

Java static variable

If you declare any variable as static, it is known as static variable.

- The static variable can be used to refer to the common property/data of all objects (that is not unique for each object) e.g. company name of employees, college name of students etc.; every instance/object will have the same value for the static variable
- The static variable gets memory only once in class area at the time of class loading.

Advantage of static variable

It makes your program **memory efficient** (i.e it saves memory).

```
class Student{  
    int rollno;           // assume 500 students  
    String name;          // 500 instances (one for each student)  
    static String college="ITS"; // → 500 int + 500 string + 1 string (college)  
}
```

static methods

- A method which belongs to the class and not to the object (or instance)
- A static method can access only static data
- A static method can call only other static methods and cannot call a non-static method from it.
- A static method can be accessed directly by the class name and doesn't need any object.

Example:

the method `Math.sqrt()` is a static method and can be called without an object.

this.

The key word "this" can be used inside the Method or constructor of Class. It(this) works as a reference to the current Object whose Method or constructor is being invoked. The this **keyword** can be used to refer to any member of the current object from within an instance Method or a constructor.

Key word "this" is used when variables of the same name are used in a class. "this" can be used even if the names are different but not necessary and not usually practiced.

Example:

```
class ExampleFor_This {
    int var = 5;

    void method1 () {
        var = 100;    // different from the var (=5) that belongs to the object
        System.out.println(" var = " + var);
        System.out.println(" this.var = " + this.var);
        // this.var is the "var" that belongs to the object
    }

    public static void main (String [] args) {
        ExampleFor_This obj_1 = new ExampleFor_This();
        obj_1.method1();
    }
}
```