

Search: Go

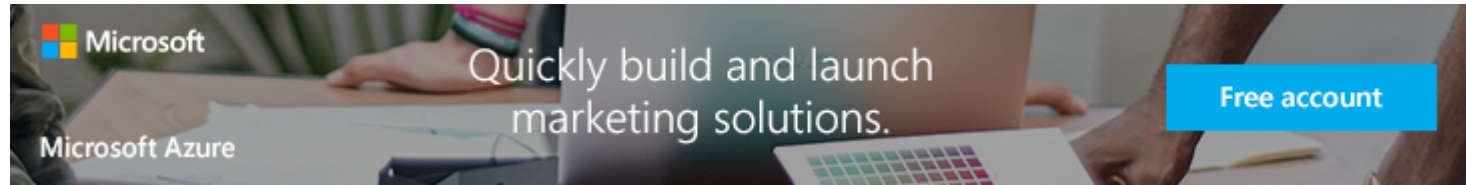
Not logged in

Reference <random>

[register](#)[log in](#)

C++
Information
Tutorials
Reference
Articles
Forum

Reference
C library:
Containers:
Input/Output:
Multi-threading:
Other:
<algorithm>
<bitset>
<chrono>
<codecvt>
<complex>
<exception>
<functional>
<initializer_list>
<iterator>
<limits>
<locale>
<memory>
<new>
<numeric>
<random>
<ratio>
<regex>
<stdexcept>
<string>
<system_error>
<tuple>
<typeindex>
<typeinfo>
<type_traits>
<utility>
<valarray>



header

<random>

Random

This header introduces random number generation facilities.

This library allows to produce random numbers using combinations of *generators* and *distributions*:

- **Generators:** Objects that generate uniformly distributed numbers.
- **Distributions:** Objects that transform sequences of numbers generated by a generator into sequences of numbers that follow a specific random variable distribution, such as [uniform](#), [Normal](#) or [Binomial](#).

Distribution objects generate random numbers by means of their `operator()` member, which takes a *generator object* as argument:

```
1 std::default_random_engine generator;
2 std::uniform_int_distribution<int> distribution(1,6);
3 int dice_roll = distribution(generator); // generates number in the range 1..6
```

For repeated uses, both can be bound together:

```
1 auto dice = std::bind ( distribution, generator );
2 int wisdom = dice()+dice()+dice();
```

Except for [random_device](#), all standard generators defined in the library are *random number engines*, which are a kind of generators that use a particular algorithm to generate series of pseudo-random numbers. These algorithms need a seed as a source of randomness, and this seed can either be a single value or an object with a very specific [generate\(\)](#) member function (see [seed_seq](#) for more info). A typical source of randomness for trivial tasks is time, such as the information provided by [time](#) or [system_clock::now](#) (for a typical example, see [uniform_int_distribution::operator\(\)](#)).

As an alternative, trivial random numbers can also be generated using `cstdlib`'s functions [rand](#) and [srand](#).

Generators

Pseudo-random number engines (templates)

<random>

distributions:

[bernoulli_distribution](#)
[binomial_distribution](#)
[cauchy_distribution](#)
[chi_squared_distribution](#)
[discrete_distribution](#)
[exponential_distribution](#)
[extreme_value_distribution](#)
[fisher_f_distribution](#)
[gamma_distribution](#)
[geometric_distribution](#)
[lognormal_distribution](#)
[negative_binomial_distribution](#)
[normal_distribution](#)
[piecewise_constant_distribution](#)
[piecewise_linear_distribution](#)
[poisson_distribution](#)
[student_t_distribution](#)
[uniform_int_distribution](#)
[uniform_real_distribution](#)
[weibull_distribution](#)

generators:

[default_random_engine](#)
[discard_block_engine](#)
[independent_bits_engine](#)
[knuth_b](#)
[linear_congruential_engine](#)
[mersenne_twister_engine](#)
[minstd_rand](#)
[minstd_rand0](#)
[mt19937](#)
[mt19937_64](#)
[random_device](#)
[ranlux24](#)
[ranlux24_base](#)
[ranlux48](#)
[ranlux48_base](#)
[shuffle_order_engine](#)
[subtract_with_carry_engine](#)

other:

[generate_canonical](#)
[seed_seq](#)

Generators that use an algorithm to generate pseudo-random numbers based on an initial seed:

linear_congruential_engine	Linear congruential random number engine (class template)
mersenne_twister_engine	Mersenne twister random number engine (class template)
subtract_with_carry_engine	Subtract-with-carry random number engine (class template)

Engine adaptors

They adapt an engine, modifying the way numbers are generated with it:

discard_block_engine	Discard-block random number engine adaptor (class template)
independent_bits_engine	Independent-bits random number engine adaptor (class template)
shuffle_order_engine	Shuffle-order random number engine adaptor (class template)

Pseudo-random number engines (instantiations)

Particular instantiations of generator engines and adaptors:

default_random_engine	Default random engine (class)
minstd_rand	Minimal Standard minstd_rand generator (class)
minstd_rand0	Minimal Standard minstd_rand0 generator (class)
mt19937	Mersenne Twister 19937 generator (class)
mt19937_64	Mersene Twister 19937 generator (64 bit) (class)
ranlux24_base	Ranlux 24 base generator (class)
ranlux48_base	Ranlux 48 base generator (class)
ranlux24	Ranlux 24 generator (class)
ranlux48	Ranlux 48 generator (class)
knuth_b	Knuth-B generator (class)

Random number generators

Non-deterministic random number generator:

random_device	True random number generator (class)
-------------------------------	--

Distributions**Uniform:**

uniform_int_distribution	Uniform discrete distribution (class template)
uniform_real_distribution	Uniform real distribution (class template)

Related to Bernoulli (yes/no) trials:

bernoulli_distribution	Bernoulli distribution (class)
binomial_distribution	Binomial distribution (class template)



geometric_distribution	Geometric distribution (class template)
negative_binomial_distribution	Negative binomial distribution (class template)

Rate-based distributions:

poisson_distribution	Poisson distribution (class template)
exponential_distribution	Exponential distribution (class template)
gamma_distribution	Gamma distribution (class template)
weibull_distribution	Weibull distribution (class template)
extreme_value_distribution	Extreme Value distribution (class template)

Related to Normal distribution:

normal_distribution	Normal distribution (class template)
lognormal_distribution	Lognormal distribution (class template)
chi_squared_distribution	Chi-squared distribution (class template)
cauchy_distribution	Cauchy distribution (class template)
fisher_f_distribution	Fisher F-distribution (class template)
student_t_distribution	Student T-Distribution (class template)

Piecewise distributions:

discrete_distribution	Discrete distribution (class template)
piecewise_constant_distribution	Piecewise constant distribution (class template)
piecewise_linear_distribution	Piecewise linear distribution (class template)

Other

seed_seq	Seed sequence (class)
generate_canonical	Generate canonical numbers (function template)