


```
class Name
{
    // class area // global area --->
}
```



Note :

1. if we create any members (variable or a method) in class area(Global area) they are known as global members.
2. A global member can be used anywhere inside any member present within the class, or it can also be used inside any member present in a different class.
3. Global members are classified into two types :
 - a. **static members**
 - b. **non-static members**

Static Members :

Any member declared in the class prefixed with **static** keyword is known as static member.

example :

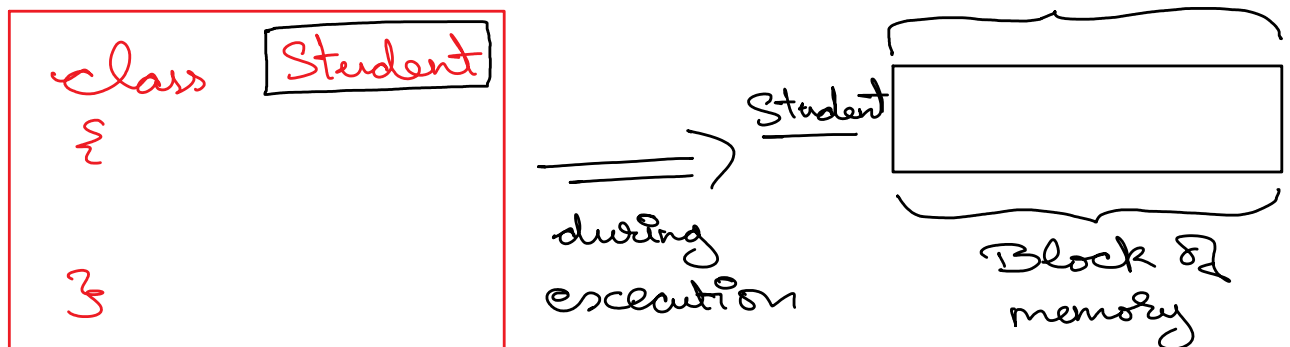
```
class Demo1
{
    static int a ; // static global primitive int variable of class Demo1
    static String name ; // static global non-primitive String variable

    static void print() // it is a static method
    {

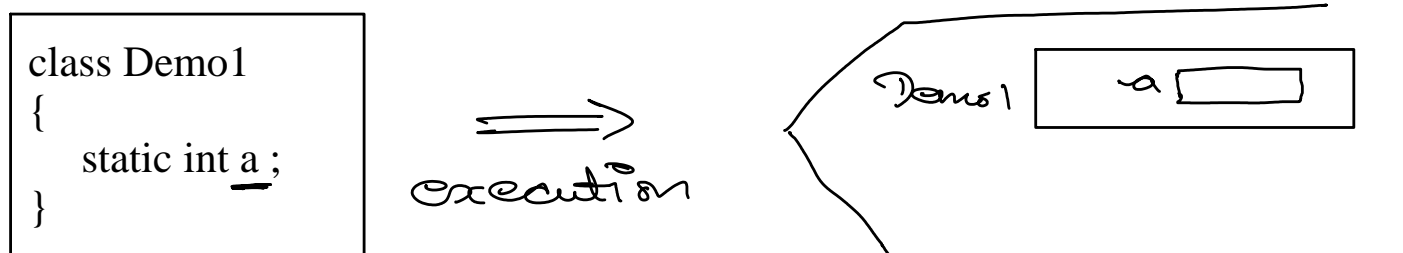
    }
}
```

Note :

1. During execution every class will have a dedicated memory space for itself, it is known as class static area



2. static members will be stored in the class static area.



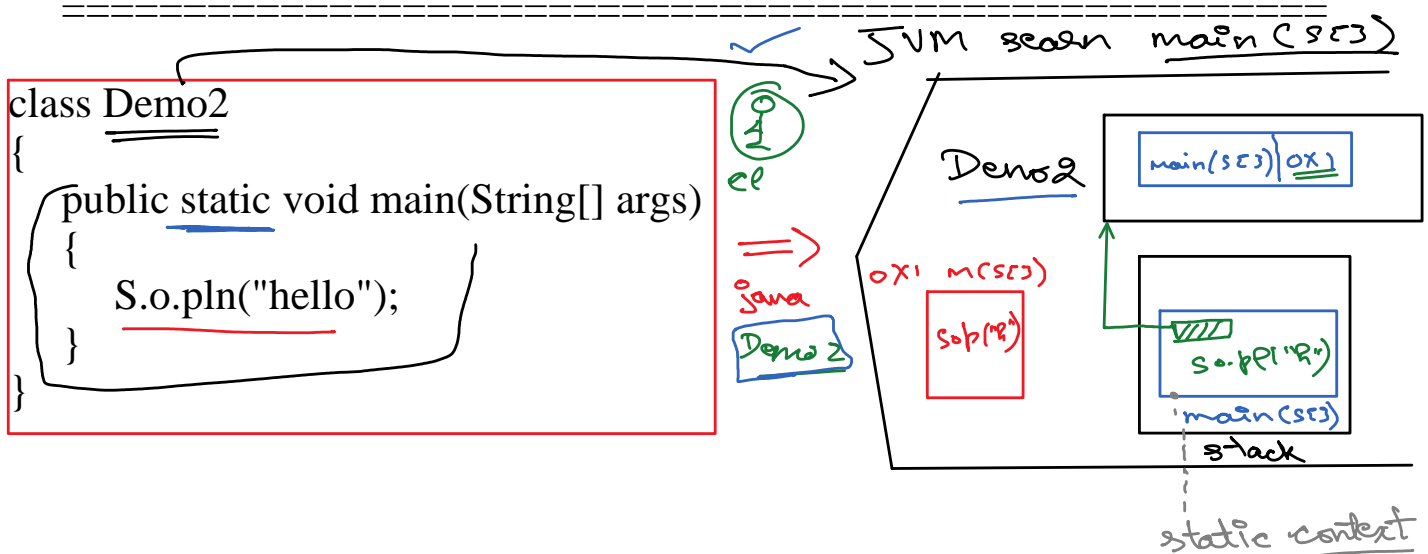
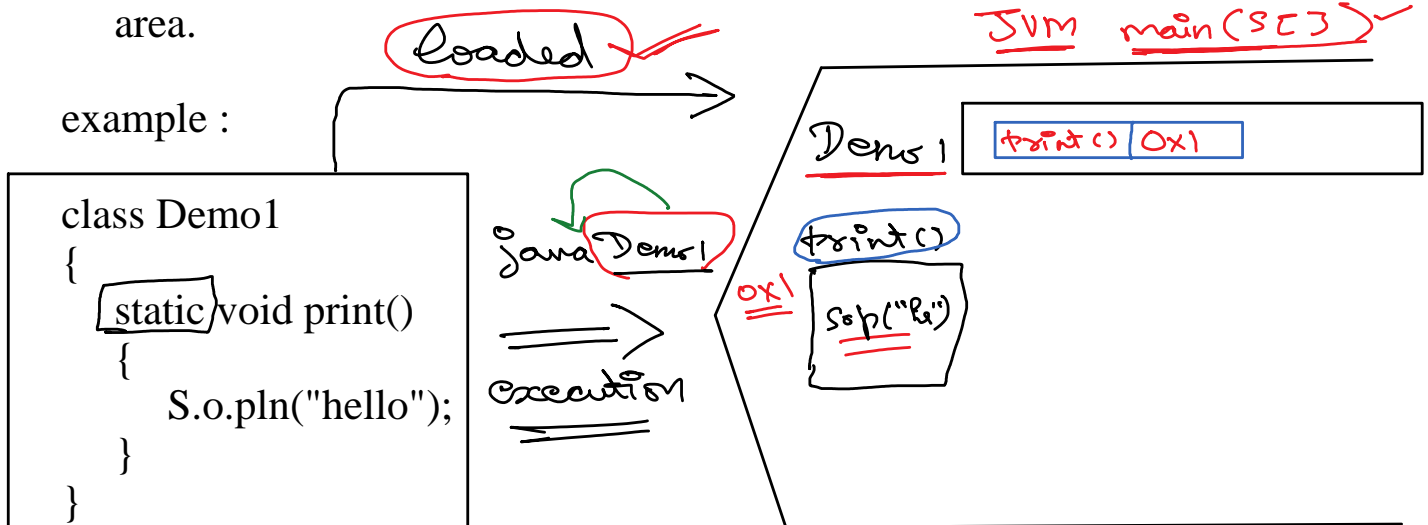
Static Members :

1. static methods
2. static variables
3. static initializers / static blocks

1. static method :

1. a method declared in the class prefixed with static keyword is known as static method.
2. the static method block will be stored in the method area.
3. the static method address (reference) will be stored inside class static area.

example :



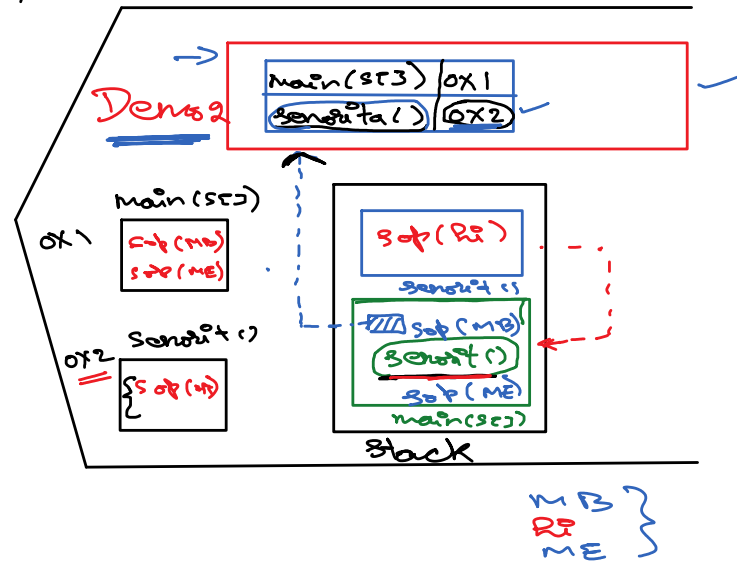
```

class Demo2
{
    public static void main(String[] args)
    {
        System.out.println("Main Begin ");
        Demo2 . senorita() ;
        System.out.println("Main End " );
    }

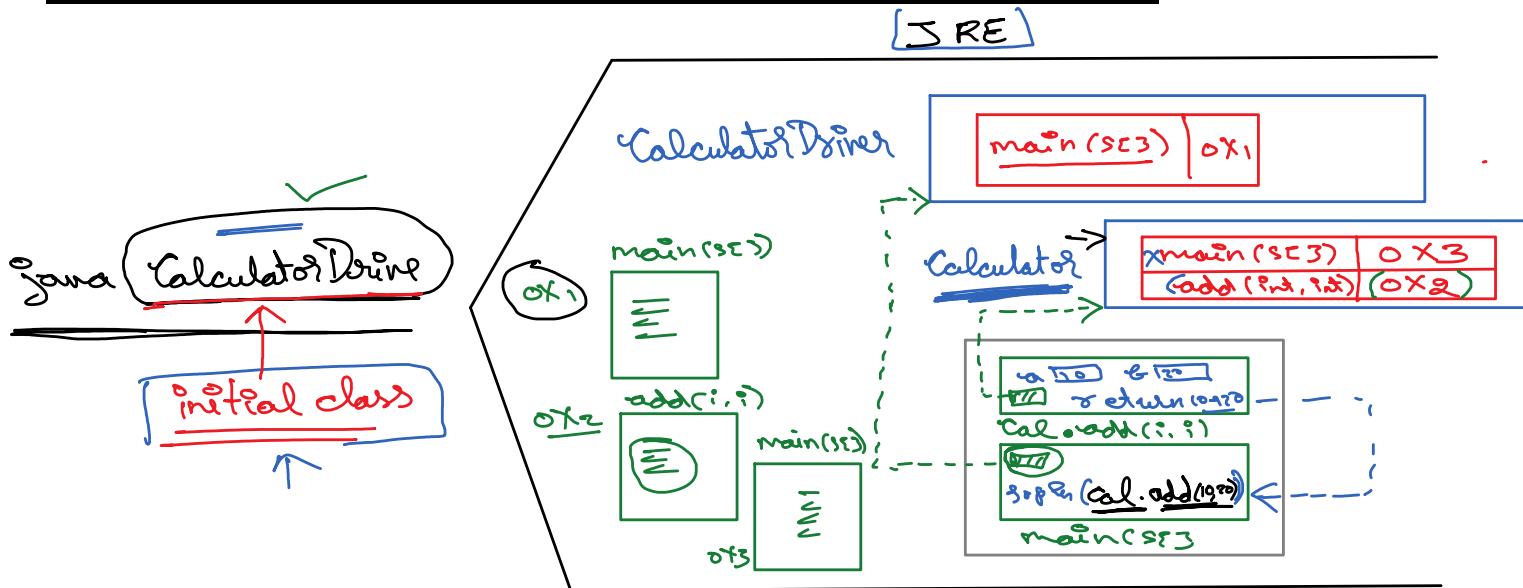
    static void senorita()
    {
        System.out.println("Hi...!!! ");
    }
}

```

Java
Demo2



Refer app8/Demo/ Calculator.java & CalculatorDriver.java



Summary :

1. static method reference is stored in class static area.
2. static method can be used either directly or with the help of class name inside the same class.
3. static method can be used with the help of class name in a different class.
4. the block which belongs static is known as **static context**.

5. static context will have reference to it's class static area.

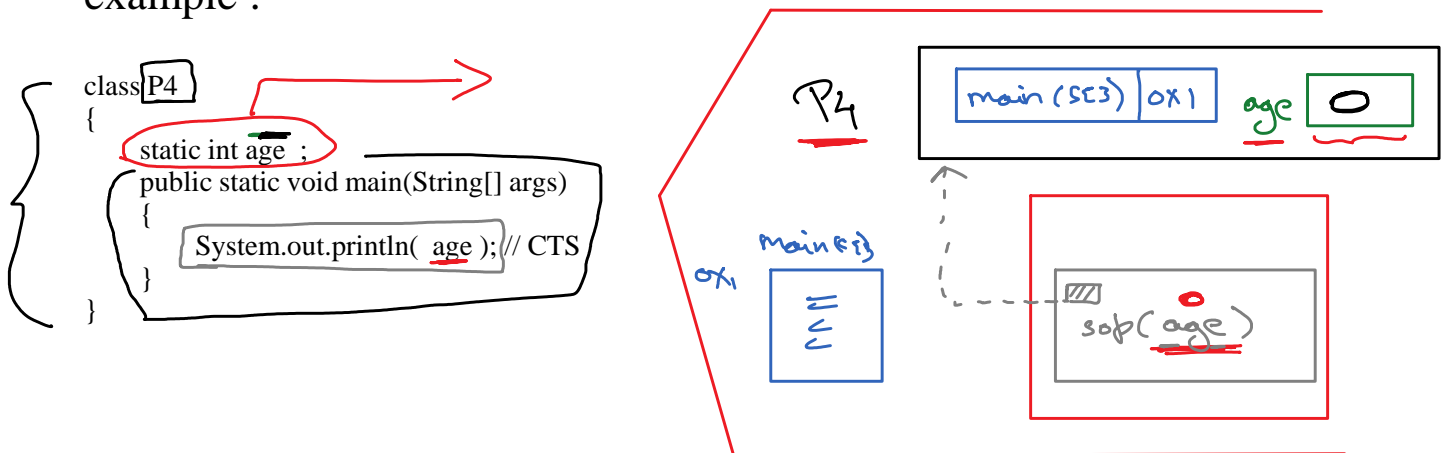
2. static variables :

A global variable prefixed with static keyword is known as static variable.

Note :

1. static variable will be assigned with default value. Therefore, we can use them without initialization.
2. static variable get its memory inside the class static area.

example :



3. We can use static variable either directly or with the help of class name within the members of the same class.

Task1 : Write a java program to create 2 static variables, and assign your name and age in the main method and display them

4. static variable is having global scope can be used :

a) inside any member of the same class.

example refer, **app8/static_variable/P7.java**

b) in a different class member with the help of class_name as reference .

Assignment 1 :

1. create a class for Employee, who has eid, name and salary as properties.
2. In Employee class create a method to display details of an employee
3. Create a Driver class to assign the values for an employee and print the data.

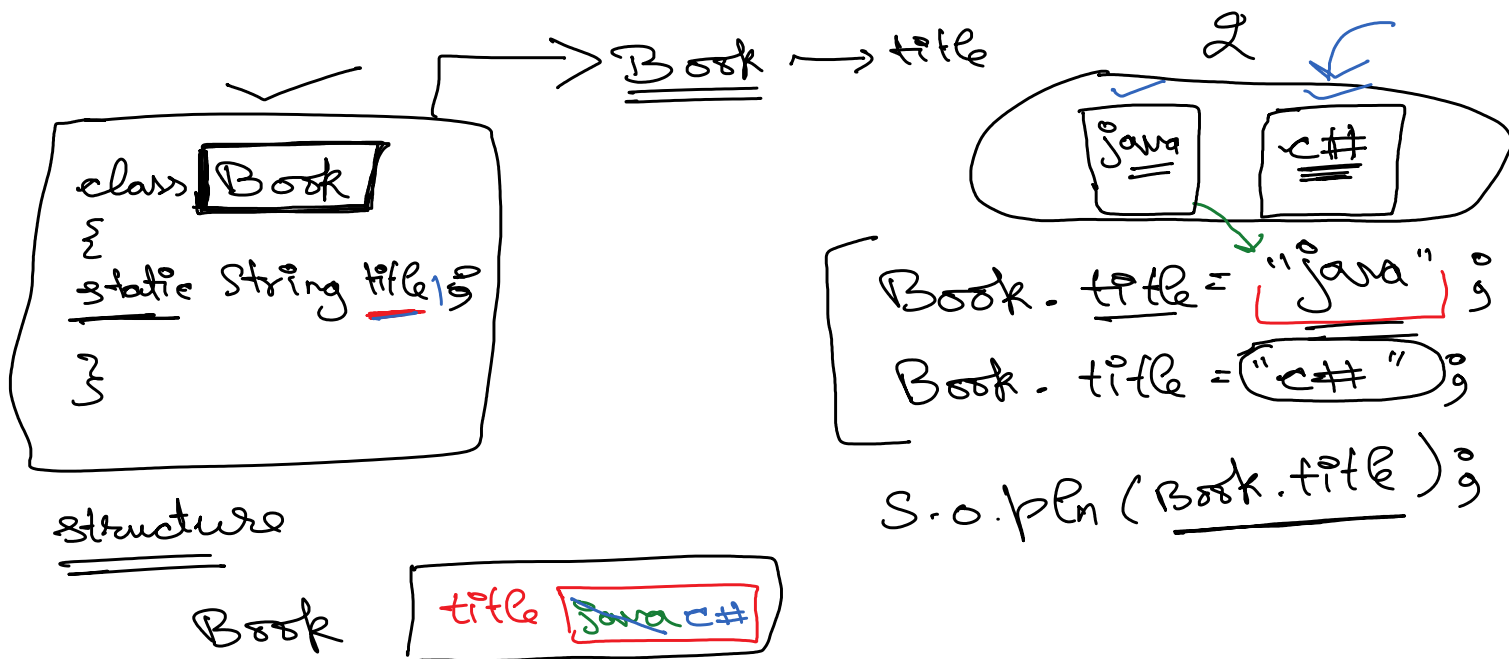
Note :

1. if the local variable and the static variable is having same name, the priority is given to local variable. We can still use the static variable with the help of class_name as the reference.

example : refer **app8/static_variable/P10.java**

Key Points :

1. static variables is also known as class variables.
2. static variable will be loaded during the loading process of a class.
3. static variable will have only one copy. (static variables will be allocated only once per class).



— Book | title java c#

static Initializer :

1. static declare and initialization statement :

```
static int a = 10 ;
```

↳ static initializer

2. static block / static Initializer bloc (SIB) :



```
static
{
    // statements
}
```

→ static initializer/
static block.

Note :

1. static initializer get executed during the loading process of a class.
2. static block doesn't have a name.
3. static block doesn't have an return type
4. a programmer cannot explicitly call a static block/static initializer
5. a static block will be executed only once, during the loading of the class.
6. We can have any number of static blocks in one class.
7. static blocks gets executed in top to bottom order.
8. static block gets executed before the execution of main method.

Loading process of a class :

1. class static area is created and can be accessed with the help of class name.

2. all the method blocks are loaded into the method area.
3. if static method is present, the reference of static method will be stored inside the class static area.
4. if static variables is present, memory will be allocated in the class static area and will be assigned with default value.
5. if static initializers are present, all the static initializers get executed in the top to bottom order.

This completes the **Loading Process of a class.**

6. if the class loaded is an **initial class**, then JVM calls `main(String[])`.

Assignment :

1. Difference between static method and static initializer.
2. Can we execute a method before the execution of main method? justify with an example program.
3. Write the loading process of the class 5 times.