Core Java 8 and Development Tools

Lesson 10 : Arrays

Lesson Objectives

- After completing this lesson, participants will be able to
 - Understand the different types of Arrays
 - Implement one and multi dimensional arrays
 - Iterate arrays using loops
 - Use varargs
 - Work with java.util.Arrays





Copyright © Capgemini 2015, All Rights Reserved

This lesson discusses about how to work java arrays.

Lesson outline:

10.1: One dimensional array

10.2: Multidimensional array

10.3: Using varargs

10.4: Using Arrays class

10.5: Best Practices

Arrays

- Arrays are used to group elements of either of primitive or reference types
- Array in java is created as Object:
 - This object will help developers to find size of array
 - Using this object developers can manipulate array
 - Can be compared with null
- All elements of array of same type
- Array is a fixed-length data structure having zero-based indexing





Copyright © Capgemini 2015. All Rights Reserved

Arrays can be created for either primitive or reference type elements. Array in java is created as Object using new operator. Once array is created, individual elements can be accessed using index number enclosed in square brackets.

Arrays indexing is zero based, it means the first element of array start at index 0, second element is at 1, and so on. The last element of array is indexed as one minus size of an array.

Array size can be captured by using public final instance variable called length. This feature will avoid any runtime exceptions resulted due to out of bounds access.

Arrays

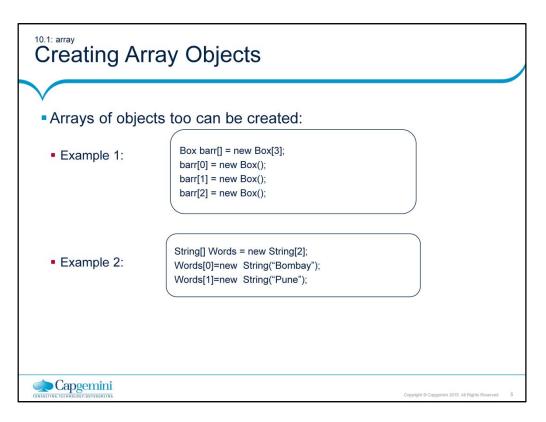
- A group of like-typed variables referred by a common name
- Array declaration and initialization:
 - int arr []; arr = new int[10];
 - int arr[] = $\{2,3,4,5\}$;
 - int twoDim [][] = new int[4][5];



Copyright © Capgemini 2015. All Rights Reserved

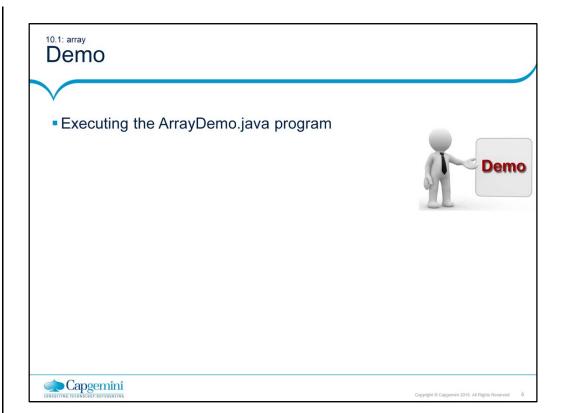
Syntax wherein the array is declared and initialized in the same statement:

```
strWords = { "quiet", "success", "joy", "sorrow", "java" };
```



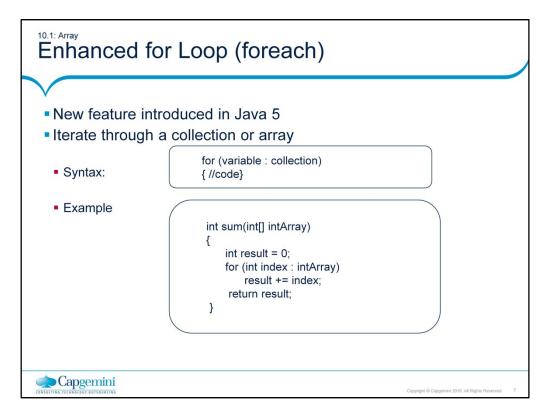
Use new operator or directly initialize an array. When you create an array object using new, all its slots are initialized for you (0 for numeric arrays, false for boolean, '\0' for character arrays, and null for objects).

Like single dimensional arrays, we can form multidimensional arrays as well. Multidimensional arrays are considered as array of arrays in java and hence can have asymmetrical arrays. See an example next.



We have seen how to pass parameters to program during compile time using parameter passing. One can pass parameters to a program at runtime too. The args parameter (a String array) in main() receives command line arguments.

```
class ArrayDemo {
   int intNumbers[];
   ArrayDemo(int i) {
     intNumbers = new int[i];
   void populateArray() {
       for(int i = 0; i < intNumbers.length; ++i) intnumbers[i] = i;
   void displayContents() {
       for(int i = 0; i <intNumbers.length; ++i)</pre>
           System.out.println("Number " + i + ": " + intNumbers[i]);
   }
   public static void main(String[] args)
      //Accepting array length as command line argument.
       int intArg = Integer.parseInt(args[0]);
       ArrayDemo ad = new ArrayDemo(intArg);
       ad.displayContents();
       ad.populateArray();
       ad.displayContents();
  }}
```



Enhanced for loop works with collections and arrays. Notice the difference between the old code where the three standard steps of initialization, conditional check and re-initialization are explicitly required to be mentioned.

Old code

```
public class OldForArray {
    public static void main(String[] args){
        int[] squares = {0,1,4,9,16,25};
        for (int i=0; i< squares.length; i++){
            System.out.printf("%d squared is %d.\n",i, squares[i]);
        }
    }
}</pre>
```

New Code

```
public class NewForArray {
    public static void main(String[] args) {
        int j = 0;
        int[] squares = {0, 1, 4, 9, 16, 25};
        for (int i : squares) {
            System.out.printf("%d squared is %d.\n", j++, i);
        }
    }
}
```

10.2 : Method with Variable Argument Lists

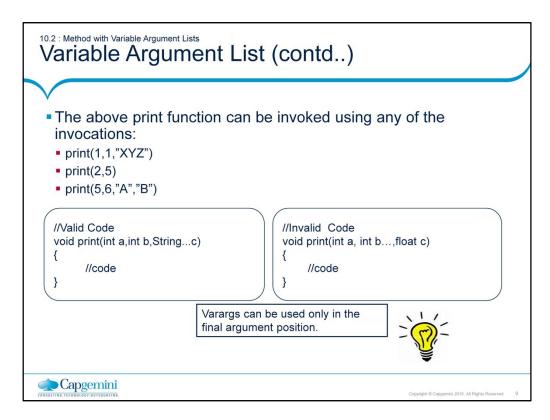
Variable Argument List

- New feature added in J2SE5.0
- Allows methods to receive unspecified number of arguments
- An argument type followed by ellipsis(...) indicates variable number of arguments of a particular type
 - Variable-length argument can take from zero to n arguments
 - Ellipsis can be used only once in the parameter list
 - Ellipsis must be placed at the end of the parameter list



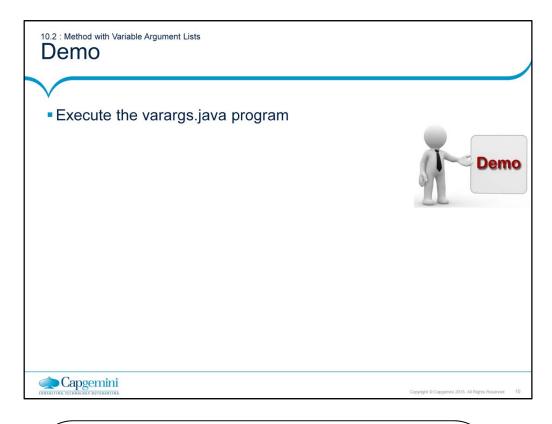
Copyright © Capgemini 2015. All Rights Reserved

J2SE5 has added a new feature that simplifies the creation of methods that need to take a variable number of arguments. This feature is called varargs and it is short for variable-length arguments.



Note that the ellipses (...) must come only after the last parameter. Putting it anywhere else is invalid.

The three periods indicate that the final argument may be passed as an array or as a sequence of arguments.



```
import static java.lang.System.*;
    public class varargs {
        static void print(int a,int y,String...s) {
            out.println(a+""+y);
            for(int i=0;i<s.length;i++) out.print(s[i]+"\t");
                out.println();
        }
        public static void main(String[] arg) {
            print(3,2,"java","java 5");
            out.println("Next invoke");
            print(1,2,"a","b","c","d","e");
        }
    }
}</pre>
```

Java Arrays expose a property called length that returns the length of the array.

```
O/P:
32
java java 5
Next invoke
12
a b c d e
```

Using java.util.Arrays Class

 This class contains lots of useful methods to manipulate contents of array

Method Name	Use
asList	Creates a new List from array
binarySearch	Use to search an element in an array
copyOf(array,n)	Creates new array of n size and copy all elements from array to new one
copyOfRange(array,n,from,to)	Creates new array of n size and copy specified elements from array to new one
sort	Sort elements of an array
equals	Compare two array elements
fill	Inserts specified value to each element of an array
stream(array)	Creates stream from an array



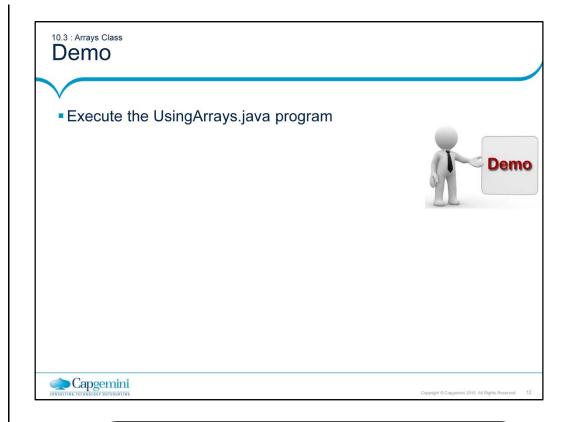
povright © Capgemini 2015. All Rights Reserved

Arrays Class:

Java provides utility Arrays class to manipulate arrays. This class is provided in the java.util package and includes lots of static methods to deal with array.

Please refer the java documentation of this class to know more about method signature and use.

Note: stream() method will be discussed in later chapter.



```
import java.util.Arrays;
    public class UsingArrays{
        static void sort(int...s) {
            Arrays.sort(s);
            for(int i=0;i<s.length;i++) out.print(s[i]+"");
                out.println();
        }
        public static void main(String[] arg) {
            sort(15,20,42,3,56,34);
        }
}</pre>
```

Summary In this lesson, you have learnt about: Creating and using array Manipulating array Iterating array Varargs Using java.util.Arrays class Capgemini

Review Question

- Question 1: If a display method accepts an integer array and returns nothing, is following call to display method is correct? State true or false.
 - display({10,20,30,40,50})
- Question 2: All methods in java.util.Arrays class are static (excluding Object class methods).
 - True/False





Copyright

Capgemini 2015, All Rights Reserved