 3
Enter elements of the second array:
2 4 6
Merged array:
1 3 5 2 4 6
```

Result discussion:

This program successfully prints out all the merged arrays.

Problem: 7

Question:

Write a Java program to remove duplicate elements from an integer array.

Objective:

Prompt the user to enter the number of elements and the elements themselves for the array. Sort the array to make it easier to identify and remove duplicates. Iterate through the sorted array and copy only unique elements to a temporary array.

Lab work:

```
import java.util.Scanner;
import java.util.Arrays;
public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] array = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }
        Arrays.sort(array);

        int[] tempArray = new int[n];
        int j = 0;
        for (int i = 0; i < n - 1; i++) {
            if (array[i] != array[i + 1]) {
                tempArray[j++] = array[i];
            }
        }
        tempArray[j++] = array[n - 1];
        System.out.println("Array without duplicates:");
        for (int i = 0; i < j; i++) {
            System.out.print(tempArray[i] + " ");
        }
        System.out.println();
    }
}
```

Output:

```
Enter number of elements: 6
Enter elements:
1 2 2 3 4 4
Array without duplicates:
1 2 3 4
```

Result discussion:

This program successfully removes duplicate elements from an array and prints the unique elements.

Problem 8:

Question:

Write a Java program to find the second largest element in an array.

Objective:

We can learn the code for printing the second largest element.

Lab work:

```
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        System.out.print("Enter number of elements: ");

        int n = sc.nextInt();
        int[] array = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }
        int largest = Integer.MIN_VALUE;

        int sL = Integer.MIN_VALUE;
        for (int i = 0; i < n; i++) {
            if (array[i] > largest) {
                sL = largest;
                largest = array[i];
            } else if (array[i] > sL && array[i] != largest) {
                sL = array[i];
            }
        }

        System.out.println("Second largest element: " + sL);
    }
}
```

Output:

```
Enter number of elements: 5
Enter elements:
7 5 8 9 6
Second largest element: 8
```

Result discussion:

This approach ensures that the program efficiently finds the second largest element in a single pass through the array.

Problem 9:

Question:

Write a Java program to rotate an array to the right by one position.

Objective:

This program shows how to make an algorithm of rotating array.

Lab work:

```
import java.util.Scanner;

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

        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] array = new int[n];
        System.out.println("Enter elements:");

        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }
        int IE = array[n - 1];
        for (int i = n - 1; i > 0; i--) {
            array[i] = array[i - 1];
        }

        array[0] = IE;

        System.out.println("Rotated array:");
        for (int i : array) {
            System.out.print(i + " ");
        }
        System.out.println();
    }
}
```

Output:

```
Enter number of elements: 4
Enter elements:
10 20 30 40
Rotated array:
40 10 20 30
```

Result discussion:

This program demonstrates rotating an array of 3 index.

Problem 10:

Question: Write a Java program to find a pair of elements in an array that sum up to a specific number.

Objective:

We can learn how to use Break statement, Boolean function from this code.

Lab work:

```
import java.util.Scanner;

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

        System.out.print("Enter number of elements: ");
        int n = sc.nextInt();
        int[] array = new int[n];
        System.out.println("Enter elements:");
        for (int i = 0; i < n; i++) {
            array[i] = sc.nextInt();
        }
        System.out.print("Enter target sum: ");
        int Sum = sc.nextInt();
        boolean pairFound = false;
        for (int i = 0; i < n - 1; i++) {
            for (int j = i + 1; j < n; j++) {
                if (array[i] + array[j] == Sum) {
                    System.out.println("Pair found: (" + array[i] + ", " + array[j] + ")");
                    pairFound = true;
                    break;
                }
            }
        }
        if (pairFound) {
            break;
        }
    }

    if (!pairFound) {
        System.out.println("No pair found with the given sum.");
    }

}
}
```

Output:

```
Enter number of elements: 5
Enter elements:
1 2 3 4 5
Enter target sum: 7
Pair found: (2, 5)
```

Result discussion:

This program successfully prints the target sum from given array elements.