Java Keyword - static

By Umesh Sir

Contact us 7758094241

static

- What is static?
- static variable?
- Problem without static variable.
- Explain System.out.println().
- static method?
- Why main() method is always static in Java?
- Static Block
- Non-Static Block.

★ What is static?

The static keyword in Java is used for memory management mainly. We can apply static keyword with variables, methods, blocks and nested classes. The static keyword belongs to the class than an instance of the class.

★ The Static Can Be -

1.1] Variable (Know as Class Variable)

1.2] Method (Know as Class Method)

1.3] Block

★ Static variable?

The static variable can be used to refer common property of all objects.

★ ex : Company name of Employee, College name for Student, bank Interest rate.

The static variable gets memory only once in class area at the time of class loading.

Advantages of Static Variable :

1] It makes program memory efficient (save memory).

★ Problem without static variable.

```
class Student{

 int rollno;
 String name;
 String college="COEP";

}
```

Suppose there are 500 students in COEP college, now all instance data members will get memory each time when the object is created. All students have its unique rollno and name, so instance data member is good in such case. Here, "college" refers to the common property of all objects. But it's instance level variable, So it will get memory each time. If we make it static as given below, this field will get the memory only once.

```
class Student{

 int rollno;
 String name;
 static String college="COEP";

}
```

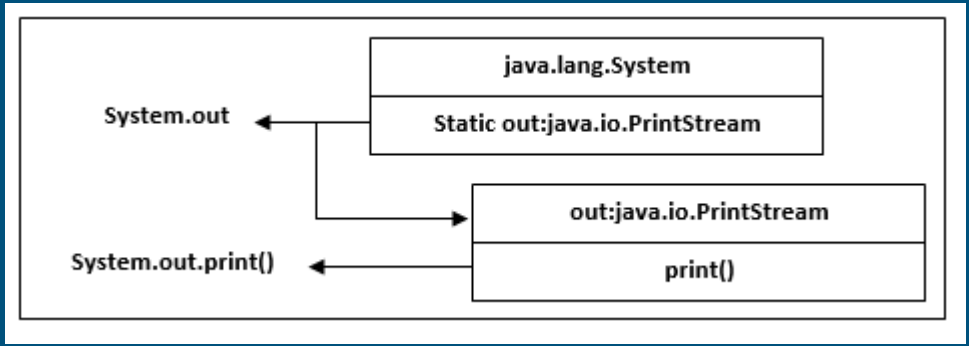
★ Explain System.out.println().

Java System.out.println() is used to print an argument that is passed to it. The statement can be broken into 3 parts which can be understood separately as:

System: It is a final class defined in the java.lang package.

out: This is an instance of PrintStream type, which is a public and static member field of the System class.

println(): As all instances of PrintStream class have a public method println(), hence we can invoke the same on out as well. This is an upgraded version of print(). It prints any argument passed to it and adds a new line to the output. We can assume that System.out represents the Standard Output Stream.



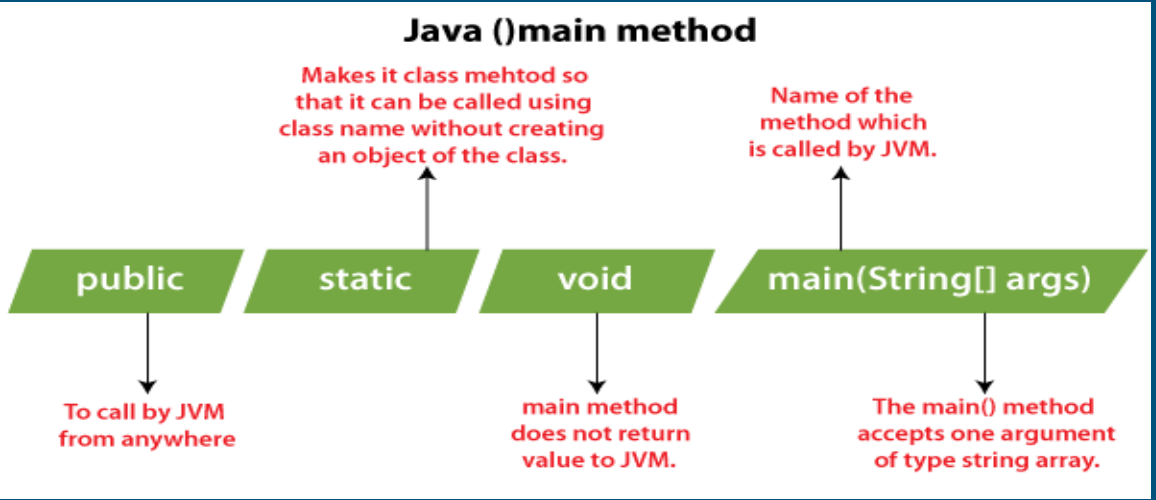
★ static method?

- ★ If we apply static keyword with any method then it is known as Static Method.
- ★ If method is Static we can call method with ought creating object directly by using class name.

★ Why main() method is always static in Java?

In Java, the main() method plays a vital role in program execution. The main() method is the first method that encounters first during execution. So, it is an entry point of a program. We cannot modify the syntax of the main() method. The only thing which we can change is the name of the String array argument. The syntax of the main() method is as follows:

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



★ Static Block

Is used to initialize the static data member.

It is executed before the main method at the time of classloading.

A class can have multiple Static blocks, which will execute in the same sequence in which they have been written into the program.

Output:

```
Static Block 1
Static Block 2
Value of num: 98
Value of mystr: Block2
```

```
class JavaExample2{
   static int num;
   static String mystr;
   //First Static block
   static{
       System.out.println("Static Block 1");
       num = 68;
       mystr = "Block1";
   }
   //Second static block
   static{
       System.out.println("Static Block 2");
       num = 98;
       mystr = "Block2";
   }
   public static void main(String args[])
   {
       System.out.println("Value of num: "+num);
       System.out.println("Value of mystr: "+mystr);
    }
}
```

★ Non-Static Block.

It is called Instance initialization blocks.

★ Similar to static blocks, Java also provides instance initialization blocks which are used to initialize instance variables, as an alternative to constructors.

★ Whenever you define an initialization block Java copies its code to the constructors. Therefore you can also use these to share code between the constructors of a class.

```
public class Student {
   String name;
   int age;
   {
      name = "Krishna";
      age = 25;
   }
   {
      System.out.println("Initialization block");
   }
   public static void main(String args[]){
      Student std = new Student();
      System.out.println(std.age);
      System.out.println(std.name);
   }
}
```

Output

```
Initialization block
25
Krishna
```

Thank you