# System.Random Class

```
[ILASM]
.class public serializable Random extends System.Object

[C#]
public class Random
```

**Assembly Info:**

- *Name:* mscorlib
- *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- *Version:* 1.0.x.x
- *Attributes:*
    - CLSCompliantAttribute(true)

## Summary

Generates psuedo-random numbers.

**Inherits From: System.Object**

**Library:** BCL

**Thread Safety:** All public static members of this type are safe for multithreaded operations. No instance members are guaranteed to be thread safe.

## Description

Instances of this class are initialized using a "seed", or starting value. The series of numbers generated by instances of the class are repeatable: given the same seed value, all instances of this class generate the same series of numbers.

[*Note:* The numbers generated by this class are chosen with equal probability from a finite set of numbers. The numbers are generated by a definite mathematical algorithm and are therefore not truly random, but are sufficiently random for practical purposes. For this reason, the numbers are considered to be psuedo-random.]

# Random() Constructor

[ILASM]
```
public rtspecialname specialname instance void .ctor()
```

[C#]
```
public Random()
```

## Summary

Constructs a new instance of the **Random** class using **System.Environment.TickCount** as the seed value.

## Description

This constructor is equivalent to **System.Random**(**System.Environment.TickCount**).

[*Note:* When generating random numbers on high performance systems, the system clock value may not produce the desired behavior. For details, see the **System.Random**(**System.Int32**) constructor.]

# Random(System.Int32) Constructor

[ILASM]

```
public rtspecialname specialname instance void .ctor(int32
Seed)
```

[C#]

```
public Random(int Seed)
```

## Summary

Constructs a new instance of the **Random** class using the specified seed value.

## Parameters

| Parameter | Description |
|---|---|
| *Seed* | A **System.Int32** used as the starting value for the pseudo-random number sequence. |

## Description

[*Note:* To construct instances that produce different random number sequences, invoke this constructor using different seed values such as may be produced by the system clock. Note, however that on high performance systems, the system clock may not change between invocations of the constructor, in which case the seed value will be the same for different instances of **Random**. When this is the case, additional operations are required to have the seed values differ in each invocation.]

## Example

The following example demonstrates using a bitwise complement operation to obtain different random numbers using a time-dependent seed value on high performance systems.

[C#]

```
using System;
class RandomTest {
    public static void Main() {
        Random rand1 = new Random();
        Random rand2 = new Random(Environment.TickCount);
        Console.WriteLine("The random number is
{0}",rand1.Next());
        Console.WriteLine("The random number is
{0}",rand2.Next());
```

```
        Random rdm1 = new
Random(unchecked(Environment.TickCount));
        Random rdm2 = new
Random(~unchecked(Environment.TickCount));
        Console.WriteLine("The random number is
{0}",rdm1.Next());
        Console.WriteLine("The random number is
{0}",rdm2.Next());
    }
}
```

The output is

```
The random number is 1990211954

The random number is 1990211954

The random number is 1990211954

The random number is 964628126
```

# Random.Next(System.Int32) Method

```
[ILASM]
.method public hidebysig virtual int32 Next(int32 maxValue)

[C#]
public virtual int Next(int maxValue)
```

## Summary

Returns a psuedo-random positive number less than the specified maximum.

## Parameters

| Parameter | Description |
|---|---|
| *maxValue* | The upper bound of the random number to be generated. *maxValue* is required to be greater than or equal to zero. |

## Return Value

A **System.Int32** set to a psuedo-random value greater than or equal to zero and less than *maxValue*. If *maxValue* is zero, returns zero.

## Behaviors

As described above.

## How and When to Override

Override this method to customize the algorithm used to generate the return value.

## Usage

Use this method to generate a psuedo-random number less than the specified maximum value.

## Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentOutOfRangeException** | *maxValue* is less than zero. |

# Random.Next(System.Int32, System.Int32) Method

[ILASM]
```
.method public hidebysig virtual int32 Next(int32 minValue,
int32 maxValue)
```

[C#]
```
public virtual int Next(int minValue, int maxValue)
```

## Summary

Returns a psuedo-random number within a specified range.

## Parameters

| Parameter | Description |
|---|---|
| *minValue* | The lower bound of the random number returned. |
| *maxValue* | The upper bound of the random number returned. |

## Return Value

A psuedo-random number greater than or equal to *minValue* and less than *maxValue*. If *minValue* and *maxValue* are equal, this value is returned.

## Behaviors

As described above.

## How and When to Override

Override this method to customize the algorithm used to generate the return value.

## Usage

Use this method to generate psuedo-random numbers in a specified range.

## Exceptions

| Exception | Condition |
|---|---|

| | |
|---|---|
| **System.ArgumentOutOfRangeException** | *minValue* is greater than *maxValue*. |

# Random.Next() Method

[ILASM]

```
.method public hidebysig virtual int32 Next()
```

[C#]

```
public virtual int Next()
```

## Summary

Returns a psuedo-random number between 0 and **System.Int32.MaxValue**.

## Return Value

A **System.Int32** greater than or equal to zero and less than **System.Int32.MaxValue**.

## Behaviors

As described above.

## How and When to Override

Override this method to customize the algorithm used to generate the return value.

## Example

The following example demonstrates using the **Next** method. The output generated by this example will vary.

[C#]

```
using System;
class RandomTest {
    public static void Main() {
        Random rand1 = new Random();
        for (int i = 0; i<10;i++)
        Console.WriteLine("The random number is
{0}",rand1.Next());

    }
}
```

The output is

```
The random number is 1544196111


The random number is 181749919


The random number is 1045210087


The random number is 1073826097


The random number is 1533078806


The random number is 1083151645


The random number is 569083504


The random number is 1711370568
The random number is 578178313


The random number is 409444742
```

# Random.NextBytes(System.Byte[]) Method

```
[ILASM]
.method public hidebysig virtual void NextBytes(class
System.Byte[] buffer)

[C#]
public virtual void NextBytes(byte[] buffer)
```

## Summary

Populates the elements of a specified array of bytes with random numbers.

## Parameters

| Parameter | Description |
|---|---|
| *buffer* | An array of bytes to be populated with random numbers. |

## Behaviors

Each element of the array of bytes is set to a random number greater than or equal to zero, and less than or equal to **System.Byte.MaxValue**.

## How and When to Override

Override this method to customize the algorithm used to generate the return value.

## Usage

Use the **NextByte** method to populate a **System.Byte** array with random numbers.

## Exceptions

| Exception | Condition |
|---|---|
| **System.ArgumentNullException** | *buffer* is a **null** reference. |

**The following member must be implemented if the ExtendedNumerics library is present in the implementation.**

# Random.NextDouble() Method

[ILASM]

```
.method public hidebysig virtual float64 NextDouble()
```

[C#]

```
public virtual double NextDouble()
```

## Summary

Returns a random number between 0.0 and 1.0.

## Return Value

A **System.Double** greater than or equal to 0.0, and less than 1.0.

## Behaviors

As described above.

## Usage

Use this method to generate a psuedo-random number greater than or equal to zero, and less than one.