**Variables**

**1.**What is meant by variable?  
**Ans:**Variables are locations in memory that can hold values. Before assigning any value to a variable, it must be declared. java.lang package.

**2.**What are the kinds of variables in Java and what are their uses?  
**Ans:** Java has three kinds of variables namely:  
a)The instance variable  
b)The local variable  
c)The class variable.  
Local variables are used inside blocks as counters or in methods as temporary variables and are used to store information needed by a single method.   
Instance variables are used to define attributes or the state of a particular object and are used to store information needed by multiple methods in the objects.  
Class variables are global to a class and to all the instances of the class and are useful for communicating between different objects of all the same class or keeping track of global states.

**3.** How are the variables declared?  
**Ans:** Variables can be declared anywhere in the method definition and can be initialized during their declaration.They are commonly declared before usage at the beginning of the definition. Variables with the same data type can be declared together. Local variables must be given a value before usage.

**4.** What are variable types?   
**Ans:**Variable types can be any data type that java supports, which includes the eight primitive data types, the name of a class or interface and an array.

**5.** How do you assign values to variables?  
**Ans:**Values are assigned to variables using the assignment operator =.

**6.** What is a literal? How many types of literals are there?   
**Ans:**A literal represents a value of a certain type where the type describes how that value behaves. There are different types of literals namely:   
a)number literals  
b)character literals  
c)boolean literals  
d)string literals and etc.

**7.** What is final varaible?   
**Ans:**If a variable is declared as final variable, then you can not change its value. It becomes constant.

**8.** What is static variable?   
**Ans:**Static variables are shared by all instances of a class.

# Java Programming/Keywords/transient

< [Java Programming](https://en.wikibooks.org/wiki/Java_Programming)‎ | [Keywords](https://en.wikibooks.org/wiki/Java_Programming/Keywords)

**transient** is a Java keyword which marks a member variable not to be serialized when it is persisted to streams of bytes. When an object is transferred through the network, the object needs to be 'serialized'. Serialization converts the object state to serial bytes. Those bytes are sent over the network and the object is recreated from those bytes. Member variables marked by the java **transient** keyword are not transferred; they are lost intentionally.

Syntax:

[**private**](https://en.wikibooks.org/wiki/Java_Programming/Keywords/private) **transient** <member-variable>;

or

**transient** [**private**](https://en.wikibooks.org/wiki/Java_Programming/Keywords/private) <member-variable>;

For example:

|  |  |
| --- | --- |
| Computer code | **public** **class** **Foo** **implements** Serializable  {  **private** String saveMe;  **private** **transient** String dontSaveMe;  **private** **transient** String password;  *//...*  } |

# Top 14 Java Interview Questions on Static keyword

#### 1.what is static in java?

* Static is a keyword in java.
* One of the Important keyword in java.
* Clear understanding of static keyword is required to build projects.
* We have static variables, static methods , static blocks.
* Static means class level.

#### 2.Why we use static keyword in java?

* Static keyword is mainly used for memory management.
* Static variables get memory when class loading itself.
* Static variables can be used to point common property all objects.

#### 3.What is static variable in java?

* Variables declared with static keyword is known as static variables.
* Static variables gets memory on class loading.
* Static variables are class level.
* If we change any static variable value using a particular object then its value changed for all objects means it is common to every object of that class.
* static int a,b;

1. package com.instanceofjavastatic;
2. class StaticDemo{
4. static int a=40;
5. static int b=60;
7. }

* We can not declare local variables as static it leads to compile time error "illegal start of expression".
* Because being static variable it must get memory at the time of class loading, which is not possible to provide memory to local variable at the time of class loading.

1. package com.instanceofjava;
2. class StaticDemo{
4. static int a=10;
5. static int b=20;
6. public static void main(String [] args){
8. //local variables should not be static
9. static int a=10;// compile time error: illegal start of expression
10. }
11. }

 Read more : [Explain about static variables in java?](http://www.instanceofjava.com/2015/04/static-variable-method-block-in-java.html)

#### 4. what is static method in java with example

* Method which is having static in its method definition is known as static method.

1. static void show(){
3. }

* JVM will not call these static methods automatically. Develioper needs to call these static methods from main method or static block or variable initialization.
* Only Main method will b called by JVM automatically.
* We can call these static methods by using class name itself no need to create object.

Read more @[what is static method in java](http://www.instanceofjava.com/2015/04/static-methods-in-java-example.html)

#### 5.what is static block in java?

* Static blocks are the blocks which will have braces and with static keyword.
* These static blocks will be called when JVM loads the class.
* Static blocks are the blocks with static keyword.
* Static blocks wont have any name in its prototype.
* Static blocks are class level.
* Static block will be executed only once.
* No return statements.
* No arguments.
* No this or super keywords supported.

* 1. static{
  3. }

 Read more @ [Static blocks in java with example programs](http://www.instanceofjava.com/2015/05/static-block-in-java-example.html)

#### 6.What is the need of static block?

* Static blocks will be executed at the time of class loading.
* So if you want any logic that needs to be executed at the time of class loading that logic need to place inside the static block so that it will be executed at the time of class loading.

#### 7.Why main method is static in java?

* To execute main method without creating object then the main method should be static so that JVM will call main method by using class name itself.

#### 8.Can we overload static methods in java?

* Yes we can overload static methods in java.

The answer is **Yes**. We can have two or more static methods with same name, but differences in input parameters. For example, consider the following Java program.

public class Test {

public static void foo() {

System.out.println("Test.foo() called ");

}

public static void foo(int a) {

System.out.println("Test.foo(int) called ");

}

public static void main(String args[])

{

Test.foo(); //Test.foo() called printed

Test.foo(10); //Test.foo(int) called printed

}

}

**Can we overload methods that differ only by static keyword?**

We **cannot** overload two methods in Java if they differ only by static keyword (number of parameters and types of parameters is same).

#### 9.Can we override static methods in java?

* NO we can not override static methods in java.

Read more  @ [Can we override static methods in java](http://www.instanceofjava.com/2015/06/can-we-override-static-methods-in-java.html)

#### 10.Can we write static public void main(String [] args)?

* Yes we can define main method like static public void main(String[] args){}
* Order of modifiers we can change.

1. package instanceofjava;
2. public MainDemo{
3. **static public**void main(String [] args){
5. }
6. }

**11.Can we call super class static methods from sub class?**

* Yes we can call super class static methods from sub class.
* **Read more @ [Can we call super class static methods from sub class?](http://www.instanceofjava.com/2016/02/can-we-call-superclass-static-method.html)**

**[12.Calling static method from non static method in java](http://www.instanceofjava.com/2016/09/calling-static-method-from-non-static.html)**  
  
**[13.How to call non static method from static method java](http://www.instanceofjava.com/2016/09/calling-non-static-method-from-static.html)**

### ****1. What is****[static keyword in Java](http://www.javainterviewpoint.com/use-of-static-keyword-in-java/)****?****

Static is a Non Access Modifier. Static can be applied to variable, method, nested class and initialization blocks (static block).

**What is a static variable ?**

* A **Static variable** gets memory allocated only once during the time of class loading.
* All the instance of the class share the same copy of the variable, a static variable can be accessed directly by calling **“<<ClassName>>.<<VariableName>>”**without need to create instance for the class.
* value of a static variable will be common for all instances

public class StaticVariableExample

{

static int a =10;

public static void main(String args[]){

StaticVariableExample s1 = new StaticVariableExample();

StaticVariableExample s2 = new StaticVariableExample();

System.out.println("s1.a value :"+s1.a);

System.out.println("s2.a value :"+s2.a);

//Change s1 a value alone

s1.a=20;

System.out.println("s1.a value :"+s1.a);

System.out.println("s2.a value :"+s2.a);

}

}

**Output will be**  
s1.a value :10  
s2.a value :10  
s1.a value :20  
s2.a value :20

* **Local variables** cannot be assigned as static it will throw compile time error **“illegal start of expression”**, as the memory cannot be assigned during class load.

**3. What is a static method ?**

* A **static method** belongs to **class** rather than object. It can be called directly by using the classname**“<<ClassName>>.<<MethodName>>”**
* A static method can access **static varaibles**directly and it cannot access **non-static**variables and can only call a **static method**directly and it cannot call a **non-static** method from it.
* Only the **main() method** which is static will be called by the JVM automatically, Not all the static method will be called automatically.

**4. Can a static block exist without a main() method ?**

**Yes**. You can have **static block** alone in the class without a main method.

**5. Can we**[**Overload**](http://www.javainterviewpoint.com/java-method-overloading-example/)**static methods in Java**

**Yes**, you can overload a static method in Java. [**Read More..**](http://www.javainterviewpoint.com/can-we-overload-static-methods-in-java/)

**6. Can we**[**Override**](http://www.javainterviewpoint.com/what-is-method-overriding-in-java/)**static methods in Java**

**No**, you cannot override a static method in [**Java**](https://www.javainterviewpoint.com/category/core-java/) as there will not be any [**Run-time Polymorphism**](http://www.javainterviewpoint.com/run-time-polymorphism-and-compile-time-in-java/) happening. [**Read More..**](http://www.javainterviewpoint.com/can-we-override-static-methods-in-java/)

**7. Why main() method is declared as static ?**

If our **main()** method is not declared as static then the JVM has to create object first and call which causes the problem of having extra memory allocation.

**8. What is a static block ?**

* A **static block**, is a block of code inside a Java class that will be executed when a class is first loaded in to the JVM. Mostly the static block will be used for initializing the variables.
* Static block will be called only one while loading and it **cannot have any return type**, or any keywords (**this** or **super**).

class test

{

int val;

static{

val = 100;

}

}

**9.  Can we have multiple static blocks in our code ?**

**Yes**, we can have more than one static block in our code. It will be executed in the same order it is written.

**10. What is a static class ?**

* In Java only **nested classes** are allowed to be declared as **static**, a **top level** class cannot be declared as static.
* Even though **static classes** are **nested** inside a class, they **doesn’t need the reference** of the outer class they act like outer class only. [**Read More..**](http://www.javainterviewpoint.com/use-of-static-keyword-in-java/)

**11. Can constructors be static in Java?**

In general a static method means that “The Method belong to class and not to any particular object” but a constructor is always invoked with respect to an object, so it makes no sense for a constructor to be **static**.

**12. Why**[**abstract method**](http://www.javainterviewpoint.com/abstract-class-java/)**cannot be static in Java?**

Suppose when you have a concrete method in a abstract class then that method can be static. Suppose we have a class like below

public class AbstractTest

{

static void disp()

{

System.out.println("disp of static method");

}

}

Then the **disp()** can be access by ***“AbstractTest.disp()”***  
However, for the same reason cannot be applied when you declare a **static** **method** to be **abstract**. Since static method can be called directly, making it abstract would make it possible to call an **undefined method** which is of no use, hence it is not allowed.

**13. Can**[**Interface in Java**](http://www.javainterviewpoint.com/interface-java/)**have static methods in it?**

**No,** Interface cannot have static methods in it because all methods are [**implicitly abstract**](http://docs.oracle.com/javase/specs/jls/se7/html/jls-9.html#jls-9.4). This is why an interface cannot have a static method.

**14. Can abstract class have static variable in it?**

**Yes,** an abstract class can have static variables in it.

Filed Under: [**Core Java**](http://www.javainterviewpoint.com/category/core-java/), [**Java**](http://www.javainterviewpoint.com/category/java/), [**Java Interview**](http://www.javainterviewpoint.com/category/java-interview/)Tagged With: [**Java**](http://www.javainterviewpoint.com/tag/java/), [**Java Interview Questions**](http://www.javainterviewpoint.com/tag/java-interview-questions/), [**Static**](http://www.javainterviewpoint.com/tag/static/), [**Static Keyword**](http://www.javainterviewpoint.com/tag/static-keyword/)

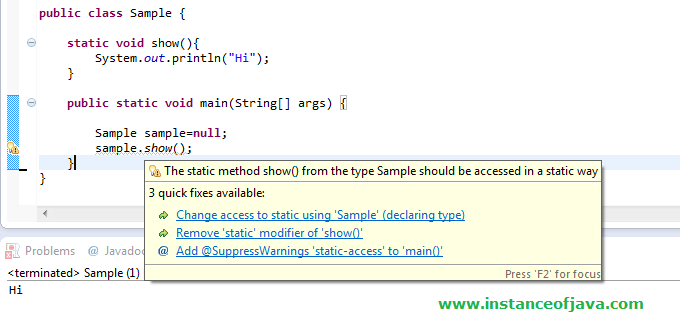
# [Can we call super class static method from subclass in java](http://www.instanceofjava.com/2016/02/can-we-call-superclass-static-method.html)

Posted by: InstanceOfJava Posted date: **Feb 16, 2016** / comment : 0

* If you want to call static method of a class we can call directly from another static method or by using its class name we can call static method of that class.
* let us see a program on how to call a static method of a class

1. package com.instanceofjava;
2. public class Sample{
3. public static void show(){
5. System.out.println("show() method called");
7. }
8. public static void main(String args[]){
10. show();
11. Sample.show();
13. }
14. }

* We can also call static method like this but this is not recommended.

[](https://4.bp.blogspot.com/-z8iqYaXuRYU/Vsat-b_gN6I/AAAAAAAAAlk/vSdqsrhuz0g/s1600/static+in+java.png)

**Output:**

1. show() method called
2. show() method called

* Now our question is can we call super class static method from sub class?
* Yes we can call super class static method inside sub class using super\_class\_method();
* We can also call super class static method using Sub\_class\_name.superclass\_staticMethod()

1. package com.instanceofjava;
3. public class SuperDemo{
4. public static void show(){
6. System.out.println("Super class show() method called");
8. }
10. }

1. package com.instanceofjava;
2. public class SubDemo extends SuperDemo{
3. public void print(){
5. System.out.println("Sub class print() method called");
7. }
8. public static void main(String args[]){
10. SuperDemo.show();
11. SubDemo.show();
12. }
13. }

**Output:**

1. Super class show() method called
2. Super class show() method called

* If the same static method defined in sub class also then we can not call super class method using sub class name if we call them sub class static method will be executed.

1. package com.instanceofjava;
3. public class SuperDemo{
4. public static void show(){
6. System.out.println("Super class show() method called");
8. }
10. }
11. package com.instanceofjava;
12. public class SubDemo extends SuperDemo{
13. public static void show(){
15. System.out.println("Sub class show() method called");
17. }
18. public static void main(String args[]){
20. SuperDemo.show();
21. SubDemo.show();
22. }
23. }

**Output:**

1. Super class show() method called
2. Sub class show() method called