JAVA BASICS

4KCJ001 – LECTURE 3

By
Sudha Agarwal
sudha.agarwal@4kitsolutions.com

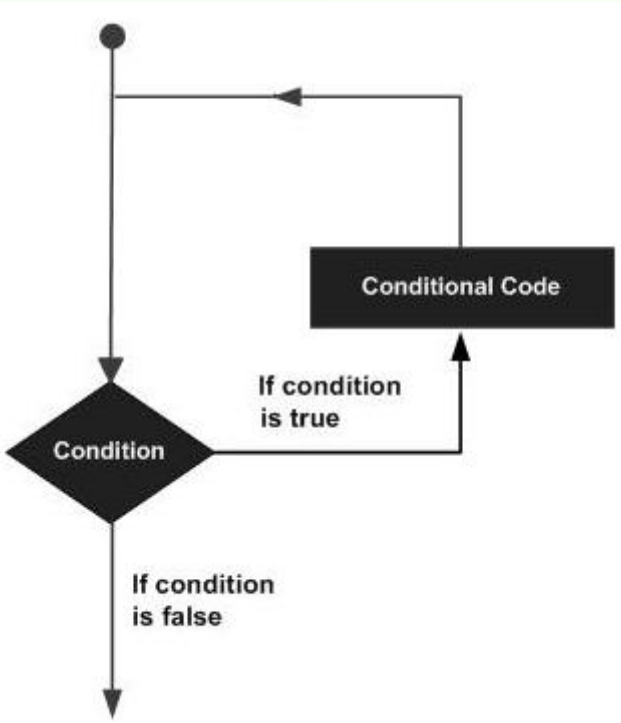
http://www.4kitsolutions.com/

INDEX

- Loops
- Decision Making
- Operators
- Arrays

LOOPS

❖There may be a situation when you need to execute a block of code several number of times. In general, statements are executed sequentially: The first statement in a function is executed first, followed by the second, and so on.



http://www.4kitsolutions.com/

TYPES OF LOOPS

❖**while loop**

❖Repeats a statement or group of statements while a given condition is true. It tests the condition before executing the loop body.

❖**for loop**

❖Execute a sequence of statements multiple times and abbreviates the code that manages the loop variable.

❖**do...while loop**

❖Like a while statement, except that it tests the condition at the end of the loop body.

LOOP CONTROL STATEMENTS

❖Loop control statements change execution from its normal sequence. When execution leaves a scope, all automatic objects that were created in that scope are destroyed.

❖**break statement**

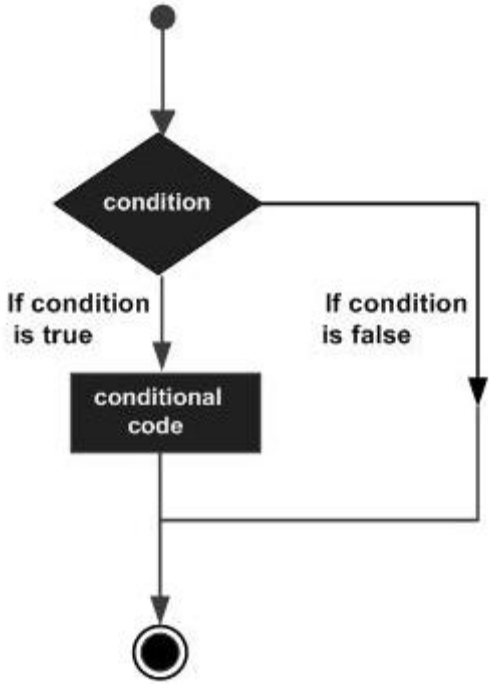
❖Terminates the loop or switch statement and transfers execution to the statement immediately following the loop or switch.

❖**continue statement**

❖Causes the loop to skip the remainder of its body and immediately retest its condition prior to reiterating.

DECISION MAKING

❖Decision making structures have one or more conditions to be evaluated or tested by the program, along with a statement or statements that are to be executed if the condition is determined to be true, and optionally, other statements to be executed if the condition is determined to be false.



http://www.4kitsolutions.com/

DECISION MAKING

❖Java programming language provides following types of decision making statements.

❖**if statement** : An if statement consists of a boolean expression followed by one or more statements.

❖**if...else statement** : An if statement can be followed by an optional else statement, which executes when the boolean expression is false.

❖**nested if statement** : You can use one if or else if statement inside another if or else if statement(s).

❖**switch statement** : A switch statement allows a variable to be tested for equality against a list of values.

OPERATORS

❖Java provides a rich set of operators to manipulate variables. We can divide all the Java operators into the following groups −

❖Arithmetic Operators
❖Relational Operators
❖Bitwise Operators
❖Logical Operators
❖Assignment Operators
❖Misc Operators

Operator Type	Category	Precedence
Unary	postfix	*expr++ expr--*
	prefix	*++expr --expr +expr -expr ~ !*
Arithmetic	multiplicative	* / %
	additive	+ -
Shift	shift	<< >> >>>
Relational	comparison	< > <= >= instanceof
	equality	== !=
Bitwise	bitwise AND	&
	bitwise exclusive OR	^
	bitwise inclusive OR	|
Logical	logical AND	&&
	logical OR	||
Ternary	ternary	? :
Assignment	assignment	= += -= *= /= %= &= ^= |= <<= >>= >>>=

ARRAYS

❖An array is a collection of similar data types.

❖Array is a container object that hold values of homogenous type.

❖It is also known as static data structure because size of an array must be specified at the time of its declaration.

❖An array can be either primitive or reference type. It gets memory in heap area. Index of array starts from zero to size-1.

❖It occupies a contiguous memory location.

ARRAY DECLARATION

```
datatype[ ] identifier;
or
datatype identifier[ ];
```

```
int[ ] arr;
char[ ] arr;
short[ ] arr;
long[ ] arr;
int[ ][ ] arr;   // two dimensional array.
```

Initialization of Array

❖**new** operator is used to initialize an array.

```
int[ ] arr = new int[10];    //this creates an empty
array named arr of integer type whose size is 10.
or
int[ ] arr = {10,20,30,40,50};  //this creates an
array named arr whose elements are given.
```

ACCESSING ARRAY ELEMENTS

❖Array index starts from 0. To access nth element of an array. Syntax

```
arrayname[n-1];
```

❖To find the length of an array, we can use the following syntax: array_name.length.

FOR EACH LOOP

- J2SE 5 introduces special type of for loop called **foreach** loop to access elements of array.
- Using **foreach** loop you can access complete array sequentially without using index of array.

```
int[] arr = {10, 20, 30, 40};
       for(int x : arr)
       {
```

COPYING ARRAY

❖Copying An Array Using for Loop.

```
int[] a = {12, 21, 0, 5, 7};   //Declaring and
initializing an array of ints

        int[] b = new int[a.length];
//Declaring and instantiating another array of ints
with same length

        for (int i = 0; i < a.length; i++)
        {
            b[i] = a[i];
        }
```

COPYING ARRAY

❖Copying An Array Using **copyOf()** Method of java.util.Array Class

```
int[] a = {12, 21, 0, 5, 7};   //Declaring and
initializing an array of ints

        //creating a copy of array 'a' using copyOf()
method of java.util.Arrays class

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

COPYING ARRAY

❖Copying An Array Using **clone()** Method.

❖All arrays will have clone() method inherited from **java.lang.Object** class. Using this method, you can copy an array.

```
int[] a = {12, 21, 0, 5, 7};   //Declaring and
initializing an array of ints

        //creating a copy of array 'a' using clone()
method

        int[] b = a.clone();
```

COPYING ARRAY

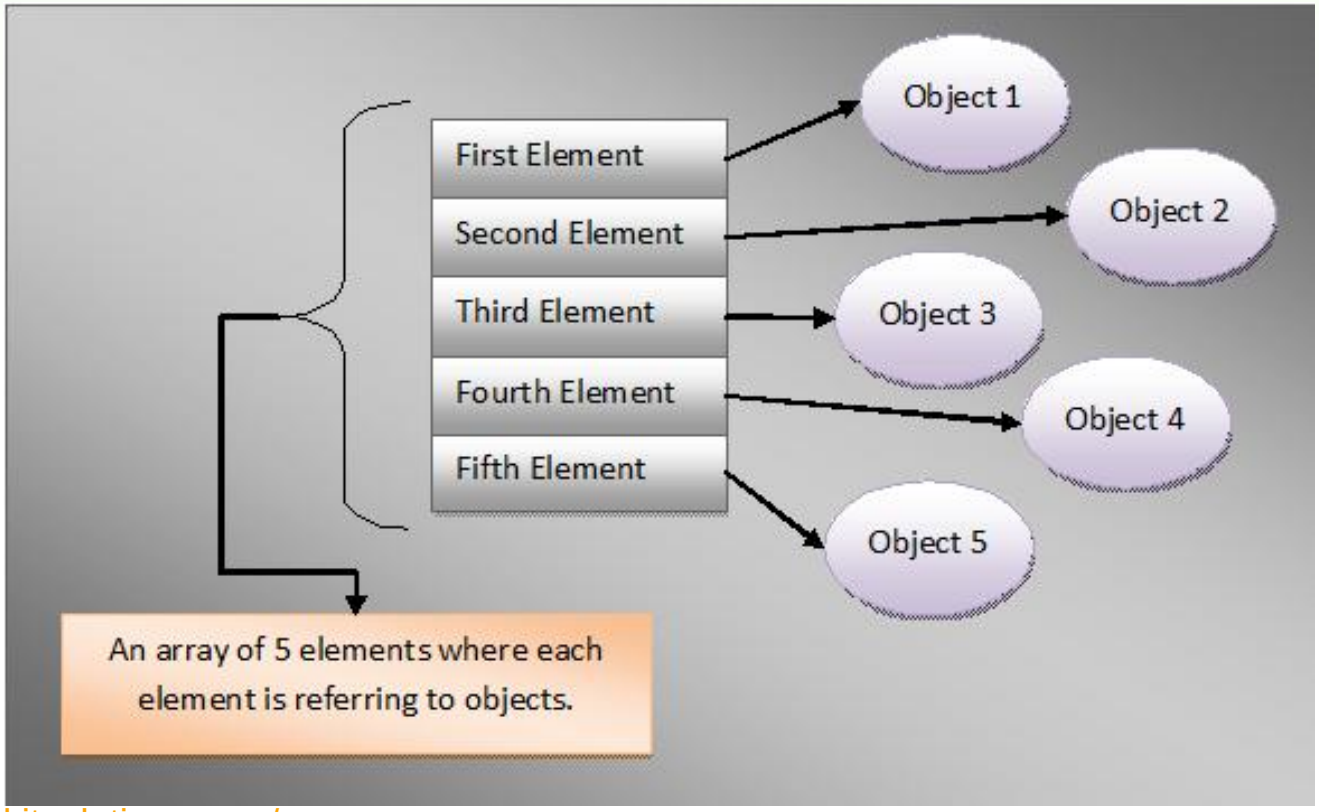
- Copying An Array Using **arraycopy()** Method Of **System Class**.
- Using **arraycopy()** method, you can copy a part of an array into another array.

```
int[] a = {12, 21, 0, 5, 7};   //Declaring and
initializing an array of ints

        //Creating an array object of same length as
array 'a'

        int[] b = new int[a.length];

        //creating a copy of array 'a' using
arraycopy() method of System class

        System.arraycopy(a, 0, b, 0, a.length);
```

ARRAYS OF OBJECTS

❖Array can hold the references to any type of objects.

❖ It is important to note that array can contain only references to the objects, not the objects itself.



Arrays as Parameters

❖Arrays can be passed to method as arguments and methods can return an array.

❖Arrays are **Passed-By-Reference**. That means, When an array is passed to a method, reference of an array object is passed not the copy of the object.

❖So, Any changes made to object in the method will be reflected in the actual object.

MULTI-DIMENSIONAL ARRAY

- A multi-dimensional array is very much similar to a single dimensional array.
- It can have multiple rows and multiple columns unlike single dimensional array

```
datatype[ ][ ] identifier;
int[ ][ ] arr = new int[10][10];
```

JAGGED ARRAY

❖**Jagged arrays** in java are arrays containing arrays of different length.

❖Jagged arrays are also multidimensional arrays. Jagged arrays in java sometimes are also called as **ragged arrays**.

```
int[ ][ ] arr = new int[3][ ];    //there will be 10
arrays whose size is variable
arr[0] = new int[3];
arr[1] = new int[4];
arr[2] = new int[5];
```

INHERITANCE (IS-A)

❖**Inheritance** is one of the key features of Object Oriented Programming.

❖Inheritance provided mechanism that allowed a class to inherit property of another class.

❖When a Class extends another class it inherits all non-private members including fields and methods.

❖Inheritance in Java can be best understood in terms of Parent and Child relationship, also known as **Super class(Parent)** and **Sub class(child)** in Java language.

LAB3

- Q1. Write a program to reverse an array.
- Q2. Write a program to traverse a two dimensional array.
- Q3. Write a program to print Fibonacci series upto given number.
- Q4. Write a program in Java to check if a number is even or odd in Java?
- Q5. Write Java program to check if a number is palindrome in Java? (121 is palindrome, 321 is not).
- Print following structure in Java?

```
*
***
*****
***
*
```

THANK YOU

http://www.4kitsolutions.com/

By
Sudha Agarwal
sudha.agarwal@4kitsolutions.com