Static constructors are not inherited, and cannot be called directly.

The static constructor for a class executes at most once in a given application domain. The execution of a static constructor is triggered by the first of the following events to occur within an application domain:

* An instance of the class is created.
* Any of the static members of the class are referenced.

If a class contains the Main method in which execution begins, the static constructor for that class executes before the Main method is called. If a class contains any static fields with initializers, those initializers are executed in textual order immediately prior to executing the static constructor.

The example

using System;

class Test

{

static void Main() {

A.F();

B.F();

}

}

class A

{

static A() {

Console.WriteLine("Init A");

}

public static void F() {

Console.WriteLine("A.F");

}

}

class B

{

static B() {

Console.WriteLine("Init B");

}

public static void F() {

Console.WriteLine("B.F");

}

}

must produce the output:

Init A

A.F

Init B

B.F

because the execution of A's static constructor is triggered by the call to A.F, and the execution of B's static constructor is triggered by the call to B.F.