جامعة المستقبل
AL MUSTAQBAL UNIVERSITY

كلية العلوم

قسم الانظمة الطبية الذكية

Lecture: (2)

Arrays Part II

Subject: Computer Programming II
Level: First
Lecturer: Dr. Maytham N. Meqdad

Arrays Part II

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.

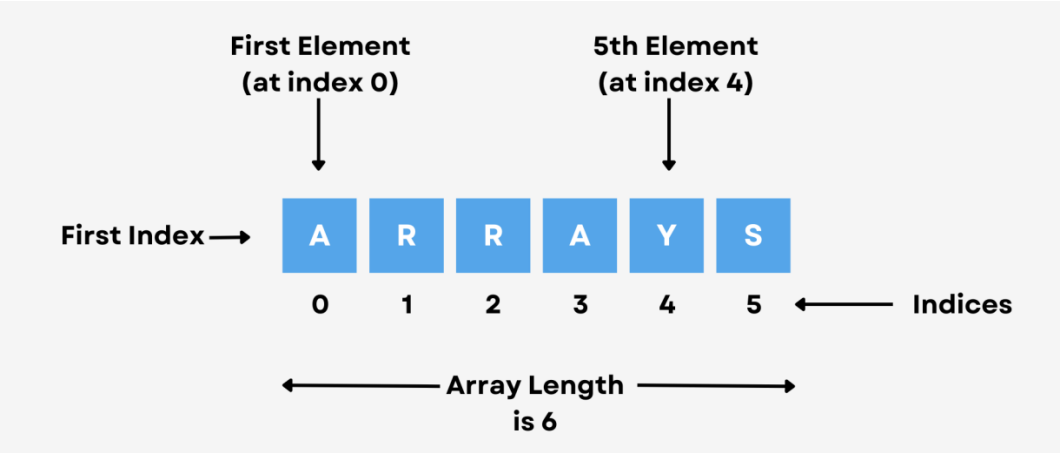
```
import java.util.Arrays;

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:

How do you read a one-dimensional array?
A single-dimensional array is a linear structure of elements in C, where each element stores data of the same data type and **can be accessed using an index**. For example, int numbers[5] = {1, 2, 3, 4, 5}; declares an integer array named 'numbers' with a size of 5 elements, containing the values 1 to 5.



```
import java.util.Scanner;

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] + " ");
        }
    }
}
```

1. **Import Statement**:

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

This line imports the Scanner class from the java.util package. Scanner is a utility class in Java that allows us to read input from various sources like the keyboard.

2. **Main Class Declaration**:

```
public class Main {
```

This line declares a public class named Main. In Java, the main method, which is the entry point of a Java program, must reside inside a class.

3. **Main Method Declaration**:

```
public static void main(String[] args) {
```

This line declares the main method. It's the starting point of the program execution.

4. **Creating Scanner Object**:

```
Scanner scanner = new Scanner(System.in);
```

Here, we create a Scanner object named scanner, which will be used to read input from the standard input stream (System.in).

5. **Reading Array Size**:

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

This code prompts the user to enter the size of the array and reads the input provided by the user using the nextInt() method of the Scanner class. The entered size is stored in the variable size.

6. **Creating Array**:

```
int[] array = new int[size];
```

Here, we create an integer array named array with the size entered by the user.

7. **Reading Array Elements**:

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

This code prompts the user to enter the elements of the array one by one and reads each element using the nextInt() method. The elements are stored in the array using a for loop.

8. **Printing Array Elements**:

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

Finally, this code prints out the elements of the array using a for loop. It iterates through each element of the array and prints it followed by a space.

9. **Closing Scanner**:

```
scanner.close();
```

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**: Iterating through each element of the array.

Example:

```
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**: Calculating the sum of all elements in the array.

Example:

```
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**: Determining the maximum or minimum value in the array.

Example (Maximum):

```
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**: Searching for a specific value in the array.

Example:

```
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**: Sorting the elements of the array in ascending or descending order.

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] + " ");
        }
    }
```