

When to use static variables and methods?

Use the static variable for the property that is common to all objects. For example, in class Student, all students share the same college name. Use static methods for changing static variables.

Consider the following java program that illustrates the use of *static* keyword with variables and methods.

```
// A java program to demonstrate use of
// static keyword with methods and variables

// Student class
class Student
{
    String name;
    int rollNo;

    // static variable
    static String cllgName;

    // static counter to set unique roll no
    static int counter = 0;

    public Student(String name)
    {
        this.name = name;

        this.rollNo = setRollNo();
    }

    // getting unique rollNo
    // through static variable(counter)
    static int setRollNo()
    {
        counter++;
        return counter;
    }

    // static method
    static void setCllg(String name){
        cllgName = name ;
    }

    // instance method
    void getStudentInfo(){
        System.out.println("name : " + this.name);
        System.out.println("rollNo : " + this.rollNo);
    }
}
```

```

        // accessing static variable
        System.out.println("cllgName : " + cllgName);
    }
}

//Driver class
public class StaticDemo
{
    public static void main(String[] args)
    {
        // calling static method
        // without instantiating Student class
        Student.setCllg("XYZ");

        Student s1 = new Student("Alice");
        Student s2 = new Student("Bob");

        s1.getStudentInfo();
        s2.getStudentInfo();

    }
}

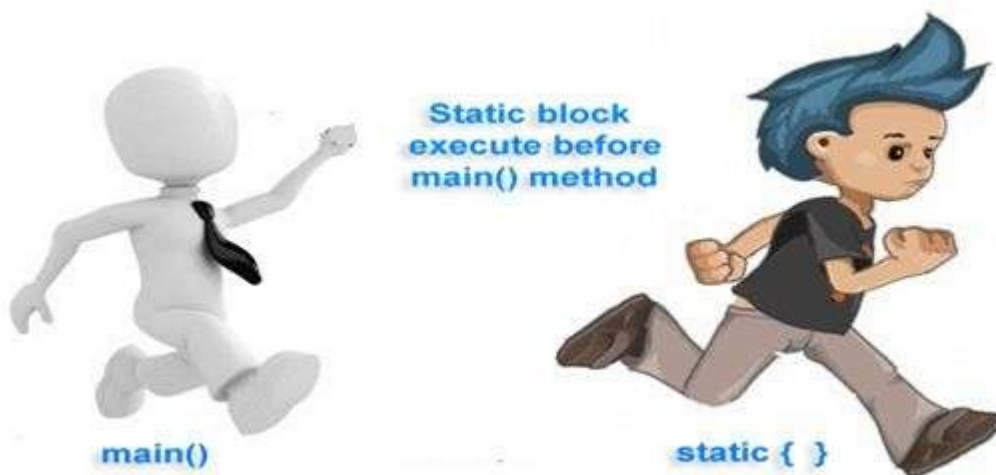
```

Static Block in Java

Static block is a set of statements, which will be executed by the JVM before execution of main method.

At the time of class loading if we want to perform any activity we have to define that activity inside static block because this block execute at the time of class loading.

In a class we can take any number of static block but all these blocks will be execute from top to bottom.



Syntax

static

{

.....

//Set of Statements

.....

}

Note: In real time applications, static block can be used whenever we want to execute any instructions or statements before execution of main method.

Example of Static Block

```
class StaticDemo{
static{
System.out.println("Hello how are u ?");
}
public static void main(String args[]){
System.out.println("This is main()");
}
}
```

Output

Hello how are u ?

This is main()

Run java program without main method

```
class StaticDemo
{
static
{
System.out.println("Hello how are u ?");
}
}
```

Output

Output

:

Hello how are u ?

Exception is thread "main" java.lang.no-suchmethodError:Main

Note: "Exception is thread "main" java.lang.no-suchmethodError:Main" warning is given in java 1.7 and its above versions

More than one static block in a program

```
class StaticDemo{
static{
System.out.println("First static block");
}
static{
System.out.println("Second Static block");
}
public static void main(String args[]){
System.out.println("This is main()");
}
}
```

Output

Output:

First static block

Second static block

This is main()

Note: "Here static block run according to their order (sequence by) from top to bottom.

Why a static block executes before the main method?

A class has to be loaded in main memory before we start using it. Static block is executed during class loading. This is the reason why a static block executes before the main method.

