Search: Go

Not logged in

Reference <cstdlib> srand

register log in

C++

Information
Tutorials
Reference
Articles
Forum

Reference

C library:
<cassert> (assert.h)
<cctype> (ctype.h)
<cerrno> (errno.h)
<cfenv> (fenv.h)
<cfloat> (float.h)
<cinttypes> (inttypes.h)
<ciso646> (iso646.h)
<climits> (limits.h)
<clocale> (locale.h)
<cmath> (math.h)
<csetjmp> (setjmp.h)
<csignal> (signal.h)
<cstdarg> (stdarg.h)
<cstdbool> (stdbool.h)
<cstddef> (stddef.h)
<cstdint> (stdint.h)
<cstdio> (stdio.h)
<cstdlib> (stdlib.h)
<cstring> (string.h)
<ctgmath> (tgmath.h)
<ctime> (time.h)
<cuchar> (uchar.h)
<cwchar> (wchar.h)
<cwctype> (wctype.h)
Containers:
Input/Output:
Multi-threading:
Other:

<cstdlib> (stdlib.h)

functions:
abort
abs
atexit
atof
atoi
atol
atoll
at_quick_exit
bsearch
calloc
div
exit
free
getenv
labs
ldiv
llabs
lldiv
malloc
mblen
mbstowcs
mbtowc
qsort
quick_exit
rand
realloc
srand
strtod
strtof
strtol
strtold
strtoll
strtoul
strtoull
system
wcstombs
wctomb
_Exit
functions (non-standard):
itoa
types:
div_t
ldiv_t
lldiv_t
size_t

function

srand

<cstdlib>

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

Initialize random number generator

The pseudo-random number generator is initialized using the argument passed as *seed*.

For every different *seed* value used in a call to `srand`, the pseudo-random number generator can be expected to generate a different succession of results in the subsequent calls to `rand`.

Two different initializations with the same *seed* will generate the same succession of results in subsequent calls to `rand`.

If *seed* is set to `1`, the generator is reinitialized to its initial value and produces the same values as before any call to `rand` or `srand`.

In order to generate random-like numbers, `srand` is usually initialized to some distinctive runtime value, like the value returned by function `time` (declared in header <ctime>). This is distinctive enough for most trivial randomization needs.

Parameters

`seed`
An integer value to be used as seed by the pseudo-random number generator algorithm.

Return Value

none

Example

```
/* srand example */
#include <stdio.h>      /* printf, NULL */
#include <stdlib.h>     /* srand, rand */
#include <time.h>       /* time */

int main ()
{
  printf ("First number: %d\n", rand()%100);
  srand (time(NULL));
  printf ("Random number: %d\n", rand()%100);
  srand (1);
  printf ("Again the first number: %d\n", rand()%100);

  return 0;
}
```

Possible output:

```
First number: 41
Random number: 13
Again the first number: 41
```

Data races

The function accesses and modifies internal state objects, which may cause data races with concurrent calls to `rand` or `srand`.

Some libraries provide an alternative function of `rand` that explicitly avoids this kind of data race: `rand_r` (non-portable).

C++ library implementations are allowed to guarantee no *data races* for calling this function.

Exceptions (C++)

No-throw guarantee: this function never throws exceptions.

See also

rand	Generate random number (function)

macro constants:

- EXIT_FAILURE
- EXIT_SUCCESS
- MB_CUR_MAX
- NULL
- RAND_MAX