The **readonly** keyword is a modifier that you can use on fields. When a field declaration includes a **readonly** modifier, assignments to the fields introduced by the declaration can only occur as part of the declaration or in a constructor in the same class.

You can assign a value to a readonly field only in the following contexts:

* When the variable is initialized in the declaration, for example:
* public readonly int y = 5;
* For an instance field, in the instance constructors of the class that contains the field declaration, or for a static field, in the static constructor of the class that contains the field declaration. These are also the only contexts in which it is valid to pass a readonly field as an **out** or **ref** parameter.

**Example**

// cs\_readonly\_keyword.cs

// Readonly fields

using System;

public class ReadOnlyTest

{

class MyClass

{

public int x;

public readonly int y = 25; // Initialize a readonly field

public readonly int z;

public MyClass()

{

z = 24; // Initialize a readonly instance field

}

public MyClass(int p1, int p2, int p3)

{

x = p1;

y = p2;

z = p3;

}

}

public static void Main()

{

MyClass p1= new MyClass(11, 21, 32); // OK

Console.WriteLine("p1: x={0}, y={1}, z={2}" , p1.x, p1.y, p1.z);

MyClass p2 = new MyClass();

p2.x = 55; // OK

Console.WriteLine("p2: x={0}, y={1}, z={2}" , p2.x, p2.y, p2.z);

}

}

**Output**

p1: x=11, y=21, z=32

p2: x=55, y=25, z=24

In the preceding example, if you use a statement like this:

p2.y = 66; // Error

you will get the compiler error message:

The left-hand side of an assignment must be an l-value

which is the same error you get when you attempt to assign a value to a constant.

**Note**The **readonly** keyword is different from the [const](https://msdn.microsoft.com/en-us/library/e6w8fe1b%28v=vs.71%29.aspx) keyword. A **const** field can only be initialized at the declaration of the field. A **readonly** field can be initialized either at the declaration or in a constructor.

Therefore, **readonly** fields can have different values depending on the constructor used. Also, while a **const** field is a compile-time constant,

the **readonly** field can be used for runtime constants as in the following

example:

public static readonly uint l1 = (uint) DateTime.Now.Ticks