# Static keyword

The static keyword can be used for

1. Variable (also known as a class variable)
2. Method (also known as a class method)
3. Block
4. Nested class

## 1) Java static variable

If you 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.

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

**class** Student{

**int** rollno;

     String name;

     String college="MGIT";

}

* Suppose there are 500 students in the college, now all instance data members will get memory each time when the object is created.
* All students have unique rollno and name, so instance data member is good in such case.
* Here, "college" refers to the common property of all [objects](https://www.javatpoint.com/object-and-class-in-java).
* If we make it static, this field will get the memory only once.

//Java Program to demonstrate the use of static variable

**class** Student{

**int** rollno;//instance variable

   String name;  //instance variable

**static** String college ="MGIT";//static variable

   //constructor

   Student(**int** r, String n){

   rollno = r;

   name = n;

   }

   //method to display the values

**void** display (){System.out.println(rollno+" "+name+" "+college);}

}

**class** TestStaticVariable1{

**public** **static** **void** main(String args[]){

 Student s1 = **new** Student(111,"sree");

 Student s2 = **new** Student(222,"Adi");

 //we can change the college of all objects by the single line of code

 //Student.college="CBIT";

 s1.display();

S2

S1

Rollno

name

Rollno

name

 s2.display();

 }

}   class area

# stack heap memory

## 2) 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 i.e without creating an object.
* A static method can access static data member and can change the value of it.

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

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

//Java Program to demonstrate the use of a static method.

class Student{

int rollno;

String name;

static String college = "MGIT";

//static method to change the value of static variable

static void change(){

college = "CBIT";

//rollno=525; //non-static variable cannot be accessed from static method

}

Student(int r, String n){

rollno = r;

name = n; }

void display(){System.out.println(rollno+" "+name+" "+college);}

}

class TestStaticMethod{

public static void main(String args[]){

Student.change();

// change();

Student s1 = new Student(111,"sree");

Student s2 = new Student(222,"Aadi");

s1.display();

s2.display();

} }

### Why is the Java main method static?

## 3) Java static block

* in order to initialize your **static variables**, you can declare a static block that gets executed exactly once, when the class is first loaded.

// Java program to demonstrate use of static blocks

class StaticBlockDemo

{

// static variable

static int a = 10;

static int b;

// static block

static {

System.out.println("Static block initialized.");

b = a \* 4;

}

public static void main(String[] args)

{

System.out.println("from main");

System.out.println("Value of a : "+a);

System.out.println("Value of b : "+b);

}

}