



Web Technology (KCS-602) Unit 1

Prepared By

Abhishek Kesharwani

Assistant Professor, UCER Naini, Allahabad

Lecture 5

- Variable
- Arrays

Java Variable Types

Variable is name of reserved area allocated in memory.

There are three kinds of variables in Java:

- Local variables
- Instance variables
- Class/static variables

```
class A{  
  int data=50;//instance variable  
  static int m=100;//static variable  
  void method(){  
    int n=90;//local variable  
  }  
} //end of class
```

Local variables :

- Local variables are declared in methods, constructors, or blocks.
- Local variables are created when the method, constructor or block is entered and the variable will be destroyed once it exits the method, constructor or block.
- Access modifiers cannot be used for local variables.
- Local variables are visible only within the declared method, constructor or block.
- Local variables are implemented at stack level internally.
- There is no default value for local variables so local variables should be declared and an initial value should be assigned before the first use.

Instance variables :

- Instance variables are declared in a class, but outside a method, constructor or any block.
- When a space is allocated for an object in the **heap** a slot for each instance variable value is created.
- Instance variables are created when an object is created with the use of the key word '**new**' and destroyed when the object is destroyed.
- Instance variables hold values that must be referenced by more than one method, constructor or block, or essential parts of an object's state that must be present through out the class.

Class/static variables :

- Class variables also known as static variables are declared with the *static* keyword in a class, but outside a method, constructor or a block.
- There would only be one copy of each class variable per class, regardless of how many objects are created from it.
- Static variables are stored in static memory. It is rare to use static variables other than declared final and used as either public or private constants.

Java Modifier Types

- Modifiers are keywords that you add to those definitions to change their meanings. The Java language has a wide variety of modifiers, including the following:
 - Java Access Modifiers
 - Non Access Modifiers

Access Control Modifiers:

- Java provides a number of access modifiers to set access levels for classes, variables, methods and constructors. The four access levels are:
- Visible to the package. the **default**. No modifiers are needed.
- Visible to the class only (**private**).
- Visible to the world (**public**).
- Visible to the package and all subclasses (**protected**).

Private access modifier

- The private access modifier is accessible only within class.

```
class A{  
  private int data=40;  
  private void msg(){System.out.println("Hello java");}  
}
```

```
public class Simple{  
  public static void main(String args[]){  
    A obj=new A();  
    System.out.println(obj.data);//Compile Time Error  
    obj.msg();//Compile Time Error  
  }  
}
```

- If you make any class constructor private, you cannot create the instance of that class from outside the class.

Note: A class cannot be private or protected except **nested class.**

default access modifier

- If you don't use any modifier, it is treated as **default** by default. The default modifier is accessible only within package.

```
package pack;
```

```
class A{
```

```
    void msg(){System.out.println("Hello");}
```

```
}
```

```
//save by B.java
```

```
package mypack;
```

```
import pack.*;
```

```
class B{
```

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

```
        A obj = new A();//Compile Time Error
```

```
        obj.msg();//Compile Time Error
```

```
}
```

protected access modifier

- The **protected access modifier** is accessible within package and outside the package but through inheritance only.
- The protected access modifier can be applied on the data member, method and constructor. It can't be applied on the class.

```
//save by A.java  
package pack;  
public class A{  
protected void msg(){System.out.println("Hello");}  
}
```

```
//save by B.java  
package mypack;  
import pack.*;  
  
class B extends A{  
    public static void main(String args[]){  
        B obj = new B();  
        obj.msg();  
    }  
} Output:Hello
```

public access modifier

- The **public access modifier** is accessible everywhere. It has the widest scope among all other modifiers.

Access Modifier	within class	within package	outside package by subclass only
Private	Y	N	N
Default	Y	Y	N
Protected	Y	Y	Y
Public	Y	Y	Y

Non Access Modifiers:

Java provides a number of non-access modifiers to achieve many other functionality.

- The *static* modifier for creating class methods and variables
- The *final* modifier for finalizing the implementations of classes, methods, and variables.
- The *abstract* modifier for creating abstract classes and methods.

Interview Questions Asked in different Companies

1. Is Empty .java file name a valid source file name? [Infosys Interview]
2. What if I write static public void instead of public static void? [TCS Interview]
3. What are the various access specifiers in Java? (wipro)
4. What is the default value of the local variables? (wipro)

Java Array

- Array is a collection of similar type of elements that have contiguous memory location.
- **Java array** is an object that contains elements of similar data type.
- It is a data structure where we store similar elements.
- We can store only fixed set of elements in a java array.
- Array in java is index based, first element of the array is stored at 0 index.

Advantage of Java Array

- **Code Optimization:** It makes the code optimized, we can retrieve or sort the data easily.
- **Random access:** We can get any data located at any index position.

Disadvantage of Java Array

- **Size Limit:** We can store only fixed size of elements in the array. It doesn't grow its size at runtime. To solve this problem, collection framework is used in java.

Types of Array in java

There are two types of array.

- Single Dimensional Array
- Multidimensional Array

Single Dimensional Array in java

Syntax to Declare an Array in java

`dataType[] arr; (or)`

`dataType []arr; (or)`

`dataType arr[];`

Instantiation of an Array in java

`Array RefVar=new datatype[size];`

Example

`int a[]=new int[5];`

Example of single dimensional java array

```
class Testarray{  
public static void main(String args[]){  
    int a[]=new int[5];  
    a[0]=10;  
    a[1]=20;  
    a[2]=70;  
    a[3]=40;  
    a[4]=50;  
    for(int i=0;i<a.length;i++)  
        System.out.println(a[i]);  
    }}
```

Declaration, Instantiation and Initialization of Java Array

We can declare, instantiate and initialize the java array together by:

```
int a[]={33,3,4,5};//declaration, instantiation and initialization
class Testarray1{
public static void main(String args[]){
    int a[]={33,3,4,5};//declaration, instantiation and initialization
    //printing array
    for(int i=0;i<a.length;i++)//length is the property of array
    System.out.println(a[i]);
}}
```


Passing Array to method in java

```
class Testarray2{
static void min(int arr[])
{
int min=arr[0];
for(int i=1;i<arr.length;i++)
if(min>arr[i])
min=arr[i];
System.out.println(min);
}

public static void main(String args[]){
int a[]={33,3,4,5};
min(a);//passing array to method
}}
```

Multidimensional array in java

In such case, data is stored in row and column based index (also known as matrix form).

Syntax to Declare Multidimensional Array in java.

- `dataType[][] arrayRefVar; (or)`
- `dataType [][]arrayRefVar; (or)`
- `dataType arrayRefVar[][]; (or)`
- `dataType []arrayRefVar[];`

Example to instantiate Multidimensional Array in java

```
int[][] arr=new int[3][3];//3 row and 3 column
```

Example to initialize Multidimensional Array in java

```
arr[0][0]=1;
```

```
arr[0][1]=2;
```

```
arr[0][2]=3;
```

```
arr[1][0]=4;
```

```
arr[1][1]=5;
```

```
arr[1][2]=6;
```

```
arr[2][0]=7;
```

```
arr[2][1]=8;
```

```
arr[2][2]=9;
```

Example of Multidimensional java array

```
class Testarray3{
public static void main(String args[]){
    //declaring and initializing 2D array
    int arr[][]={{1,2,3},{2,4,5},{4,4,5}};
    //printing 2D array
    for(int i=0;i<3;i++){
        for(int j=0;j<3;j++){
            System.out.print(arr[i][j]+" ");
        }
        System.out.println();
    }
}
```

```
class Testarray5{  
public static void main(String args[]){  
int a[][]={{1,3,4},{3,4,5}};  
int b[][]={{1,3,4},{3,4,5}};  
int c[][]=new int[2][3];  
for(int i=0;i<2;i++){  
for(int j=0;j<3;j++){  
c[i][j]=a[i][j]+b[i][j];  
System.out.print(c[i][j]+" ");  
}  
System.out.println();//new line  
}  
}}
```