**Java – static variable with example**

[**OOPS CONCEPT**](http://beginnersbook.com/category/oops-concept/)

Earlier we discussed [**static class**](http://beginnersbook.com/2013/04/java-static-class-block-methods-variables/), [**static methods**](http://beginnersbook.com/2013/05/static-vs-non-static-methods/) and [**static import**](http://beginnersbook.com/2013/05/java-static-import/) in java. In this tutorial we will see what are **static variables** and how they can be used in java.

**Static variable**

syntax:

static int age;

**Note**: Static variable’s value is same for all the object(or instances) of the class or in other words you can say that all instances(objects) of the same class share a single copy of static variables.

lets understand this with an example:

class VariableDemo

{

static int count=0;

public void increment()

{

count++;

}

public static void main(String args[])

{

VariableDemo obj1=new VariableDemo();

VariableDemo obj2=new VariableDemo();

obj1.increment();

obj2.increment();

System.out.println("Obj1: count is="+obj1.count);

System.out.println("Obj2: count is="+obj2.count);

}

}

**Output:**

Obj1: count is=2

Obj2: count is=2

As you can see in the above example that both the objects of class, are sharing a same copy of static variable that’s why they displayed the same value of count.

**Static variable initialization**

1. Static variables are initialized when class is loaded.
2. Static variables in a class are initialized before any object of that class can be created.
3. Static variables in a class are initialized before any static method of the class runs.

Default values for declared and uninitialized static and non-static variables are same

primitive integers(long, short etc): 0  
primitive floating points(float, double): 0.0  
boolean: false  
object references: null

**Static final variables**

static final variables are constants. consider below code:

public class MyClass{

public static final int MY\_VAR=27;

}

**Note: Constant variable name should be in Caps! you can use underscore(\_) between.**  
1) The above code will execute as soon as the class MyClass is loaded, before static method is called and even before any static variable can be used.  
2) The above variable MY\_VAR is **public** which means any class can use it. It is a **static** variable so you won’t need any object of class in order to access it. It’s **final** so this variable can never be changed in this or in any class.

**Key points:**  
final variable always needs initialization, if you don’t initialize it would throw a compilation error. have a look at below example-

public class MyClass{

public static final int MY\_VAR;

}

Error: variable MY\_VAR might not have been initialized