

BCA – 401: Java Programming

Rahul Kumar Singh

In today's Class we have discussed on static members and nesting of methods in Java.

Static members in Java:-

Any members of the class with the keyword static is known as static members.

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 member of class can be:

Variable (also known as a class variable)

Method (also known as a class method)

Java static variable:-

If we declare any variable as static, it is known as a static variable.

The static variable can be used to refer to the common property of all objects (which is not unique for each object), for example, the company name of employees, college name of students, etc.

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

Advantages of static variable:-

It makes our program memory efficient (i.e., it saves memory).

Understanding the problem without static variable

```
class Student
{
    int rollno;
    String name;
    String college="LNT";
}
```

Suppose there are 500 students in our 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. If we make it static, this field will get the memory only once.

Example of static variable:-

//Java Program to demonstrate the use of static variable

```
class Student
{
    int rollno;           //instance variable
    String name;
```

```
static String college = "LNT";    //static variable
Student(int r,String n)          //constructor
{
    rollno = r;
    name = n;
}

void display ()    //method to display the values
{
    System.out.println(rollno+" "+name+" "+college);
}
}

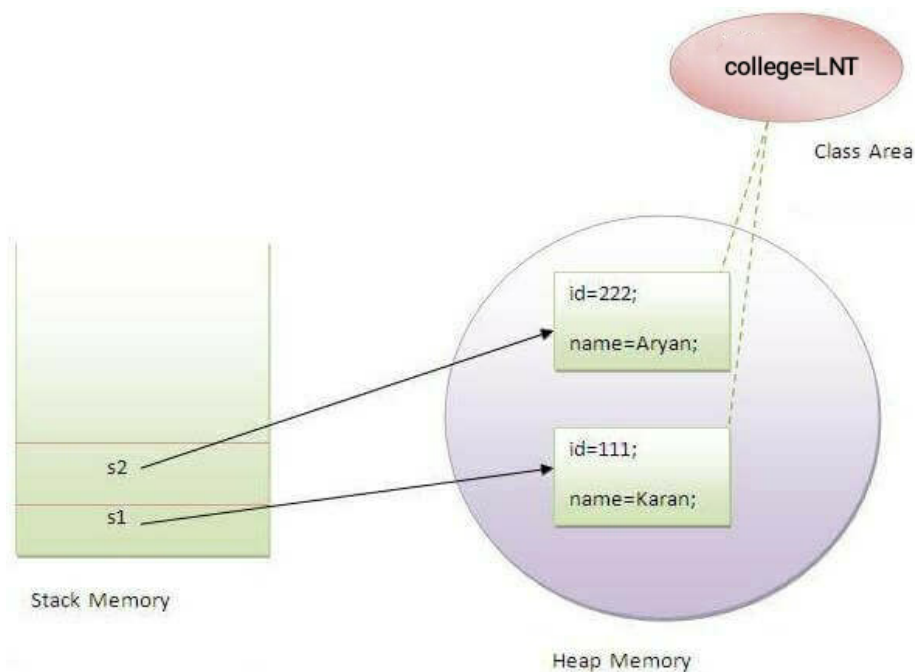
//Test class to show the values of static variable
public class TestStaticVariable
{
    public static void main(String args[])
    {
        Student s1 = new Student(222,"Aryan");
        Student s2 = new Student(111,"Karan");
        s1.display();
        s2.display();
    }
}
```

}

}

222 Aryan LNT

111 Karan LNT



Java static method:-

If you apply static keyword with any method, it is known as static method.

- A static method belongs to the class rather than the object of a class.
- A static method can be invoked without the need for creating an instance of a class.
- A static method can access static data member and can change the value of it.

Example of static method:-

```
class Student
```

```
{
```

```
    int rollno;
```

```
    String name;
```

```
    static String college = "LNT";
```

```
    static void change()
```

```
    {
```

```
        college = "LNT College";
```

```
    }
```

```
    Student(int r, String n)
```

```
    {
```

```
        rollno = r;
```

```
        name = n;
```

```
    }
```

```
    void display()
```

```
    {
```

```
        System.out.println(rollno+" "+name+" "+college);
```

```
    }
```

```
}
```

```
public class TestStaticMethod
{
    public static void main(String args[])
    {
        Student.change();//calling change method
        //creating objects
        Student s1 = new Student(111,"Sajal");
        Student s2 = new Student(222,"Suraj");
        Student s3 = new Student(333,"Ujjawal");
        //calling display method
        s1.display();
        s2.display();
        s3.display();
    }
}
```

Output:-

111 Sajal LNT College

222 Suraj LNT College

333 Ujjawal LNT College

Another example of a static method that performs a normal calculation.

//Java Program to get the cube of a given number using the static method.

```
class Calculate
{
    static int cube(int x)
    {
        return x*x*x;
    }

    public static void main(String args[])
    {
        int result=Calculate.cube(5);
        System.out.println(result);
    }
}
```

Output:-

25

Restrictions for the static method:-

There are two main restrictions for the static method. They are:

- The static method can not use non static data member or call non-static method directly.
- this and super cannot be used in static context.

Example:-

```
class A
{
    int a=40;           //non static
    public static void main(String args[])
    {
        System.out.println(a);
    }
}
```

Output:-

Compile Time Error

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

```
System.out.println(a);
```

^ 1 error

Q) Why is the Java main method static?

Ans) It is because the object is not required to call a static method. If it were a non-static method, JVM creates an object first then call main() method that will lead the problem of extra memory allocation.

Nesting of methods:-

In java, the methods and variables which we create in a class can only be called by using the object of that class or, in case of static methods, we can directly call it by using the name of the class. The methods and variables can be called with the help of the dot operator. But there is a special case that a method can also be called by another method with the help of class name, but the condition is they should be present in the same class.

Syntax:-

```
class Main
{
    method1()
    {
        // statements
    }
    method2()
```

```
{  
    // statements  
    // calling method1() from method2()  
    method1();  
}  
method3()  
{  
    // statements  
    // calling of method2() from method3()  
    method2();  
}  
}
```

Example 1:

In the following example, the class contains two methods namely swap() and add(). The swap() method is used to swap two variables and the add() method is used for adding two numbers. The main() method is calling the add() method with two values to add two numbers and the add() method is calling swap() method for performing swapping of two numbers.

// Java Program implementing the Nesting of Methods

```
public class NestingMethod
{
    public void swap(int x, int y)
    {
        System.out.println( "This is swap method");
        System.out.println("Before swapping:x=" + x + " and "+
"y=" + y);
        // swap the numbers
        int z = x;
        x = y;
        y = z;
        System.out.println("After Swapping:x=" + x + " and "+
"y=" + y);
    }
    public void add(int a, int b)
    {
        System.out.println( "This is add method");
        System.out.println("Before performing operation:a="+ a
+ " and "+ "b=" + b);
        a = a + 10;
        b = b + 12;
```

```

        System.out.println("After operation:a=" + a + " and " +
        "b=" + b);

        // calling of swap() method from add() method
        swap(a, b);
    }

    public static void main(String args[])
    {
        // creating the object of class NestingMethod
        NestingMethod nm = new NestingMethod();
        int a = 20, b = 30;
        // calling of method add() from main method
        nm.add(a, b);
    }
}

```

Output:-

This is add method

Before performing operation:a=20 and b=30

After operation:a=30 and b=42

This is swap method

Before swapping:x=30 and y=42

After Swapping:x=42 and y=30