

1 System.Random Class

```
2 [ILAsm]  
3 .class public serializable Random extends System.Object  
4 [C#]  
5 public class Random
```

6 Assembly Info:

- 7 • *Name:* mscorlib
- 8 • *Public Key:* [00 00 00 00 00 00 00 00 04 00 00 00 00 00 00 00]
- 9 • *Version:* 2.0.x.x
- 10 • *Attributes:*
 - 11 ○ CLSCompliantAttribute(true)

12 Summary

13 Generates psuedo-random numbers.

14 Inherits From: System.Object

15

16 **Library:** BCL

17

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

20

21 Description

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

25

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

30

31

32

1 Random() Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor()  
4 [C#]  
5 public Random()
```

6 Summary

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

9 Description

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

11
12 [*Note:* When generating random numbers on high performance systems, the system
13 clock value might not produce the desired behavior. For details, see the
14 `System.Random(System.Int32)` constructor.]
15
16

17

1 Random(System.Int32) Constructor

```
2 [ILAsm]  
3 public rtspecialname specialname instance void .ctor(int32 Seed)  
4 [C#]  
5 public Random(int Seed)
```

6 Summary

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

8 Parameters

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

9 10 Description

11 [Note: To construct instances that produce different random number sequences, invoke
12 this constructor using different seed values such as might be produced by the system
13 clock. Note, however that on high performance systems, the system clock might not
14 change between invocations of the constructor, in which case the seed value will be the
15 same for different instances of Random. When this is the case, additional operations are
16 required to have the seed values differ in each invocation.]
17
18

19 Example

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

```
23 [C#]  
24  
25 using System;  
26 class RandomTest {  
27     public static void Main() {  
28         Random rand1 = new Random();  
29         Random rand2 = new Random(Environment.TickCount);  
30         Console.WriteLine("The random number is {0}",rand1.Next());  
31         Console.WriteLine("The random number is {0}",rand2.Next());  
32  
33         Random rdml = new Random(unchecked(Environment.TickCount));  
34         Random rdm2 = new Random(~unchecked(Environment.TickCount));  
35         Console.WriteLine("The random number is {0}",rdml.Next());  
36         Console.WriteLine("The random number is {0}",rdm2.Next());
```

```
1     }
2   }
3
4   The output is
5
6
7   The random number is 1990211954
8
9
10  The random number is 1990211954
11
12
13  The random number is 1990211954
14
15
16  The random number is 964628126
17
18
```

1 Random.Next(System.Int32) Method

```
2 [ILAsm]  
3 .method public hidebysig virtual int32 Next(int32 maxValue)  
4 [C#]  
5 public virtual int Next(int maxValue)
```

6 Summary

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

8 Parameters

Parameter	Description
<i>maxValue</i>	The upper bound of the random number to be generated. <i>maxValue</i> is required to be greater than or equal to zero.

9 Return Value

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

13 Behaviors

14 As described above.

16 How and When to Override

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

19 Usage

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

23 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	<i>maxValue</i> is less than zero.

1

2

1 Random.Next(System.Int32, System.Int32)

2 Method

```
3 [ILAsm]  
4 .method public hidebysig virtual int32 Next(int32 minValue, int32  
5 maxValue)  
6 [C#]  
7 public virtual int Next(int minValue, int maxValue)
```

8 Summary

9 Returns a psuedo-random number within a specified range.

10 Parameters

Parameter	Description
<i>minValue</i>	The lower bound of the random number returned.
<i>maxValue</i>	The upper bound of the random number returned.

11

12 Return Value

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

15 Behaviors

16 As described above.

17

18 How and When to Override

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

20

21 Usage

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

23

24 Exceptions

Exception	Condition
System.ArgumentOutOfRangeException	<i>minValue</i> is greater than <i>maxValue</i> .

1

2

1 Random.Next() Method

```
2 [ILAsm]  
3 .method public hidebysig virtual int32 Next()  
4 [C#]  
5 public virtual int Next()
```

6 Summary

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

8 Return Value

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

10 Behaviors

11 As described above.

12

13 How and When to Override

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

15

16 Example

17 The following example demonstrates using the `Next` method. The output generated by
18 this example will vary.

```
19 [C#]  
20  
21 using System;  
22 class RandomTest {  
23     public static void Main() {  
24         Random rand1 = new Random();  
25         for (int i = 0; i<10;i++)  
26             Console.WriteLine("The random number is {0}",rand1.Next());  
27     }  
28 }  
29 }
```

30
31 The output is

32

33 The random number is 1544196111

34

35

1 The random number is 181749919
2
3
4 The random number is 1045210087
5
6
7 The random number is 1073826097
8
9
10 The random number is 1533078806
11
12
13 The random number is 1083151645
14
15
16 The random number is 569083504
17
18
19 The random number is 1711370568
20
21
22 The random number is 578178313
23
24
25 The random number is 409444742
26
27

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

```
2 [ILAsm]  
3 .method public hidebysig virtual void NextBytes(class System.Byte[]  
4 buffer)  
5 [C#]  
6 public virtual void NextBytes(byte[] buffer)
```

7 Summary

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

9 Parameters

Parameter	Description
<i>buffer</i>	An array of bytes to be populated with random numbers.

10

11 Behaviors

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

14

15 How and When to Override

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

17

18 Usage

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

20

21 Exceptions

Exception	Condition
<code>System.ArgumentNullException</code>	<i>buffer</i> is a null reference.

22

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

3 Random.NextDouble() Method

```
4 [ILAsm]  
5 .method public hidebysig virtual float64 NextDouble()  
  
6 [C#]  
7 public virtual double NextDouble()
```

8 Summary

9 Returns a random number between 0.0 and 1.0.

10 Return Value

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

12 Behaviors

13 As described above.

15 Usage

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