# **Static Variable, Methods and Classes**

**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?**

Static variables belong to the class and static variables are shared by all instances of a class.

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.

**How do we access static members in java?**

Instance variables and instance methods can be accessed using reference variable . But to access static variables or static methods we use Class name in java.

**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.

**Can Static methods access instance variables in java?**

No. Instance variables can’t be accessed in static methods. When we try to access instance variable in static method we get compilation error. The error is as follows:

“Cannot make a static reference to the non static field name”

Though Static methods cannot access the instance variables directly, They can access them using instance handler.

**When are static variables loaded in memory?**

They are loaded at runtime when the respective Class is loaded.

**Can we serialize static variables?**

No. Only Object and its members are serialized. Static variables are shared variables and doesn't correspond to a specific object.

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

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

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

**Yes**, you can overload a static method in Java.

**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.

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

main() method is called by the JVM even before the instantiation of the class hence it 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.

**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 once while loading and it **cannot have any return type**, or any keywords (**this** or **super**).

class test

{

int val;

static{

val = 100;

}

}

**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.

**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 don’t need the reference of the outer class they act like outer class only.

**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**.

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

Suppose when you have a concrete method in an 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.

**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.

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

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

**non-static method cannot be referenced from a static context?**

public class Test

{

/\*\* Non Static main method with String[] args\*\*/

public static void main(String[] args)

{

welcome();

}

void welcome()

{

System.out.println("Welcom to JavaInterviewPoint");

}

}

The welcome() method which we tried calling is an instance-level method, we do not have an instance to call it . static methods belong to the class, non-staticmethods belong to instances of the class and hence it throws the error ” non-static method cannot be referenced from a static context “.

**Can a class be declared as static?**

We can not declare top level class as static, but only inner class can be declared static.

public class Test

{

static class InnerClass

{

public static void InnerMethod()

{ System.out.println("Static Inner Class!"); }

}

public static void main(String args[])

{

Test.InnerClass.InnerMethod();

}

}

//output: Static Inner Class!

**When will you define a method as static?**

When a method needs to be accessed even before the creation of the object of the class then we should declare the method as static.

**What are the restriction imposed on a static method or a static block of code?**

A static method should not refer to instance variables without creating an instance and cannot use "this" operator to refer the instance.

**I want to print "Hello" even before main() is executed. How will you acheive that?**

Print the statement inside a static block of code. Static blocks get executed when the class gets loaded into the memory and even before the creation of an object. Hence it will be executed before the main() method. And it will be executed only once.

**package** intquestions;

**class** testclass

{

**static** {

System.***out***.println("hello");

}

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

}

}

This will print the result as “hello”.

**What is the importance of static variable?**

static variables are class level variables where all objects of the class refer to the same variable. If one object changes the value then the change gets reflected in all the objects.

**Can we declare a static variable inside a method?**

Static varaibles are class level variables and they can't be declared inside a method. If declared, the class will not compile.

**What is the difference between a static and a non-static inner class?**

A non-static inner class may have object instances that are associated with instances of the class's outer class. A static inner class does not have any object instances.

**What happens to a static variable that is defined within a method of a class ?**

Can't do it. You'll get a compilation error.

**How many static initializers can you have?**

As many as you want, but the static initializers and class variable initializers are executed in textual order and may not refer to class variables declared in the class whose declarations appear textually after the use, even though these class variables are in scope.

**Why main() method is public, static and void in java ?**

public : “public” is an access specifier which can be used outside the class. When main method is declared public it means it can be used outside class.

static : To call a method we require object. Sometimes it may be required to call a method without the help of object. Then we declare that method as static. JVM calls the main() method without creating object by declaring keyword static.

void : void return type is used when a method does’nt return any value . main() method does’nt return any value, so main() is declared as void.

**Explain the scope or life time of class variables or static variables?**

Static variables do not belong to instances of the class. We can access static fields even before instantiating the class. Static variable remain in memory till the life time of application.

**Explain about static imports in java?**

From Java 5.0 we can import static variables in to source file. Importing static member to source file is referred as static import. The advantage of static import is we can access static variables without class or interface name.

Syntax :

import static packagename.classname.staticvariablename;

Ex : import static com.abc.Employee.eno;

To import all static variables from a class in to our source file we use \*.

import static com.abc.Employee.\*