



جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم
قسم الانظمة الطبية الذكية

Lab: (3)

❖ Arrays Part II

Subject: Computer Programming (I)

Level: First

Lecturer: Asst. Lect. Ali Saleem Haleem



Ex: program creates a 1D array of integers, initializes it with some values, and then prints each element of the array using a method called printArray().

```
public class PrintArray1D {
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5}; // Sample 1D array

        System.out.println("Printing 1D Array:");
        printArray(array);
    }

    // Method to print a 1D array
    public static void printArray(int[] arr) {
        for (int i = 0; i < arr.length; i++) {
            System.out.print(arr[i] + " ");
        }
        System.out.println();
    }
}
```

Output

Printing 1D Array:
1 2 3 4 5



EX: program, instead of iterating over the array manually, we use `Arrays.toString()` method from the `java.util.Arrays` class to convert the array into a string representation, which is then printed directly using `System.out.println()`. This approach simplifies the code and achieves the same result.

```
import java.util.Arrays;

public class PrintArray1D {
    public static void main(String[] args) {
        int[] array = {1, 2, 3, 4, 5}; // Sample 1D array

        System.out.println("Printing 1D Array:");
        System.out.println(Arrays.toString(array));
    }
}
```

Read one-dimensional arrays:

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

        System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();

        int[] array = new int[size];

        System.out.println("Enter the elements of the array:");
        for (int i = 0; i < size; i++) {
            array[i] = scanner.nextInt();
        }

        System.out.println("The elements of the array are:");
        for (int i = 0; i < size; i++) {
            System.out.print(array[i] + " ");
        }
    }
}
```



Processing 1D-Arrays

Processing one-dimensional arrays typically involves performing various operations on the elements stored in the array. Some common operations include:

1. Traversal

```
public class Main {
    public static void main(String[] args) {
        // Defining the array
        int[] array = {1, 2, 3, 4, 5};

        // Traversing the array and printing its elements
        System.out.println("The elements of the array are:");
        for (int i = 0; i < array.length; i++) {
            System.out.println(array[i]);
        }
    }
}
```

2. Summation

```
public class Main {
    public static void main(String[] args) {
        // Defining the array
        int[] array = {1, 2, 3, 4, 5};

        // Calculating the sum of array elements
        int sum = 0;
        for (int i = 0; i < array.length; i++) {
            sum += array[i];
        }
        System.out.println("Sum of array elements: " + sum);
    }
}
```



3. Finding Maximum or Minimum

```
public class Main {
    public static void main(String[] args) {
        // Defining the array
        int[] array = {1, 2, 3, 4, 5};
        // Finding the maximum element in the array
        int max = array[0];
        for (int i = 1; i < array.length; i++) {
            if (array[i] > max) {
                max = array[i];
            }
        }
        System.out.println("Maximum element: " + max);
    }
}
```

4. Searching

```
public class Main {
    public static void main(String[] args) {
        // Defining the array
        int[] array = {5, 2, 7, 1, 3};

        // Searching for a specific value in the array
        int searchValue = 3;
        boolean found = linearSearch(array, searchValue);
        if (found) {
            System.out.println("Value " + searchValue + " found in the
array.");
        } else {
            System.out.println("Value " + searchValue + " not found in the
array.");
        }
    }

    // Linear Search implementation
    public static boolean linearSearch(int[] array, int searchValue) {
        for (int i = 0; i < array.length; i++) {
            if (array[i] == searchValue) {
                return true;
            }
        }
        return false;
    }
}
```



5. Sorting

Example (Ascending Order using Bubble Sort):

```
public class Main {
    public static void main(String[] args) {
        // Defining the array
        int[] array = {5, 2, 7, 1, 3};

        // Sorting the array using Bubble Sort
        for (int i = 0; i < array.length - 1; i++) {
            for (int j = 0; j < array.length - 1 - i; j++) {
                if (array[j] > array[j + 1]) {
                    int temp = array[j];
                    array[j] = array[j + 1];
                    array[j + 1] = temp;
                }
            }
        }

        System.out.println("Sorted array:");
        for (int i = 0; i < array.length; i++) {
            System.out.print(array[i] + " ");
        }
    }
}
```