

JAVA PROGRAMMING BASICS

Module 1: Java Overview

Training program

1. Java Fundamentals
2. Start programming with Java, create simple console application
3. Classification of Data Types
4. Primitive types in java
5. Control Flow Statements
6. Arrays

Module contents

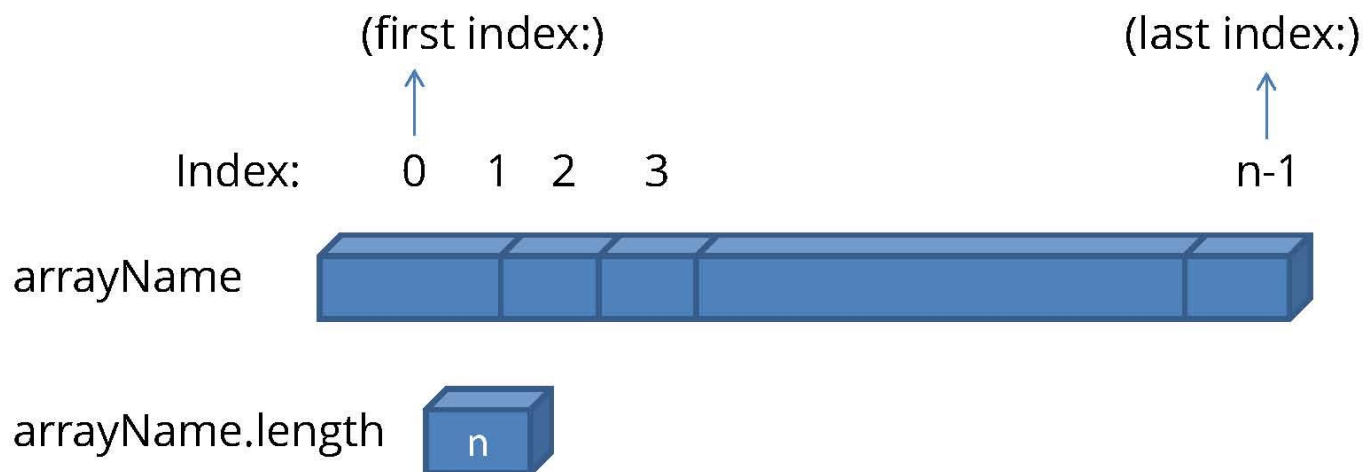
- Arrays
 - One dimensional arrays. Length of array
 - Working with Arrays and Array Elements
 - Multidimensional arrays
 - Resize an array
 - The Arrays Class
 - Array Manipulations: copying, equal check, search element, sort.

Module contents

- Arrays
 - One dimensional arrays. Length of array
 - Working with Arrays and Array Elements.
 - Multi - dimention arrays
 - Resize an array
 - The Arrays Class.
 - Array Manipulations: copying, equal check, search element, sort.

One dimensional arrays 1/2

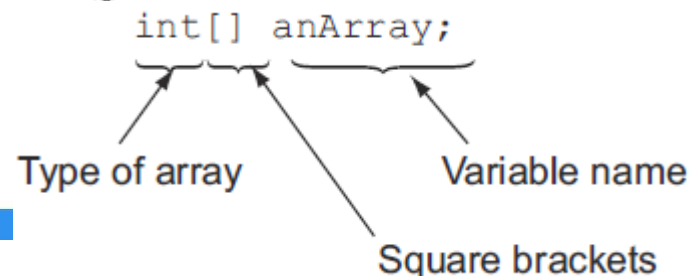
- An *array* is a container object that holds a fixed number of values of a single type. The length of an array is established when the array is created. After creation, its length is fixed.



In the Java *arrays* are objects, are dynamically created, and may be assigned to variables of type `Object`. All methods of class `Object` may be invoked on an array.

One dimensional arrays 2/2

- ← refers to null
1. **int**[] arr1; *// Declare an int array named marks*
 2. **int** arr2[]; *// Same as above, (not recommended)*
 3. arr1 = **new int**[5]; *// Allocate 5 elements*
 4. *// Declare and allocate a 100-element array*
 5. **int**[] arr3 = **new int**[100];
 6. *// Declare, allocate a 6-element array*
 7. **int**[] arr4 = {12, 25, 37, 44, 55, 66};
 8. *// size of array deduced from the number of items*



Length of array

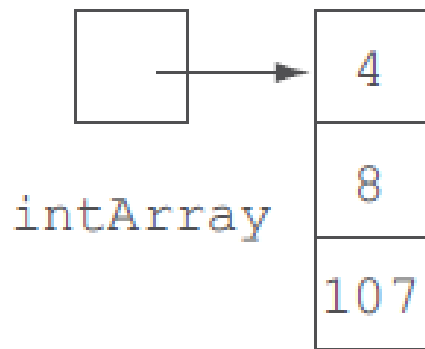
In Java, the length of array is kept in an *associated variable* called length and can be retrieved using "*arr.length*"

1. **int**[] arr = **new int**[100];
2. **int** num = arr.**length**; // *num is 100*
3. System.**out**.println(num);

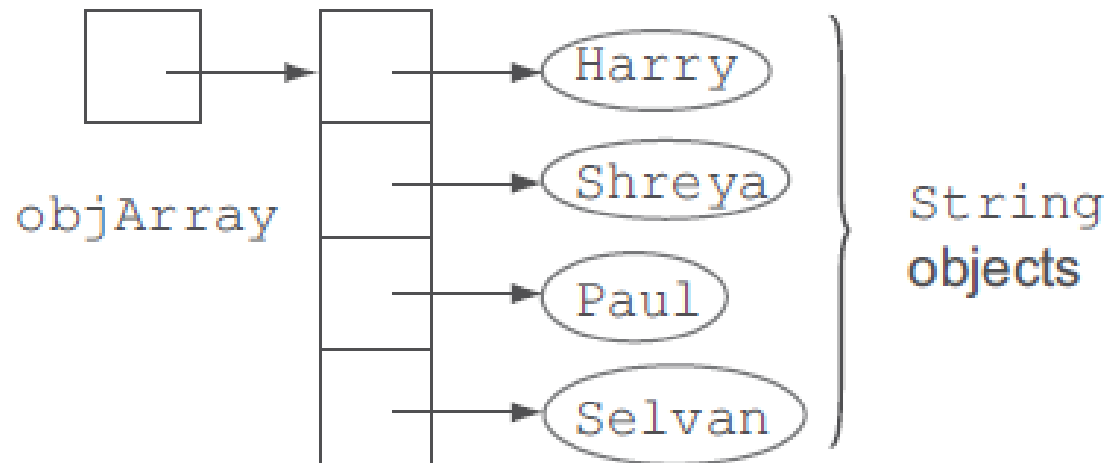
Array of primitive data vs array of objects

```
int[] intArray = {4,8,107};
```

```
String[] objArray = {"Harry", "Shreya", "paul", "Salvan"};
```



Array of primitive data



Array of objects

An array of primitives stores a collection of values that constitute the primitive values themselves. (With primitives, there are no objects to reference.)

An array of objects stores a collection of values, which are in fact heap-memory addresses or pointers.

Module contents

- Arrays
 - One dimensional arrays. Length of array.
 - Working with Arrays and Array Elements.
 - Multi - dimension arrays
 - Resize an array
 - The Arrays Class.
 - Array Manipulations: copying, equal check, search element, sort.

Working with arrays and array elements

ArraysDemo.java

for vs for-each (enhanced for) loop

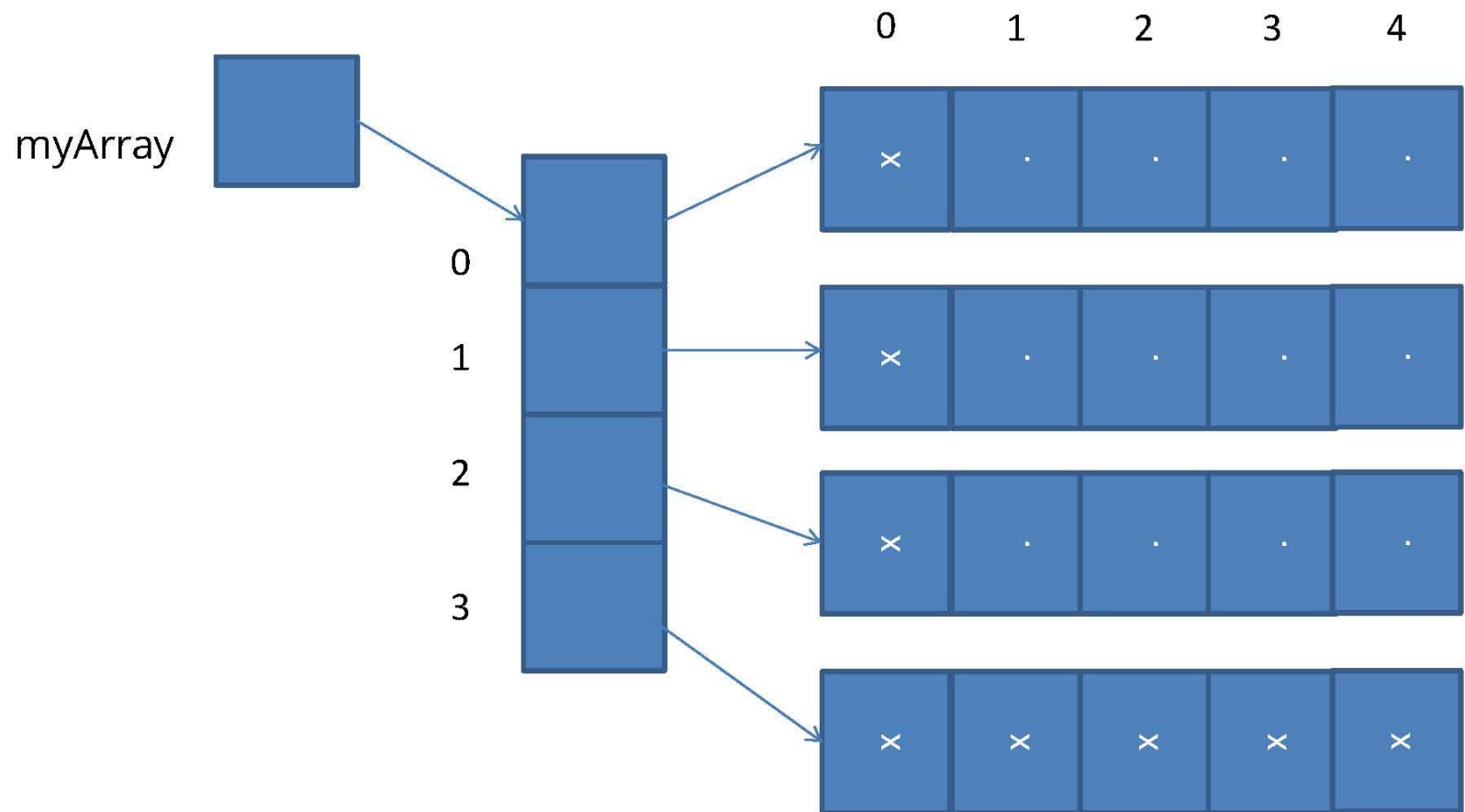
- The *enhanced for loop* can't be used to initialize an array and modify its elements.
- The *enhanced for loop* can't be used to delete the elements of a collection.
- The *enhanced for loop* can't be used to iterate over multiple collections or arrays in the same loop.

ForEachDemo.java

Module contents

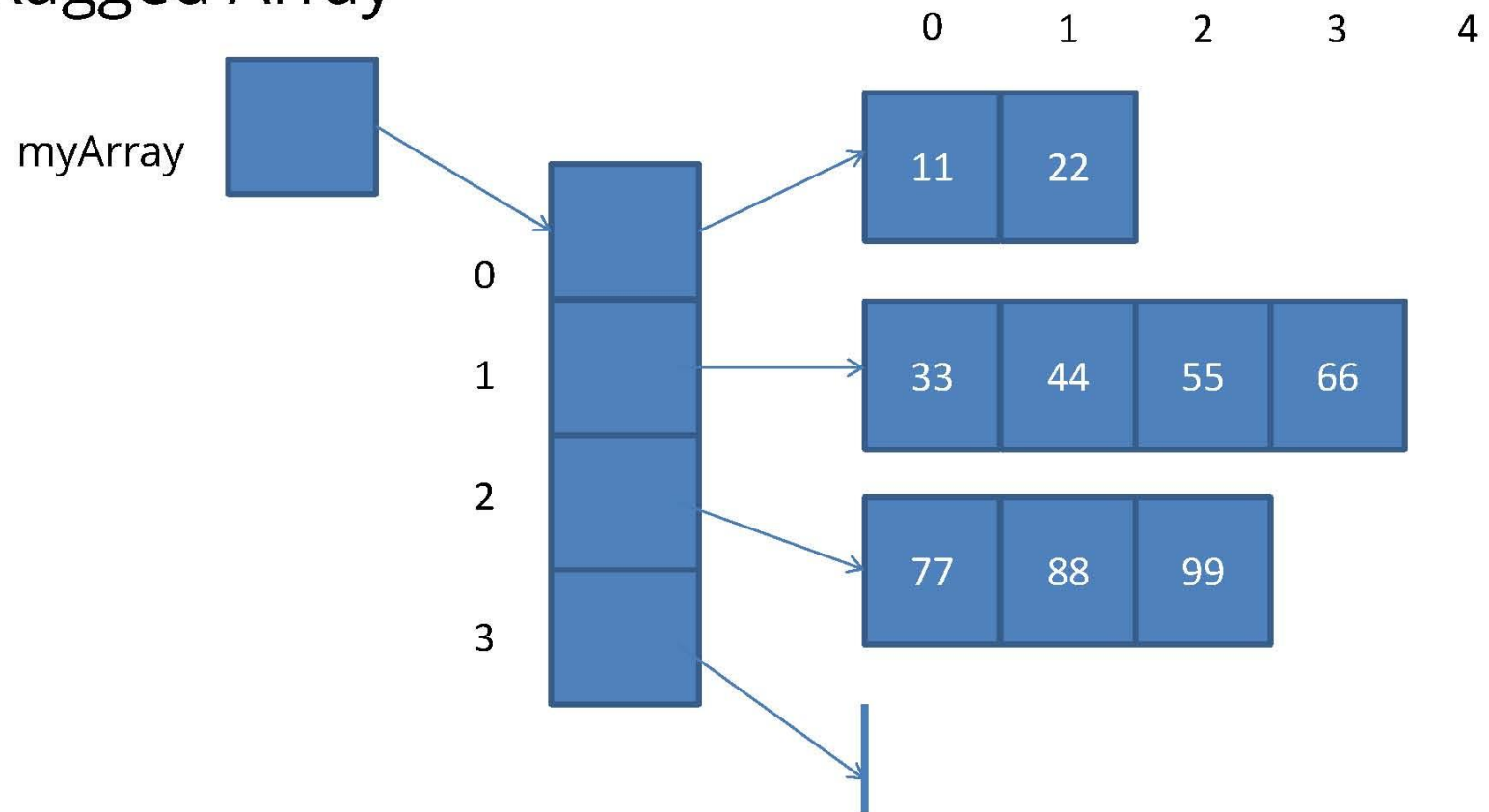
- Arrays
 - One dimensional arrays. Length of array.
 - Working with Arrays and Array Elements.
 - Multi - dimention arrays
 - Resize an array
 - The Arrays Class.
 - Array Manipulations: copying, equal check, search element, sort.

Multi - dimention arrays 1/3



Multi – dimention (Ragged) arrays 2/3

- Ragged Array



Multi - dimention arrays 3/3

1. **int**[][] marr = {{11,22},{33,44,55,66},{77,88,99},{}};
2. marr[1][2] = 123;
3. marr[2][2] = 555;
4. System.*out*.println(marr[0][0]);
5. System.*out*.println(marr[1][0]);
6. System.*out*.println(marr[0][1]);

Module contents

- Arrays
 - One dimensional arrays. Length of array.
 - Working with Arrays and Array Elements.
 - Multi - dimention arrays
 - **Resize an array**
 - The Arrays Class.
 - Array Manipulations: copying, equal check, search element, sort.

Resize an array

- You cannot resize an array.
- You can use the same reference variable to refer to an entirely new array, such as:
- **int**[] myArray = **new int**[6];
- myArray = **new int**[10];

Module contents

- Arrays
 - One dimensional arrays. Length of array.
 - Working with Arrays and Array Elements.
 - Multi - dimation arrays
 - Resize an array
 - The Arrays Class.
 - Array Manipulations: copying, equal check, search element, sort.

The Arrays Class 1/2

- Class Arrays providing static methods for common array manipulations:
 - **sort**(array) : Arranges array elements into increasing order.
 - **binarySearch**(array , element) : Determines whether an array contains a specific value and, if so, returns where the value is located.
 - **equal**(array) : Compares arrays.
 - **fill**(array , element) : Places Values into an array.
 - **toString**() : Converts array to String.

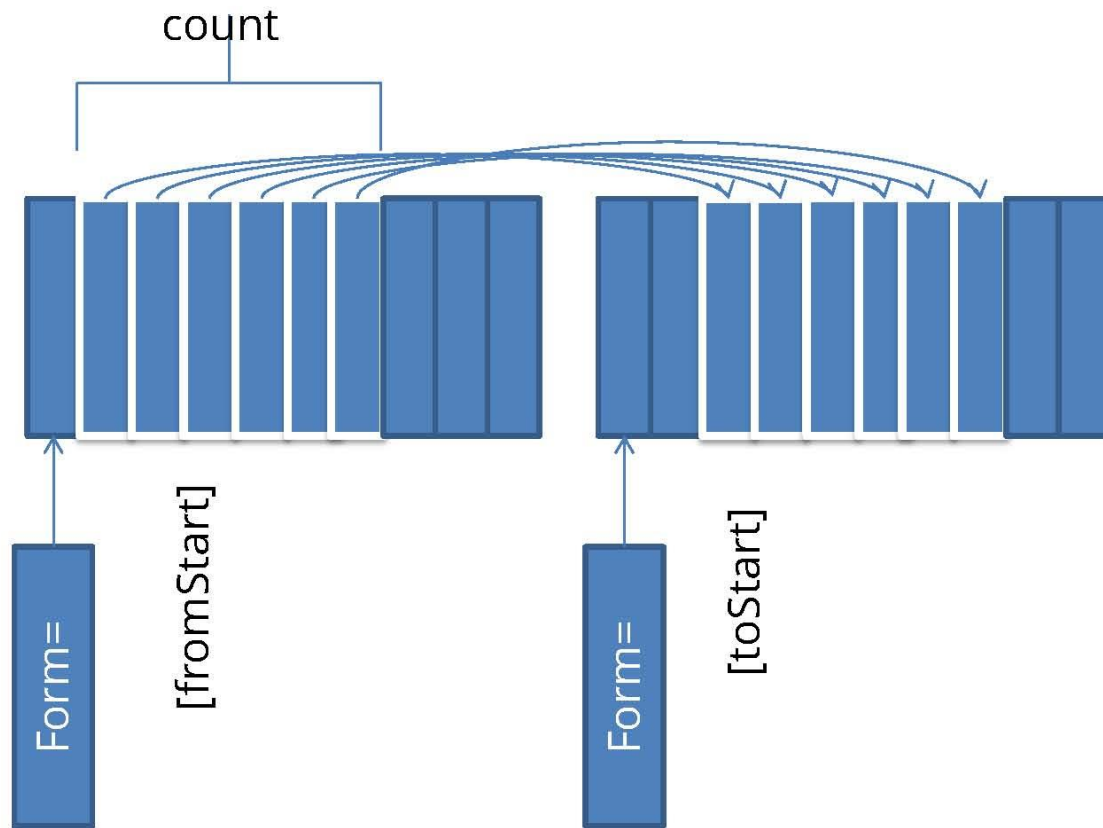
The Arrays Class 2/2

- We can copy arrays using **copyof** method of the class Arrays Or using class System's static arraycopy method.
- To use Arrays Class we import it by import `java.util.Arrays ;`

Module contents

- Arrays
 - One dimensional arrays. Length of array.
 - Working with Arrays and Array Elements.
 - Multi - dimation arrays
 - Resize an array
 - The Arrays Class.
 - Array Manipulations: copying, equal check, search element, sort.

Copying Arrays 2/2



Binary Search element

