

Static class in Java [source: <https://www.geeksforgeeks.org>]

Can a class be static in Java?

The answer is YES, we can have static class in java. In java, we have static instance variables as well as static methods and also static block. Classes can also be made static in Java.

Java allows us to define a class within another class. Such a class is called a nested class. The class which enclosed nested class is known as Outer class. In java, we can't make Top level class static. Only nested classes can be static.

What are the differences between static and non-static nested classes?

Following are major differences between static nested class and non-static nested class. Non-static nested class is also called Inner Class.

1) Nested static class doesn't need reference of Outer class, but Non-static nested class or Inner class requires Outer class reference.

2) Inner class (or non-static nested class) can access both static and non-static members of Outer class. A static class cannot access non-static members of the Outer class. It can access only static members of Outer class.

3) An instance of Inner class cannot be created without an instance of outer class and an Inner class can reference data and methods defined in Outer class in which it nests, so we don't need to pass reference of an object to the constructor of the Inner class. For this reason, Inner classes can make program simple and concise.

```

/* Java program to demonstrate how to implement static and non-static
   classes in a java program. */

class OuterClass{
    private static String msg = "The Good The Bad & the Ugly";

    // Static nested class
    public static class NestedStaticClass{

        // Only static members of Outer class are directly accessible in nested
        // static class
        public void printMessage()
        {

            // Try making 'msg' a non-static variable, you will get compiler error
            System.out.println("Message from nested static class: " + msg);
        }
    }

    // non-static nested class - also called Inner class
    public class InnerClass{

        // Both static and non-static members of Outer class are accessible in
        // this Inner class
        public void display()
        {
            System.out.println("Message from non-static nested class: "+ msg);
        }
    }
}

class Main
{
    // How to create instance of static and non-static nested class?
    public static void main(String args[]){

        // create instance of nested Static class
        OuterClass.NestedStaticClass printer = new OuterClass.NestedStaticClass();

        // call non-static method of nested static class
        printer.printMessage();

        // In order to create instance of Inner class we need an Outer class
        // instance. Let us create Outer class instance for creating
        // non-static nested class
        OuterClass outer = new OuterClass();
        OuterClass.InnerClass inner = outer.new InnerClass();

        // calling non-static method of Inner class
        inner.display();

        // we can also combine above steps in one step to create instance of
        // Inner class
        OuterClass.InnerClass innerObject = new OuterClass().new InnerClass();

        // similarly we can now call Inner class method
        innerObject.display();
    }
}

```

Output:

Message from nested static class: The Good The Bad & the Ugly

Message from non-static nested class: The Good The Bad & The Ugly

Message from non-static nested class: The Good The Bad & The Ugly