The static can be:

1. [Static Variables](https://www.guru99.com/java-static-variable-methods.html#1)
2. [Static Methods](https://www.guru99.com/java-static-variable-methods.html#2)
3. [Static Blocks Of Code.](https://www.guru99.com/java-static-variable-methods.html#3)

Let’s look at static variables and static methods first.

**What is Static Variable in Java?**

**Static variable in Java** is variable which belongs to the class and initialized only once at the start of the execution. It is a variable which belongs to the class and not to object(instance ). Static variables are initialized only once, at the start of the execution. These variables will be initialized first, before the initialization of any instance variables.

* A single copy to be shared by all instances of the class
* A static variable can be accessed directly by the class name and doesn’t need any object

**Syntax:**

<***class-name>.<variable-name>***

**What is Static Method in Java?**

**Static method in Java** is a method which belongs to the class and not to the object. A static method can access only static data. It is a method which belongs to the class and not to the object(instance). A static method can access only static data. It cannot access non-static data (instance variables).

* A static method can call only other static methods and can not call a non-static method from it.
* A static method can be accessed directly by the class name and doesn’t need any object
* A static method cannot refer to “this” or “super” keywords in anyway

**Syntax:**

<***class-name>.<method-name>***

Text

Description automatically generated

### Example: How to call static variables & methods

**Step 1)** Copy the following code into a editor

public class Demo{

public static void main(String args[]){

Student s1 = new Student();

s1.showData();

Student s2 = new Student();

s2.showData();

//Student.b++;

//s1.showData();

}

}

class Student {

int a; //initialized to zero

static int b; //initialized to zero only when class is loaded not for each object created.

Student(){

//Constructor incrementing static variable b

b++;

}

public void showData(){

System.out.println("Value of a = "+a);

System.out.println("Value of b = "+b);

}

//public static void increment(){

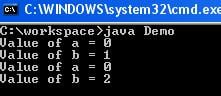
//a++;

//}

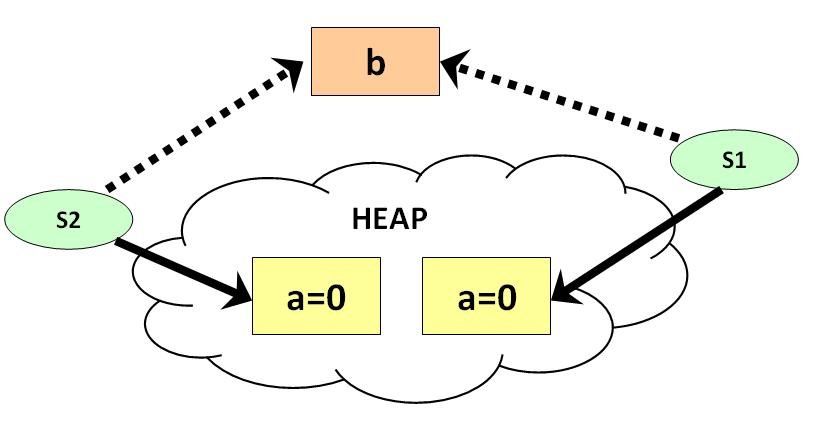
}

**Step 2)** Save & Compile the code. Run the code as, **java Demo**.

**Step 3)** Expected output show below



Following diagram shows, how reference variables & objects are created and static variables are accessed by the different instances.



**Step 4)** It is possible to access a static variable from outside the class using the syntax **ClassName.Variable\_Name**. Uncomment line # 7 & 8 . Save , Compile & Run . Observe the output.

Value of a = 0

Value of b = 1

Value of a = 0

Value of b = 2

Value of a = 0

Value of b = 3

**Step 5)** Uncomment line 25,26 & 27 . Save , Compile & Run.

error: non-static variable a cannot be referenced from a static context a++;

**Step 6)** Error = ? This is because it is not possible to access instance variable “**a**” from java static class method “**increment**“.

## ****What is Static Block in Java?****

The **static block** is a block of statement inside a Java class that will be executed when a class is first loaded into the JVM. A **static block helps to initialize the static data members**, just like constructors help to initialize instance members.

class Test{

static {

//Code goes here

}

}

Following program is the example of java static block.

### Example: How to access static block

public class Demo {

static int a;

static int b;

static {

a = 10;

b = 20;

}

public static void main(String args[]) {

System.out.println("Value of a = " + a);

System.out.println("Value of b = " + b);

}

}

**you will get following output of the program.**

Value of a = 10

Value of b = 20