std::srand

Defined in header <cstdlib>

```
void srand( unsigned seed );
```

Seeds the pseudo-random number generator used by std::rand() with the value seed.

If std::rand() is used before any calls to srand(), std::rand() behaves as if it was seeded with `srand(1)`.

Each time std::rand() is seeded with the same seed, it must produce the same sequence of values.

srand() is not guaranteed to be thread-safe.

Parameters

seed - the seed value

Return value

(none)

Notes

Generally speaking, the pseudo-random number generator should only be seeded once, before any calls to rand(), at the start of the program. It should not be repeatedly seeded, or reseeded every time you wish to generate a new batch of pseudo-random numbers.

Standard practice is to use the result of a call to `std::time(0)` as the seed. However, std::time returns a `std::time_t` value, and `std::time_t` is not guaranteed to be an integral type. In practice, though, every major implementation defines `std::time_t` to be an integral type, and this is also what POSIX requires.

Example

Run this code

```
#include <cstdlib>
#include <iostream>
#include <ctime>

int main()
{
    std::srand(std::time(0)); //use current time as seed for random generator
    int random_variable = std::rand();
    std::cout << "Random value on [0 " << RAND_MAX << "]: "
              << random_variable << '\n';
}
```

Possible output:

```
Random value on [0 2147483647]: 1373858591
```

See also

rand	generates a pseudo-random number (function)
RAND_MAX	maximum possible value generated by std::rand (macro constant)
reseed	reseeds the per-thread random engine (function)
C documentation for **srand**	

Retrieved from "https://en.cppreference.com/mwiki/index.php?title=cpp/numeric/random/srand&oldid=135848"