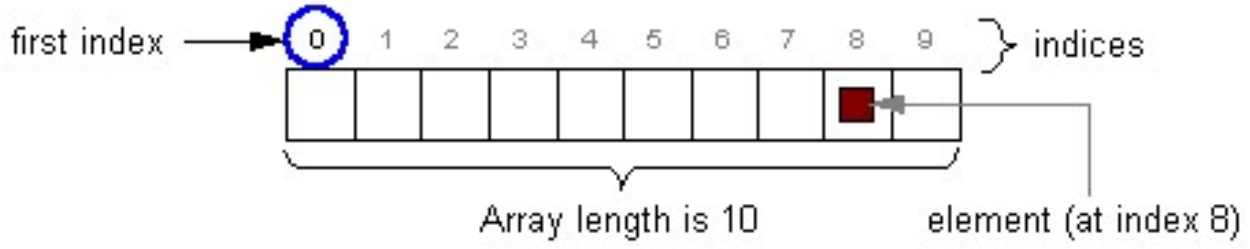

Arrays
Bok, Jong Soon
javaexpert@nate.com
https://github.com/swacademy/Core-Java

What is a array ?

- Is a structure that holds multiple values of the same type.
- The length of an array is established when the array is created (at runtime).
- After creation, an array is a fixed-length structure.



Declaring Arrays

- Group data objects of the same type.
- You access each individual value through an integer ***index***.
- Declare arrays of primitive or class types.

```
char s [] ;
Point p [] ;
char [] s;
Point [] p;
```

- Create space for a reference.
- An array is an object ; it is created with **`new`**.

Creating Arrays

- Use the **new** keyword to create an array object.

```
s = new char[20];
p = new Point[100];

p[0]= new Point();
p[1]= new Point();
```

Initializing Arrays

- Initialize an array element.
- Create an array with initial values:

```
1 String names[];
2 names = new String[3];
3 names[0] = "Sujan";
4 names[1] = "Michael";
5 names[2] = "John";
```

```
1 String names[] = {
2         "Sujan",
3         "Michael";
4         "John"
5 }
```

```
1 MyDate [] dates;
2 dates = new MyDate[3];
3 dates[0] = new MyDate(1978, 9, 15);
4 dates[1] = new MyDate(1986, 3, 5);
5 dates[2] = new MyDate(2003, 6, 28);
```

```
1 MyDate [] dates = {
2         new MyDate(1978, 9, 15),
3         new MyDate(1986, 3, 5),
4         new MyDate(2003, 6, 28)
5 }
```

Multi-Demensional Arrays

- Arrays of arrays:

```
int [][] twoDim = new int [4][];
twoDim[0] = new int[5];
twoDim[1] = new int[5];

int [][] twoDim = new int [][4]; //illegal
```

```
double[][] balances;
balances = new double[12][10];

int[][] matrix = {
    {1, 5, 2, 8},
    {2, 7, 1, 9},
    {7, 2, 6, 2},
    {9, 1, 4, 8}
};
```

ArrayEx.java

Multi-dimensional Arrays

1	5	2	8
2	7	1	9
7	2	6	2
9	1	4	8

Ragged Arrays

1			
1	2		
1	2	3	
1	2	3	4

Multi-Dimensional Arrays (Cont.)

- Non-rectangular arrays of arrays:

```
twoDim[0] = new int[2];
twoDim[1] = new int[4];
twoDim[2] = new int[6];
twoDim[3] = new int[8];
```

- Array of four arrays of five integers each:

```
int [][] twoDim = new int[4][5];
```

Arrays Bounds

- All array subscripts begin at 0:

```
int list[] = new int[10];
for (int i = 0; i < list.length; i++){
   System.out.println(list[i]);
}
```

Array Resizing

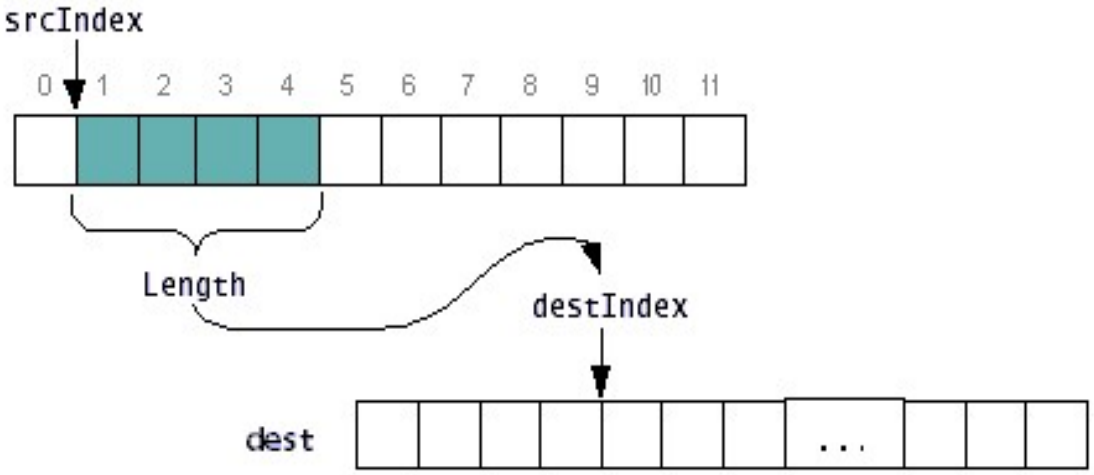
- Cannot resize an array.
- Can use the same reference variable to refer to an entirely new array:

```
int elements[] = new int[6];
elements = new int[10];
```

Copying Arrays

- The `System.arraycopy()` method.
- This method requires five arguments.

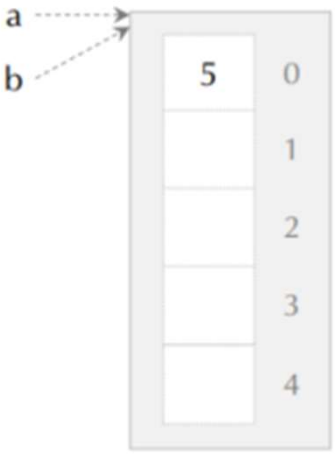
```
public static void arraycopy (Object source,
                              int srcIndex,
                              Object dest,
                              int destIndex,
                              int length)
```



Copying Arrays (Cont.)

- 배열 변수를 복사하는 경우, 참조변수만 복사되어 실제로는 두 변수가 동일한 배열을 참조합니다.

```
int[] a = new int[5];
int[] b = a;
b[0] = 5; // a[0]의 값도 변경된다.
```



- Arrays.copyOf() 메소드는 배열의 내용까지 복사합니다.

```
int[] copy = Arrays.copyOf(a, a.length);
```

- 배열 복사와 동시에 배열의 크기를 늘릴 수 있습니다.

```
int[] copy = Arrays.copyOf(a, 2 * a.length);
```

ArrayEx.java

Command Line Parameters

- Every Java program has a main method with a `String [] args` parameter.
- Indicates that the main method receives an array of strings, namely, the arguments specified on the command line.

Sorting an Array

- If you want to sort an array of numbers, you can use one of the `sort` methods in the `Arrays` class.
- This method uses a tuned version of the *QuickSort* algorithm that is claimed to be very efficient on most data sets.

Exercise : Using Arrays

- Level 1 Lab : Create a Basic Array

1. Create a class called BasicArray. In the main() method, declare two variables called thisArray and thatArray. They should be of type array of int.
2. Create an array of 10 int values that range from 1 to 10. Assign the reference of this third array to the variable thisArray.
3. Use a for() loop to print out all values of thisArray.
4. For each element of thisArray, set the value to be the factorial of the index value. Print out the values of the array.
5. Assign the reference of thisArray to the variable thatArray. Print out all the elements of thatArray.

Exercise : Using Arrays

- Level 1 Lab : Create a Basic Array (cont.)

6. Modify some of the elements or thisArray. Print out the value of thatArray.
7. Create an array of 20 int values. Assign the reference of the new array to the variable thatArray. Print out the values of thatArray.
8. Copy the values of thisArray into thatArray.
9. Print out the values of thatArray.
10. Change some values of thatArray. Print out both thisArray and thatArray.

Exercise : Using Arrays

- Level 2 Lab : Create Arrays of Arrays

1. Create a class called Array2D. In the main() method declare a variable called twoD, of type array of array of int. Make the first dimension equal to 4 ([4][]).
2. Create an array of element type int. The array should have four elements and be assigned to element [0] of the variable, twoD.
3. Write two nested for() loops that print out all the values of twoD. Arrange the output in a matrix format (System.out.print()).
4. Runtime error returns(NullPointerException).

Exercise : Using Arrays

- Level 2 Lab : Create Arrays of Arrays (cont.)

5. Create further arrays of `int`s containing five, six, and seven elements, respectively. Assign the reference to these to the elements [1],[2], and [3] of `twoD`, respectively. Ensure that the code for this is inserted before the nested `for()` loop described in step 3.
6. Assign a distinct nonzero value to each element of the `twoD` array(Hint- Use `Math.random()`).
7. Declare a variable called `oneD` of type array of `int`. Then, create an array of `int` that holds four elements, and assign the reference to this array to the first element of array `twoD` and array `oneD`. After making the assignmen, print out both the `oneD` and `twoD` arrays.