



STATIC KEYWORD IN JAVA

Velocity



JULY 28, 2022

VELOCITY
Pune

Introduction to Static Keyword in Java

Static keyword in java:

- In Java, static keyword is mainly used for memory management.
- It can be used with variables, methods, blocks, and nested classes.
- It is a keyword which is used to share the same variable or method of a given class. Basically, static is used for a constant variable or a method that is same for every instance of a class.
- It means single copy storage.

Q> How static keyword helps in memory management?

A> The main purpose of using the static keyword in Java is to save memory. When we create a variable in a class that will be accessed by other classes, we must first create an instance of the class and then assign a new value to each variable instance – even if the value of the new variables is supposed to be the same across all new classes/objects.

But when we create a static variable, its value remains constant across all other classes, and we do not have to create an instance to use the variable. This way, we are creating the variable once, so memory is only allocated once.

Static Variable in Java:

- A variable which is defined with static keyword called as “static variable.”
- It is used to refer the common property of all the objects.
- Static variables get loaded into memory at the time of class loading.

How to declare static variable.

Example-

```
static int a; //Declaration
```

```
static int b =10; //Initialization
```

Note:- Local variables cannot be static.

How to access static variable.

There are two ways to access the static variables.

1. By using class name
2. By using object name.
3. Direct calling

```

public class StaticContext {

    static int x = 10;

    public static void main(String[] args) {

        StaticContext object = new StaticContext();

        System.out.println(StaticContext.x); // by using class name

        System.out.println(object.x); // by using object name

        System.out.println(x); // by direct calling

    }

}

```

Q> Why it is called as single copy storage?

A>Memory allocation for a static variable happens only once in the class area when the class is loaded in the memory. It is also known as a class variable. It is common to all the objects of the class. In this, a single copy of a static variable is created and shared among all the objects of the class.

```
package com.velocity;
```

```

public class Test10 {

    int a = 5;
    static int b = 5;

    public static void main(String args[]) {

        Test10 sd2 = new Test10();
        System.out.println("non static>>" + sd2.a++);
        System.out.println("static>>" + sd2.b++);
        Test10 sd3 = new Test10();
        System.out.println("non static>>" + sd3.a++);
        System.out.println("static>>" + sd3.b++);
        Test10 sd4 = new Test10();
        System.out.println("non static>>" + sd4.a++);
        System.out.println("static>>" + sd4.b++);
        Test10 sd5 = new Test10();
        System.out.println("non static>>" + sd5.a++);
        System.out.println("static>>" + sd5.b++);

    } }

```

Output-

```
non static>>5
static>>5
non static>>5
static>>6
non static>>5
static>>7
non static>>5
static>>8
```

Note- We cannot call non-static member from static member because static variables stored into memory before object creation and non-static variables stored into memory after object creation.

How to access static members from non-static members. Following program shows you.

```
package Mypackage2;

public class Demo {

    void test() {
        System.out.println("This is Non-static method");
        Demo.x1(); //Calling static method from non-static method
    }

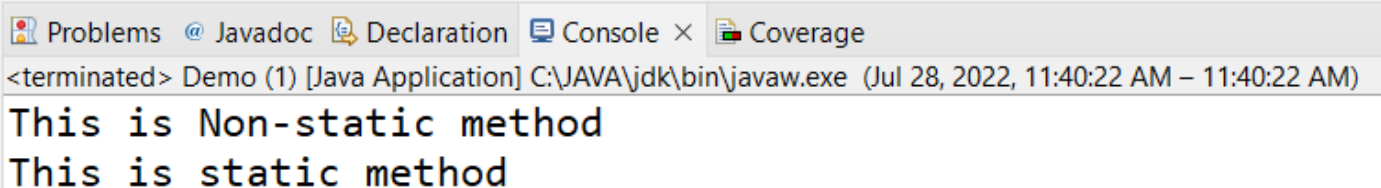
    static void x1() {
        System.out.println("This is static method");
    }

    public static void main(String[] args) {

        Demo demo = new Demo();
        demo.test();

    }

}
```



The screenshot shows an IDE interface with tabs for Problems, Javadoc, Declaration, Console, and Coverage. The Console tab is active, displaying the output of the Java application. The output consists of two lines: "This is Non-static method" and "This is static method". The console title bar indicates the application is "Demo (1) [Java Application]" running at "C:\JAVA\jdk\bin\javaw.exe" on "Jul 28, 2022, 11:40:22 AM".

```
<terminated> Demo (1) [Java Application] C:\JAVA\jdk\bin\javaw.exe (Jul 28, 2022, 11:40:22 AM – 11:40:22 AM)
This is Non-static method
This is static method
```

Static Method in java

If you define any method with static keyword, then it is called as static method.

It belongs to class rather than object of class.

It loads into memory before object creation.

It can access only static data member only.

Note: - Main method is static method.

```
public class Demo {  
    static void x1() {  
        System.out.println("This is static method");  
    }  
    public static void main(String[] args) {  
        Demo.x1();  
    }  
}
```

Static block in java

It is group of statements that are executed when class is loading into memory by Class loader.

It is widely used to create the static resource.

We cannot access non-static variable into static block.

It is always executed first.

```
1 package Mypackage2;  
2  
3 public class Demo {  
4  
5     static {  
6         System.out.println("This is static block executing 1st ...");  
7     }  
8  
9     public static void main(String[] args) {  
10  
11         System.out.println("This is main method");  
12  
13     }  
14  
15 }  
16
```

Problems Javadoc Declaration Console × Coverage
<terminated> Demo (1) [Java Application] C:\JAVA\jdk\bin\javaw.exe (Jul 28, 2022, 11:48:48 AM – 11:48:48 AM)
This is static block executing 1st ...
This is main method

Here, we will get first output is “This is static block executing 1st” because it is executed first than main method.

Note- Outer class cannot be static but inner class can be static.

Note- Constructor cannot be static.

When to use which type :

Static Variable:

You can use Static Variables to save memory or in an operation where you want all threads to maintain one variable instead of having a different variable for every thread.

Static Method:

Used when methods are more relevant to class than an instance of a class. Or common behavior needs to be shared by all objects.

Static Block:

Used when you want to initialize any data before the execution control goes to other methods like the main method.

Static Control flow mechanism in java

The Static Control Flow mechanism performs the following 3 steps in the exact chronological order:

1. Identification of static members from top to bottom. All the static variables, methods, and blocks are identified during this step.
2. Execution of static variable assignments and static blocks from top to bottom.
3. Finally, the static main method is executed in the last step of the static control flow.

class StaticDemo

```
{  
  (1) static int i = 0; (7)  
  (2) static  
  {  
    m1(); (8)  
    System.out.println("First Static Block"); (10)  
  }  
  (3) public static void main(String[] args)  
  {  
    m1(); (13)  
    System.out.println("Main Method"); (15)  
  }  
  (4) public static void m1()  
  {  
    System.out.println(i); (9) (14)  
  }  
  (5) static  
  {  
    System.out.println("Second Static Block"); (11)  
  }  
  (6) static int j = 20; (12)  
}
```