## What is Java

**--------------------------------**

**Java is a high level, robust, object-oriented and secure programming language.**

**Java static keyword**

---------------------------

The static keyword in Java is used for memory management mainly.

We can apply java static keyword with variables, methods, blocks and nested class.

The static keyword belongs to the class than an instance of the class.

**The static can be:**

-------------------------------

Variable (also known as a class variable)

Method (also known as a class method)

Block

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.

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

### Advantages of static variable

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

#### Understanding the problem without static variable

-------------------------------------------------------------------------

**class** Student{

**int** rollno;

String name;

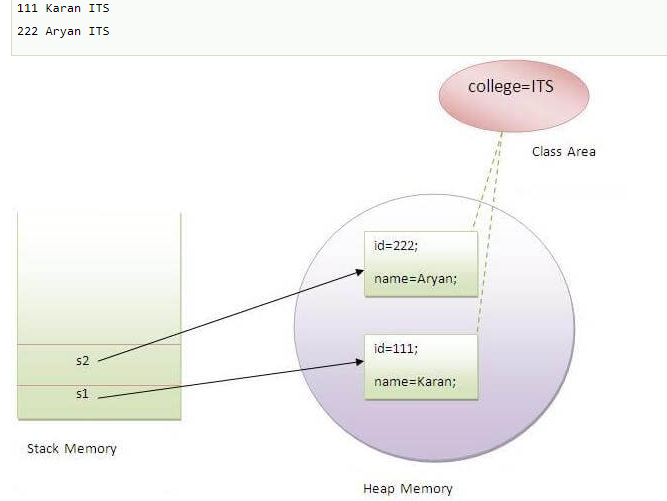
String college="ITS";

}

Suppose there are 500 students in my 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

1. //Java Program to demonstrate the use of static variable
2. **class** Student{
3. **int** rollno;//instance variable
4. String name;
5. **static** String college ="ITS";//static variable
6. //constructor
7. Student(**int** r, String n){
8. rollno = r;
9. name = n;
10. }
11. //method to display the values
12. **void** display (){System.out.println(rollno+" "+name+" "+college);}
13. }
14. //Test class to show the values of objects
15. **public** **class** TestStaticVariable1{
16. **public** **static** **void** main(String args[]){
17. Student s1 = **new** Student(111,"Karan");
18. Student s2 = **new** Student(222,"Aryan");
19. //we can change the college of all objects by the single line of code
20. //Student.college="BBDIT";
21. s1.display();
22. s2.display();
23. }
24. }



### Program of the counter without static variable

In this example, we have created an instance variable named count which is incremented in the constructor. Since instance variable gets the memory at the time of object creation, each object will have the copy of the instance variable. If it is incremented, it won't reflect other objects. So each object will have the value 1 in the count variable.

1. //Java Program to demonstrate the use of an instance variable
2. //which get memory each time when we create an object of the class.
3. **class** Counter{
4. **int** count=0;//will get memory each time when the instance is created
6. Counter(){
7. count++;//incrementing value
8. System.out.println(count);
9. }
11. **public** **static** **void** main(String args[]){
12. //Creating objects
13. Counter c1=**new** Counter();
14. Counter c2=**new** Counter();
15. Counter c3=**new** Counter();
16. }
17. }

**[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=Counter)**

Output:

1  
1  
1

### Program of counter by static variable

As we have mentioned above, static variable will get the memory only once, if any object changes the value of the static variable, it will retain its value.

1. //Java Program to illustrate the use of static variable which
2. //is shared with all objects.
3. **class** Counter2{
4. **static** **int** count=0;//will get memory only once and retain its value
6. Counter2(){
7. count++;//incrementing the value of static variable
8. System.out.println(count);
9. }
11. **public** **static** **void** main(String args[]){
12. //creating objects
13. Counter2 c1=**new** Counter2();
14. Counter2 c2=**new** Counter2();
15. Counter2 c3=**new** Counter2();
16. }
17. }

**[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=Counter2)**

Output:

1  
2  
3

## 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.
* A static method can access static data member and can change the value of it.

### Example of static method

1. //Java Program to demonstrate the use of a static method.
2. **class** Student{
3. **int** rollno;
4. String name;
5. **static** String college = "ITS";
6. //static method to change the value of static variable
7. **static** **void** change(){
8. college = "BBDIT";
9. }
10. //constructor to initialize the variable
11. Student(**int** r, String n){
12. rollno = r;
13. name = n;
14. }
15. //method to display values
16. **void** display(){System.out.println(rollno+" "+name+" "+college);}
17. }
18. //Test class to create and display the values of object
19. **public** **class** TestStaticMethod{
20. **public** **static** **void** main(String args[]){
21. Student.change();//calling change method
22. //creating objects
23. Student s1 = **new** Student(111,"Karan");
24. Student s2 = **new** Student(222,"Aryan");
25. Student s3 = **new** Student(333,"Sonoo");
26. //calling display method
27. s1.display();
28. s2.display();
29. s3.display();
30. }
31. }

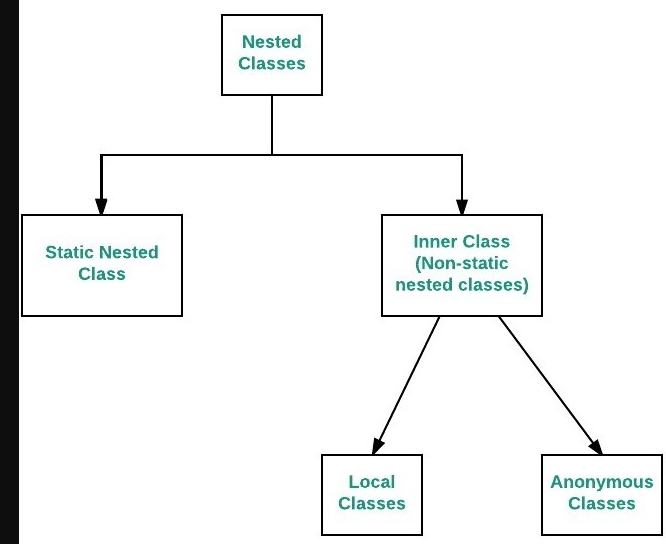
**[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=TestStaticMethod)**

Output:111 Karan BBDIT  
 222 Aryan BBDIT  
 333 Sonoo BBDIT

|  |
| --- |
| [**next →**](https://www.javatpoint.com/this-keyword)**[← prev](https://www.javatpoint.com/java-constructor)** Java static keyword  1. [Static variable](https://www.javatpoint.com/static-keyword-in-java#staticv) 2. [Program of the counter without static variable](https://www.javatpoint.com/static-keyword-in-java#staticvcounter1) 3. [Program of the counter with static variable](https://www.javatpoint.com/static-keyword-in-java#staticvcounter2) 4. [Static method](https://www.javatpoint.com/static-keyword-in-java#staticm) 5. [Restrictions for the static method](https://www.javatpoint.com/static-keyword-in-java#staticmr) 6. [Why is the main method static?](https://www.javatpoint.com/static-keyword-in-java#staticwhymain) 7. [Static block](https://www.javatpoint.com/static-keyword-in-java#staticblock) 8. [Can we execute a program without main method?](https://www.javatpoint.com/static-keyword-in-java#staticwithoutmain)   The **static keyword** in Java is used for memory management mainly. We can apply java static keyword with variables, methods, blocks and nested class. The static keyword belongs to the class than an instance of the class.  The static can be:   1. Variable (also known as a class variable) 2. Method (also known as a class method) 3. Block 4. Nested class   Static in Java 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. * The static variable gets memory only once in the class area at the time of class loading.  Advantages of static variable It makes your program **memory efficient** (i.e., it saves memory). Understanding the problem without static variable  1. **class** Student{ 2. **int** rollno; 3. String name; 4. String college="ITS"; 5. }   Suppose there are 500 students in my 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. Java static property is shared to all objects.Example of static variable  1. //Java Program to demonstrate the use of static variable 2. **class** Student{ 3. **int** rollno;//instance variable 4. String name; 5. **static** String college ="ITS";//static variable 6. //constructor 7. Student(**int** r, String n){ 8. rollno = r; 9. name = n; 10. } 11. //method to display the values 12. **void** display (){System.out.println(rollno+" "+name+" "+college);} 13. } 14. //Test class to show the values of objects 15. **public** **class** TestStaticVariable1{ 16. **public** **static** **void** main(String args[]){ 17. Student s1 = **new** Student(111,"Karan"); 18. Student s2 = **new** Student(222,"Aryan"); 19. //we can change the college of all objects by the single line of code 20. //Student.college="BBDIT"; 21. s1.display(); 22. s2.display(); 23. } 24. }   **[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=TestStaticVariable1)**  Output:  111 Karan ITS 222 Aryan ITS  Static Variable Program of the counter without static variable In this example, we have created an instance variable named count which is incremented in the constructor. Since instance variable gets the memory at the time of object creation, each object will have the copy of the instance variable. If it is incremented, it won't reflect other objects. So each object will have the value 1 in the count variable.   1. //Java Program to demonstrate the use of an instance variable 2. //which get memory each time when we create an object of the class. 3. **class** Counter{ 4. **int** count=0;//will get memory each time when the instance is created 6. Counter(){ 7. count++;//incrementing value 8. System.out.println(count); 9. } 11. **public** **static** **void** main(String args[]){ 12. //Creating objects 13. Counter c1=**new** Counter(); 14. Counter c2=**new** Counter(); 15. Counter c3=**new** Counter(); 16. } 17. }   **[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=Counter)**  Output:  1 1 1 Program of counter by static variable As we have mentioned above, static variable will get the memory only once, if any object changes the value of the static variable, it will retain its value.   1. //Java Program to illustrate the use of static variable which 2. //is shared with all objects. 3. **class** Counter2{ 4. **static** **int** count=0;//will get memory only once and retain its value 6. Counter2(){ 7. count++;//incrementing the value of static variable 8. System.out.println(count); 9. } 11. **public** **static** **void** main(String args[]){ 12. //creating objects 13. Counter2 c1=**new** Counter2(); 14. Counter2 c2=**new** Counter2(); 15. Counter2 c3=**new** Counter2(); 16. } 17. }   **[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=Counter2)**  Output:  1 2 3 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. * A static method can access static data member and can change the value of it.  Example of static method  1. //Java Program to demonstrate the use of a static method. 2. **class** Student{ 3. **int** rollno; 4. String name; 5. **static** String college = "ITS"; 6. //static method to change the value of static variable 7. **static** **void** change(){ 8. college = "BBDIT"; 9. } 10. //constructor to initialize the variable 11. Student(**int** r, String n){ 12. rollno = r; 13. name = n; 14. } 15. //method to display values 16. **void** display(){System.out.println(rollno+" "+name+" "+college);} 17. } 18. //Test class to create and display the values of object 19. **public** **class** TestStaticMethod{ 20. **public** **static** **void** main(String args[]){ 21. Student.change();//calling change method 22. //creating objects 23. Student s1 = **new** Student(111,"Karan"); 24. Student s2 = **new** Student(222,"Aryan"); 25. Student s3 = **new** Student(333,"Sonoo"); 26. //calling display method 27. s1.display(); 28. s2.display(); 29. s3.display(); 30. } 31. }   **[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=TestStaticMethod)**  Output:111 Karan BBDIT  222 Aryan BBDIT  333 Sonoo BBDIT   Restrictions for the static method There are two main restrictions for the static method. They are:   1. The static method can not use non static data member or call non-static method directly. 2. this and super cannot be used in static context. 3. **class** A{ 4. **int** a=40;//non static 6. **public** **static** **void** main(String args[]){ 7. System.out.println(a); 8. } 9. }   **[Test it Now](http://www.javatpoint.com/opr/test.jsp?filename=A)**  Output:Compile Time 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. |

### **Static Nested Class**

------------------------------



// Java program to demonstrate accessing

// a static nested class

// outer class

class OuterClass

{

// static member

static int outer\_x = 10;

// instance(non-static) member

int outer\_y = 20;

// private member

private static int outer\_private = 30;

// static nested class

static class StaticNestedClass

{

void display()

{

// can access static member of outer class

System.out.println("outer\_x = " + outer\_x);

// can access display private static member of outer class

System.out.println("outer\_private = " + outer\_private);

// The following statement will give compilation error

// as static nested class cannot directly access non-static membera

// System.out.println("outer\_y = " + outer\_y);

}

}

}

// Driver class

public class StaticNestedClassDemo

{

public static void main(String[] args)

{

// accessing a static nested class

OuterClass.StaticNestedClass nestedObject = new OuterClass.StaticNestedClass();

nestedObject.display();

}

}