

# Sri Lanka Institute of Information Technology

## Faculty of Computing

IT1120 - Introduction to Programming

Dr.Junius Anjana

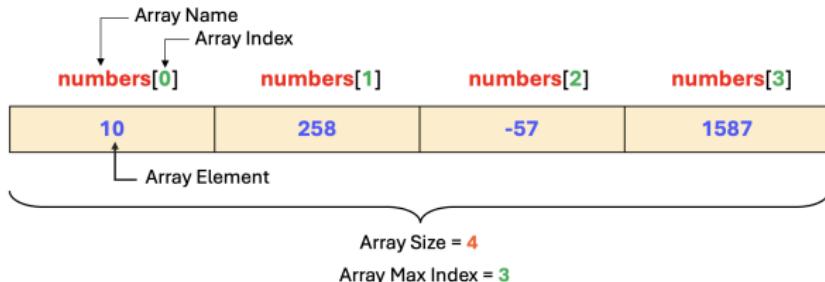
Year 01 and Semester 01

# Lecture 7

Arrays

# Arrays

- Array is a data structure which store data items of the same data type
- Array store all the data items in a continuous memory location
- Arrays can be initialized to any size
- Once initialized, **the size of an array** is fixed and cannot be changed
- If need to change the size of an array, you must create a new array with the desired size and copy the elements from old array to the new one



Reading an Array	
Array Index (location)	Array Element (value)
numbers[0]	10
numbers[1]	258
numbers[2]	-57
numbers[3]	1587

# Declaring and Allocating Arrays

## Declaration:

- This step specifies the **data type** of the array elements and the **name** of the array variable.

Declare an array using `[]`

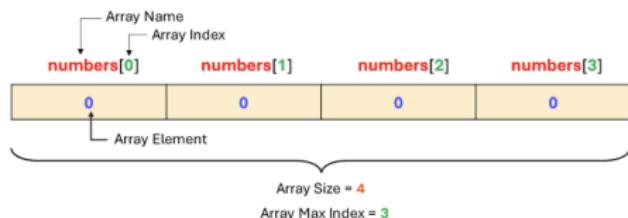
Example: `int numbers[];`

## Memory Allocation:

- This step allocates memory for the array elements.

Allocate an array using keyword **new** with `[]`

Example: `numbers = new int[4];`



## Combining Declaration and Allocation:

- You can combine declaration and allocation in a single statement:

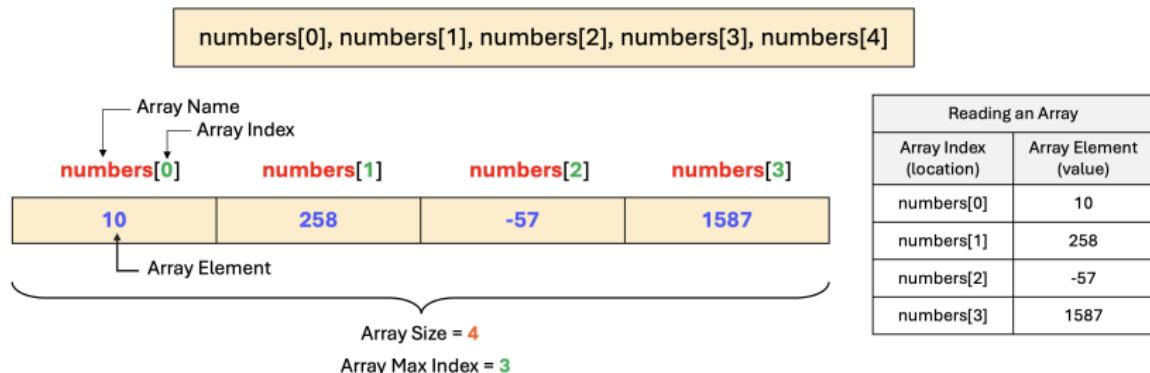
Example: `int numbers[] = new int[4];`

This allocates 4 integer elements of memory for the 'numbers' array.

Array name, like other variables can contain only letters, digits and underscore and cannot begin with a digit.

# Accessing elements in an array

- To refer to a specific element in an array, we need to specify array name followed by the **index** (subscript) of the array element location in square brackets
- Array index always starts with **Zero**, hence **first element** in the array is the **zero<sup>th</sup> index [0]**
- The last array element index always equal to: **Array Size – 1**



# Using arrays

Print the first element in the array:

```
System.out.print(numbers[0]);
```

Print the sum of first three elements in the array:

```
System.out.print(numbers[0] + numbers[1] + numbers[2]);
```

Add number 2 to the fourth element:

```
numbers[3] += 2;
```

# Declare and Initialize an Array

```
class Lecture7 {  
    public static void main(String[] args) {  
        int numbers[] = new int[5];  
        int i;  
        for (i = 0; i < numbers.length; i++) {  
            numbers[i] = 0;  
        }  
  
        System.out.println("Index\tValue");  
  
        //output content of array numbers in a tabular format  
        for (i = 0; i < numbers.length; i++) {  
            System.out.println(i + "\t\t" + numbers[i]);  
        }  
    }  
}
```

## Output

Index	Value
0	0
1	0
2	0
3	0
4	0

The **length** property in Java arrays is used to determine the **size of the array**.

## Initialize an **Array (1D)** using an **Initializer List**

```
class Lecture7 {  
    public static void main(String[] args) {  
        int numbers[] = {5, 12, 34, 56, 23};  
        int i;  
  
        System.out.println("Index\tValue");  
  
        //output content of array numbers in a tabular format  
        for (i = 0; i < numbers.length; i++) {  
            System.out.println(i + "\t\t" + numbers[i]);  
        }  
    }  
}
```

### Output

Index	Value
0	5
1	12
2	34
3	56
4	23

# Summing the elements of a Number Array (Integer / Whole Numbers)

```
import java.util.Scanner;
class Lecture7 {
    public static void main(String[] args) {
        Scanner value = new Scanner(System.in);

        int numbers[] = new int[5];
        int i, total = 0;
        for (i = 0; i < numbers.length; i++) {
            System.out.print("Enter a number : ");
            numbers[i] = value.nextInt();
        }

        //Calculate the total of array elements
        for (i = 0; i < numbers.length; i++)
            total += numbers[i];

        System.out.println("Sum of array elements is : " + total);
    }
}
```

## Output

```
Enter a number : 12
Enter a number : -2
Enter a number : 5
Enter a number : 3
Enter a number : 1
Sum of array elements is : 19
```



# Example of a **Float Array** (Real Numbers / Decimals)

```
import java.util.Scanner;
class Lecture7 {
    public static void main(String[] args) {
        Scanner value = new Scanner(System.in);
        float income[] = new float[7];
        float total = 0;
        int i;

        for (i = 0; i < income.length; i++) {
            System.out.print("Enter the daily income : ");
            income[i] = value.nextFloat();
        }

        //Calculate the total income
        for (i = 0; i < income.length; i++)
            total += income[i];

        System.out.println("Total income is " + total);
    }
}
```

Write a Java program to do the following:

- Declare a float array called **income** of size 7
- Prompt the user to input the daily income of a seller within a week and store them in the array
- Display the total income of the relevant week

## Output

```
Enter the daily income : 1500.00
Enter the daily income : 1000.00
Enter the daily income : 2500.00
Enter the daily income : 2000.00
Enter the daily income : 500.00
Enter the daily income : 500.00
Enter the daily income : 1200.00
Total income is 9200.0
```

# Example of a Char Array (Characters)

```
import java.util.Scanner;
class Lecture7 {

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

        char grade[] = new char[5];
        int i;

        for (i = 0; i < grade.length; i++) {
            System.out.print("Enter the grade of student " + (i + 1) + ": ");
            grade[i] = value.next().charAt(0);
        }
        System.out.println();
        //Display the array element
        for (i = 0; i < grade.length; i++)
            System.out.println("Grade of Student No. " + (i + 1) + " is " + grade[i]);
    }
}
```

Write a Java program to do the following:

- a) Declare a character array called **grade** of size 5
- b) Prompt the user to input grade of English module and store them in the array accordingly
- c) Display the student grades

## Output

```
Enter the grade of student 1: A
Enter the grade of student 2: A
Enter the grade of student 3: B
Enter the grade of student 4: C
Enter the grade of student 5: A
```

```
Grade of Student No. 1 is A
Grade of Student No. 2 is A
Grade of Student No. 3 is B
Grade of Student No. 4 is C
Grade of Student No. 5 is A
```

# Exercises

- ① Write a Java program to do the following:
  - Declare an integer array called **counts** with 10 elements
  - Initialize all elements to **-1**
  - Read and store 10 numbers each of which is between 10 to 100
  - Find the maximum number from the stored numbers.
- ② Write a Java program to do the following:
  - Declare a double array called salary of size 8
  - Prompt the user to input ten salaries of employees from the keyboard and store in the array
  - Display the salaries stored in the array
  - Count and display the number of employees in the following three salary ranges

salary < 10,000.00

10,000.00 <= salary <= 50,000

salary > 50,000

# Exercises

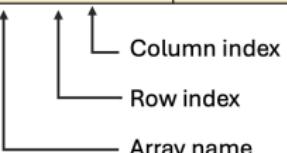
- ③ Write a Java program to do the following:
- Declare a character array called **shirts** of size 10
  - Prompt the user to input the sizes of the t-shirts that ten customers need to purchase and store in the array
  - The t-shirt sizes are small(**S**), medium(**M**) and large(**L**). Assume that the customer will purchase one t-shirt at a time
  - Display the quantity of each t-shirt sizes that need to be arranged

## Multidimensional Arrays

- Java language has arrays with multiple indexes (subscripts)
- These arrays are referred as multidimensional arrays
- An array with two subscripts is called **double-subscripted** or **Two-Dimensional (2D) array**

## Two-Dimensional Arrays

	Column 0	Column 1	Column 2	Column 3
Row 0	item[0][0]	item[0][1]	item[0][2]	item[0][3]
Row 1	item[1][0]	item[1][1]	item[1][2]	item[1][3]
Row 2	item[2][0]	item[2][1]	item[2][2]	item[2][3]



The diagram illustrates the structure of a 2D array. It shows a 3x4 grid of cells. The columns are labeled 'Column 0', 'Column 1', 'Column 2', and 'Column 3'. The rows are labeled 'Row 0', 'Row 1', and 'Row 2'. Each cell contains a text label 'item' followed by its row and column indices, such as 'item[0][0]' for the top-left cell. Below the grid, three arrows point upwards from the bottom right towards the labels 'Array name', 'Row index', and 'Column index'. The 'Array name' arrow points to the first 'item' in the grid. The 'Row index' arrow points to the label 'Row 2'. The 'Column index' arrow points to the label 'Column 3'.

- Two-dimensional arrays represent values in a tabular format arranged into rows and columns

## Initialize a 2D Array using an Initializer List

```
class Lecture7{  
    public static void main(String[] args) {  
  
        int values[][] = {{10, 2, 34, 84}, {50, 16, 7, 1}};  
  
        int i,j;  
        for(i = 0; i < values.length; ++i) {  
            for(j = 0; j < values[i].length; ++j) {  
                System.out.print(values[i][j] + " ");  
            }  
            System.out.print("\n");  
        }  
    }  
}
```

### Output

```
10 2 34 84  
50 16 7 1
```

## Initialize a 2D Array using User Inputs via Keyboard

```
import java.util.Scanner;  
  
public class Lecture7 {  
  
    public static void main(String[] args) {  
  
        Scanner value = new Scanner(System.in);  
  
        int numbers[][] = new int[2][3];  
  
        int i, j;  
  
        //Take keyboard inputs and store them in 2D array  
        for (i = 0; i < 2; i++) {  
            for (j = 0; j < 3; j++) {  
                System.out.print("Enter a number : ");  
                numbers[i][j] = value.nextInt();  
            }  
        }  
    }  
}
```

```
//Display the 2D array elements  
  
for (i = 0; i < 2; i++) {  
  
    for (j = 0; j < 3; j++) {  
        System.out.print(numbers[i][j] + " ");  
    }  
    System.out.println();  
}  
  
}  
}
```

**Output**

```
Enter a number : 10  
Enter a number : 8  
Enter a number : 6  
Enter a number : 12  
Enter a number : 1  
Enter a number : 25  
10 8 6  
12 1 25
```

# Summing the elements of a 2D Number Array (Integer / Whole Numbers)

```
import java.util.Scanner;  
  
public class Lecture7 {  
  
    public static void main(String[] args) {  
  
        Scanner value = new Scanner(System.in);  
  
        int numbers[][] = new int[2][3];  
  
        int i, j, total = 0;  
  
        for (i = 0; i < 2; i++) {  
  
            for (j = 0; j < 3; j++) {  
  
                System.out.print("Enter a number : ");  
  
                numbers[i][j] = value.nextInt();  
  
            }  
  
        }  
    }  
}
```

```
//Calculate the summation of array elements  
  
for (i = 0; i < 2; i++) {  
  
    for (j = 0; j < 3; j++) {  
  
        total = total + numbers[i][j];  
  
    }  
  
}  
  
System.out.println("Summation = " + total);  
}  
}
```

## Output

```
Enter a number : 10  
Enter a number : 2  
Enter a number : 6  
Enter a number : -5  
Enter a number : 100  
Enter a number : -50  
Summation = 63
```

# Thank You!