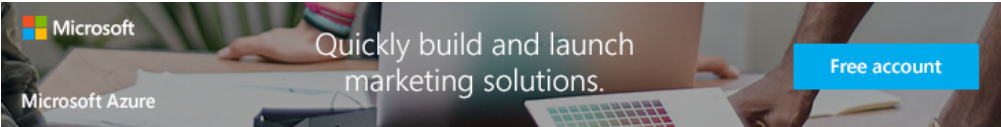


C++
Information
Tutorials
Reference
Articles
Forum

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>

<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

normal_distribution
normal_distribution::(constructor)
member functions:
normal_distribution::max
normal_distribution::mean
normal_distribution::min
normal_distribution::operator()



public member function

std::normal_distribution::operator()

<random>

```
(1) template<class URNG>
    result_type operator()(URNG& g);

(2) template<class URNG>
    result_type operator()(URNG& g, const param_type& parm);
```

Generate random number

Returns a new random number that follows the distribution's parameters associated to the object (version 1) or those specified by *parm* (version 2).

The generator object (*g*) supplies uniformly-distributed random integers through its `operator()` member function. The `normal_distribution` object transforms the values obtained this way so that successive calls to this member function with the same arguments produce floating-point values that follow a *Normal distribution* with the appropriate parameters.

Parameters
<div>g</div> <div>A uniform random number generator object, used as the source of randomness. URNG shall be a <i>uniform random number generator</i> type, such as one of the standard generator classes.</div> <div>parm</div> <div>An object representing the distribution's parameters, obtained by a call to member function <code>param</code>. param_type is a member type.</div>

Return value

A new random number.
result_type is a member type, defined as an alias of the first class template parameter (RealType).

Example

```
1 // normal_distribution example
2 #include <iostream>
3 #include <chrono>
4 #include <random>
5
6 int main()
7 {
8     // construct a trivial random generator engine from a time-based seed:
9     unsigned seed = std::chrono::system_clock::now().time_since_epoch().count();
10    std::default_random_engine generator (seed);
11
12    std::normal_distribution<double> distribution (0.0,1.0);
13
14    std::cout << "some Normal-distributed(0.0,1.0) results:" << std::endl;
15    for (int i=0; i<10; ++i)
16        std::cout << distribution(generator) << std::endl;
17
18    return 0;
19 }
```

Possible output:

```
some Normal-distributed(0.0,1.0) results:
1.01253
-1.6811
0.722295
-1.73855
0.0196423
-2.51224
1.37467
0.999222
1.24636
2.02573
```

Complexity

Amortized constant (a constant number of invocations of `g.operator()`).

See also

<code>normal_distribution::param</code>	Distribution parameters (public member function)
---	--

normal_distribution::param
normal_distribution::reset
normal_distribution::stddev

non-member functions:
operator<<
operator>>
relational operators

Ultra Tune Car Servicing

Contact us for your servicing, tyre, vehicle
maintenance and repair needs. Go to ultratune.com.au