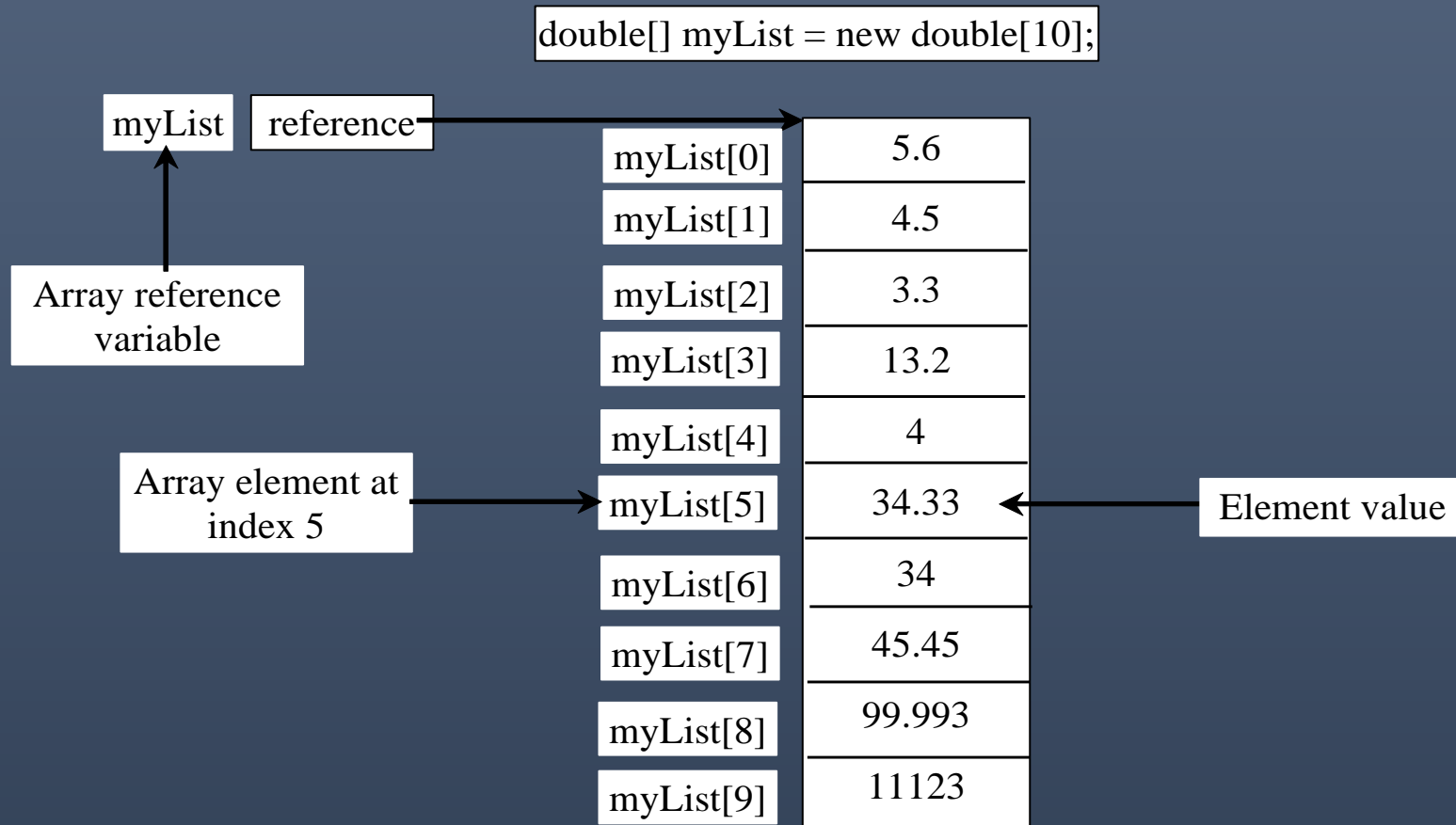# Java Array

Lecture - 5

Tokey Ahmmed

Lecturer, Dept. of CSE

Varendra University

tokey@vu.edu.bd

Introducing Arrays

- Array is a data structure that represents a collection of the same types of data.
- Arrays in Java are objects that can be treated just like other objects in the language

double[] myList = new double[10];

myList	reference		myList[0]	5.6
			myList[1]	4.5
Array reference variable			myList[2]	3.3
			myList[3]	13.2
			myList[4]	4
Array element at index 5			myList[5]	34.33	Element value
			myList[6]	34
			myList[7]	45.45
			myList[8]	99.993
			myList[9]	11123

Declaring Array Variables

❑Syntax 1:

```
datatype[] arrayRefVar;
```

- Example:

double[] myList;

❑Syntax 2:

```
datatype arrayRefVar[];
```
// This style is allowed, but not preferred

- Example:

double myList[];

Creating Arrays

❑ **Declaring and Creating in Two Steps**

```
double[] myList;
myList = new double[10];
```

❑ **Declaring and Creating in One Step**

```
double[] myList = new double[10];
```

The Length of an Array

- Once an array is created, its size is fixed. It cannot be changed.
- To find the size of an array:

double[] myList = new double[10];

      arrayRefVar.length

For example,

      myList.length returns 10

myList[0]	5.6
myList[1]	4.5
myList[2]	3.3
myList[3]	13.2
myList[4]	4
myList[5]	34.33
myList[6]	34
myList[7]	45.45
myList[8]	99.993
myList[9]	11123

Declaring, creating, initializing Using the Shorthand Notation

double[] myList = {1.9, 2.9, 3.4, 3.5};

This shorthand notation is equivalent to the following statements:

double[] myList = new double[4];

myList[0] = 1.9;

myList[1] = 2.9;

myList[2] = 3.4;

myList[3] = 3.5;

Initializing arrays with input values

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

        double myList[]=new double[5];
        Scanner input = new  Scanner(System.in);
          System.out.print("Enter " + myList.length + " values: ");
        for (int i = 0; i < myList.length; i++) {
                myList[i] = input.nextDouble();
         }
         for (int i = 0; i < myList.length; i++) {
                System.out.println(myList[i]);}
         }
     }
}
```

Initializing arrays with random values

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


        double myList[]=new double[5];
        for (int i = 0; i < myList.length; i++) {
                myList[i] = Math.random() * 100;
                System.out.println(myList[i]);
        }
    }
}
```

Enhanced for Loop (for-each loop)

In general, the syntax is

```
for (elementType value: arrayRefVar) {
  // Process the value
}
```

Output:

10,20,30,40,50,

```
public class Test{
public static void main(String args[]){
    int[] numbers ={10,20,30,40,50};
    for(int x : numbers ){
        System.out.print(x);
        System.out.print(",");
    }
  }
}
```

Sending an Array as a parameter to a Method

```java
public class ArrayMethod {
static public void add(int[] a)
    {
        int sum=0;
        for(int i=0;i<a.length;i++){
            sum=sum+a[i];
            System.out.println("array value:" + a[i]);        }
        System.out.println("summation:" + sum);
    }


public static void main(String[] args) {
    int a[]={ 2,3,5 } ;
        add(a);
}
 }
```

Output is:

array value:2
array value:3
array value:5
summation:10

Two-dimensional Arrays (Declaration)

// Declare array ref var

dataType[ ][ ] refVar;

// Create array and assign its reference to variable

refVar = new dataType[10][10];

// Combine declaration and creation in one statement

dataType[ ][ ] refVar = new dataType[10][10];

// Alternative syntax

dataType refVar[ ][ ] = new dataType[10][10];

Declaring, Creating, and Initializing Using Shorthand Notations

You can also use an array initializer to declare, create and initialize a two-dimensional array. For example,

```
int[][] array = {
    {1, 2, 3},
    {4, 5, 6},
    {7, 8, 9},
    {10, 11, 12}
};
```

Same as

```
int[][] array = new int[4][3];
array[0][0] = 1; array[0][1] = 2; array[0][2] = 3;
array[1][0] = 4; array[1][1] = 5; array[1][2] = 6;
array[2][0] = 7; array[2][1] = 8; array[2][2] = 9;
array[3][0] = 10; array[3][1] = 11; array[3][2] = 12;
```