

Lecture: 3 Array

Sub: Object Oriented Programming (OOP)

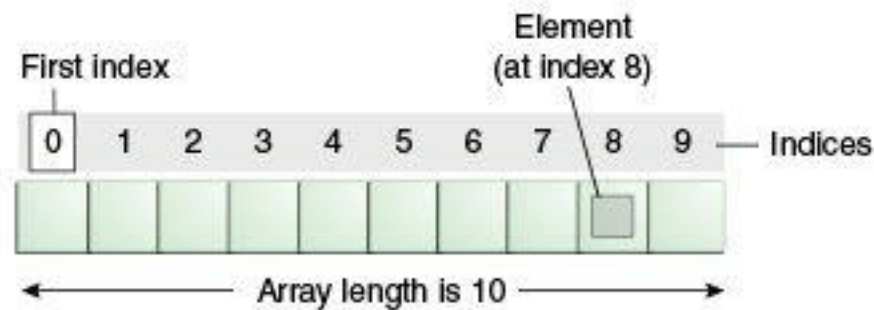
Code: CSE-1205

Array

- Array is a collection of similar type of elements that have contiguous memory location.
- **Java array** is an object that contains elements of similar data type. It is a data structure where we store similar elements. We can store only a fixed set of elements in a Java array.

Array

- Array in java is index based, first element of the array is stored at 0 index.



Advantage of Java Array

- **Code Optimization:** It makes the code optimized, we can retrieve or sort the data easily.
- **Random access:** We can get any data located at any index position.

Syntax to Declare an Array in java

- `dataType[] arr;` (or)
 - `dataType []arr;` (or)
 - `dataType arr[];`
-
- **Instantiation of an Array**
`arrayRefVar=new datatype[size];`

Declare arrays

- `byte[] anArrayOfBytes;`
- `short[] anArrayOfShorts;`
- `long[] anArrayOfLongs;`
- `float[] anArrayOfFloats;`
- `double[] anArrayOfDoubles;`
- `boolean[] anArrayOfBooleans;`
- `char[] anArrayOfChars;`
- `String[] anArrayOfStrings;`

Types of Array in java

- There are two types of array.
 - Single Dimensional Array
 - Multidimensional Array
- **single dimensional array**

```
class Testarray{
    public static void main(String args[]){
        int a[]=new int[5];//declaration and instantiation
        a[0]=10;//initialization
        a[1]=20;
        a[2]=70;
        a[3]=40;
        a[4]=50;
        //printing array
        for(int i=0;i<a.length;i++)//length is the property of array
            System.out.println(a[i]);
    }
}
```

Initializing array

```
// create an array of integers
```

```
anArray = new int[10];
```

```
anArray[0] = 100; // initialize first element
```

```
anArray[1] = 200; // initialize second element
```

Example of Array

```
class ArrayDemo {
    public static void main(String[] args) {
        // declares an array of integers
        int[] anArray;

        // allocates memory for 5 integers
        anArray = new int[5];

        // initialize first element
        anArray[0] = 100;
        // initialize second element
        anArray[1] = 200;
        // and so forth
        anArray[2] = 300;
        anArray[3] = 400;
        anArray[4] = 500;

        System.out.println("Element at index 0: " + anArray[0]);
        System.out.println("Element at index 1: " + anArray[1]);
        System.out.println("Element at index 2: " + anArray[2]);
        System.out.println("Element at index 3: " + anArray[3]);
        System.out.println("Element at index 4: " + anArray[4]);
    }
}
```

Instantiation and Initialization of Array

- `int a[]={33,3,4,5};` // declaration, instantiation and initialization

```
class Testarray1{
    public static void main(String args[]){

        int a[]={33,3,4,5}; // declaration, instantiation and
        initialization

        //printing array
        for(int i=0;i<a.length;i++) //length is the property of
        array
        System.out.println(a[i]);

    }
}
```

Passing Array to method

```
class Testarray2{
static void min(int arr[]){
int min=arr[0];
for(int i=1;i<arr.length;i++)
if(min>arr[i])
min=arr[i];

System.out.println(min);
}

public static void main(String args[]){

int a[]={33,3,4,5};
min(a);//passing array to method

}}
```

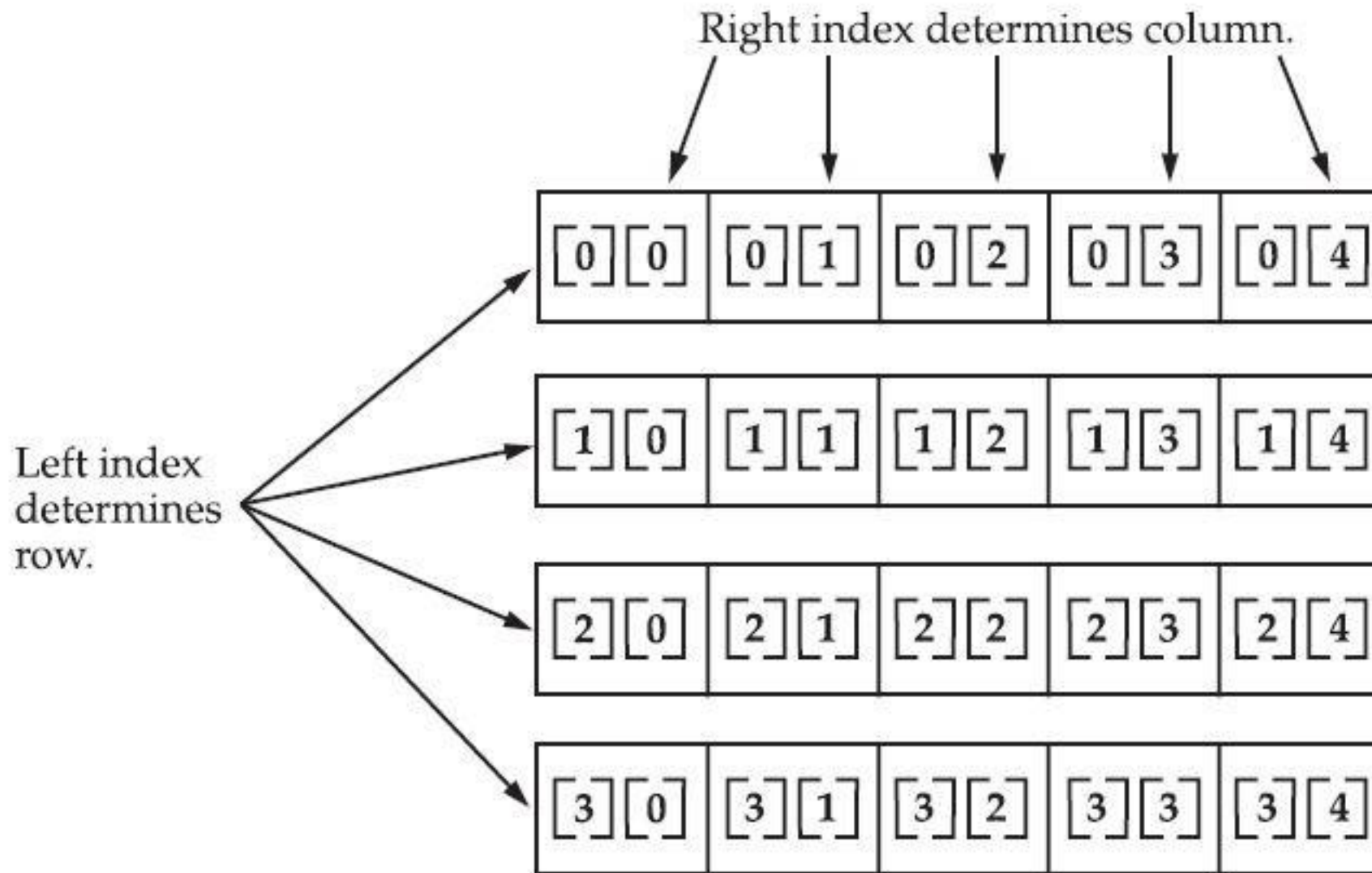
Multidimensional array

- data is stored in row and column based index (also known as matrix form).

Syntax to Declare Multidimensional Array

- `dataType[][] arrayRefVar; (or)`
- `dataType [][]arrayRefVar; (or)`
- `dataType arrayRefVar[][]; (or)`
- `dataType []arrayRefVar[];`

Multidimensional array



Given: `int twoD [] [] = new int [4] [5];`

Multidimensional array

- `int[][] arr=new int[3][3];` // 3 row and 3 column

initialize

- `arr[0][0]=1;`
- `arr[0][1]=2;`
- `arr[0][2]=3;`
- `arr[1][0]=4;`
- `arr[1][1]=5;`
- `arr[1][2]=6;`
- `arr[2][0]=7;`
- `arr[2][1]=8;`
- `arr[2][2]=9;`

Multidimensional array

- `long[][][] beans = new long[5][10][30];`
- `long[][][] beans = new long[3][][]; // Three two-dimensional arrays`

- `beans[0] = new long[4][];`
- `beans[1] = new long[2][];`
- `beans[2] = new long[5][];`

Example of Multidimensional array

```
class Testarray3{
public static void main(String args[]){

//declaring and initializing 2D array
int arr[][]={{1,2,3},{2,4,5},{4,4,5}};

//printing 2D array
for(int i=0;i<3;i++){
for(int j=0;j<3;j++){
System.out.print(arr[i][j]+" ");
}
System.out.println();
}

}}
```

Output: 1 2 3
2 4 5
4 4 5

Example of Multidimensional array

```
class TwoDAgain {  
public static void main(String args[]) {  
int twoD[][] = new int[4][];  
twoD[0] = new int[1];  
twoD[1] = new int[2];  
twoD[2] = new int[3];  
twoD[3] = new int[4];  
int i, j, k = 0;  
for(i=0; i<4; i++)  
for(j=0; j<i+1; j++) {  
twoD[i][j] = k;  
k++;  
}  
for(i=0; i<4; i++) {  
for(j=0; j<i+1; j++)  
System.out.print(twoD[i][j] + " ");  
System.out.println();  
}  
}  
}
```

output:
0
1 2
3 4 5
6 7 8 9

Initialization of Array using Method

- `double[] data = new double[50];`

```
import java.util.Arrays;
```

```
Arrays.fill(data, 1.0);
```

Copying Arrays

- The System class has an arraycopy method that you can use to efficiently copy data from one array into another
 - arraycopy(Object src, int srcPos, Object dest, int destPos, int length)

```
class ArrayCopyDemo {
    public static void main(String[] args) {
        char[] copyFrom = { 'd', 'e', 'c', 'a', 'f', 'f', 'e',
                           'e', 'e', 'a', 't', 'e', 'd' };
        char[] copyTo = new char[7];

        System.arraycopy(copyFrom, 2, copyTo, 0, 7);
        System.out.println(new String(copyTo));
    }
}
```

output:
caffee

Addition of 2 matrices

```
class Testarray5{
    public static void main(String args[]){
        //creating two matrices
        int a[][]={{1,3,4},{3,4,5}};
        int b[][]={{1,3,4},{3,4,5}};

        //creating another matrix to store the sum of two matrices
        int c[][]=new int[2][3];

        //adding and printing addition of 2 matrices
        for(int i=0;i<2;i++){
            for(int j=0;j<3;j++){
                c[i][j]=a[i][j]+b[i][j];
                System.out.print(c[i][j]+" ");
            }
            System.out.println();//new line
        }
    }
}
```

Output:2 6 8
6 8 10

Java Programs