

we will discuss the functions/approaches to generate random numbers in C++ in detail.

**What You Will Learn:**[[hide](https://www.softwaretestinghelp.com/random-number-generator-cpp/)]

* [Pseudo-Random Number Generator (PRNG) In C++](https://www.softwaretestinghelp.com/random-number-generator-cpp/#Pseudo-Random_Number_Generator_PRNG_In_C)
* [rand And srand Functions In C++](https://www.softwaretestinghelp.com/random-number-generator-cpp/#rand_And_srand_Functions_In_C)
* [Difference Between rand () And srand ()](https://www.softwaretestinghelp.com/random-number-generator-cpp/#Difference_Between_rand_And_srand)
* [C++ Random Float](https://www.softwaretestinghelp.com/random-number-generator-cpp/#C_Random_Float)
* [C++ Random Number Between 0 And 1](https://www.softwaretestinghelp.com/random-number-generator-cpp/#C_Random_Number_Between_0_And_1)
* [C++ Random Number Between 1 And 10](https://www.softwaretestinghelp.com/random-number-generator-cpp/#C_Random_Number_Between_1_And_10)
* [Frequently Asked Questions](https://www.softwaretestinghelp.com/random-number-generator-cpp/#Frequently_Asked_Questions)
* [Conclusion](https://www.softwaretestinghelp.com/random-number-generator-cpp/#Conclusion)
* [Recommended Reading](https://www.softwaretestinghelp.com/random-number-generator-cpp/#Recommended_Reading)

Pseudo-Random Number Generator (PRNG) In C++

In general, a pseudo-random number generator (PRNG) can be defined as a program that takes a seed or a starting number and transforms it into some other number that is different from seed using mathematical operations.

This process is carried out repeatedly by taking the last generated number every time. Each time the number generated is unrelated to the previous numbers. Thus this program is able to generate a series of numbers that appear random.

C++ language comes with an in-built pseudo-random number generator and provides two function rand () and srand () that can be used to generate random numbers.

Let’s discuss these two functions in detail.

rand And srand Functions In C++

**srand ()**

**Function prototype:** void srand (unsigned int seed);  
**Parameters:** seed – An integer value to be used as seed by the pseudo-random number generator algorithm.  
**Return value:** None  
**Description:** srand function is used to initialize the sequence of pseudo-random numbers with a parameter called ‘random seed’.

It makes the output of the rand function look random. Otherwise, the output of the rand () function will be the same every time we call it.

Thus, if we seed the random number generator with the srand () function, it will start the generator from the point that is dependent on the argument value passed to srand. If we set the random number generator with the system time **For example,** before the first call to the rand () function, then it will generate the random numbers every time we run the program.

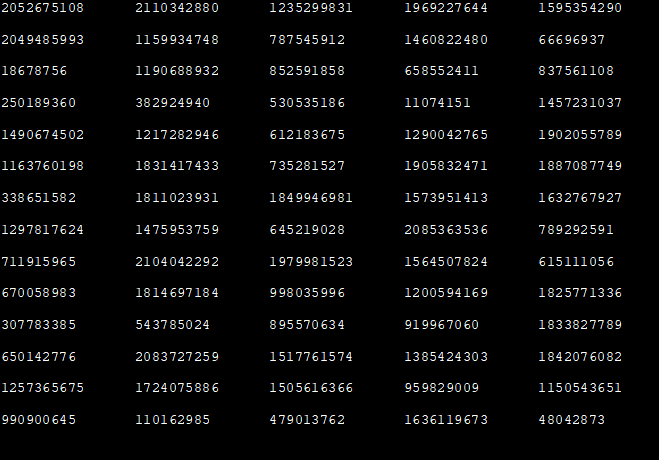
Note that we need to usually call the srand () function only once before the call to rand () function and not every time we generate random numbers.

**rand ()**

**Function prototype:** int rand (void);  
**Parameters:**none  
**Return value:** An integer value between 0 and RAND\_MAX.  
**Description:** The rand () function generates the next random number in the sequence. The number generated is the pseudo-random integer between 0 and RAND\_MAX. RAND\_MAX is a constant in the header <cstdlib> generally set to value 32767.

|  |
| --- |
| #include <iostream>  #include <cstdlib>  #include <ctime>    int main()  {      std::srand(static\_cast<unsigned int>(std::time(nullptr)));        for (int count=1; count <= 100; ++count)      {          std::cout << std::rand() << "\t";            // display 5 random numbers per row          if (count % 5 == 0)              std::cout << "\n";       }    return 0;  } |

**Output:**

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/09/1-7.png)

In the above program, we have generated the first 100 random numbers, by giving system clock as the seed for srand function. In this program, we have used both srand as well as rand functions. Note that because of the system clock as seed, the output generated will be different every time when we execute the program.

Difference Between rand () And srand ()

| **rand()** | **srand()** |
| --- | --- |
| Used to generate random numbers. | Seeds the PRNG used by rand () function. |
| Called as many times as we want to generate the random numbers. | Called only once to see the random number generator. |
| Doesn’t take any arguments. | Takes the parameter that is used to seed the random number generator. |
| Returns the sequence of random numbers each time it is called. | Does not return a value. |

C++ Random Float

The rand () function that we have seen above by default returns an integer value which can cause an overflow in certain cases. Thus, we can use float or double value. We can generate float random numbers by casting the return value of the rand () function to ‘float’.

**Thus the following will generate a random number between float 0.0 and 1.0 (both inclusive).**

cout<<static\_cast <float> (rand()) / static\_cast <float> (RAND\_MAX);

**Similarly, the below line will generate a random number between 1.2 and 3.4**

cout<<1.2 + static\_cast <float> (rand()) / ( static\_cast <float> (RAND\_MAX/(3.4-1.2)));

In our subsequent example below we make use of random float to generate the output.

C++ Random Number Between 0 And 1

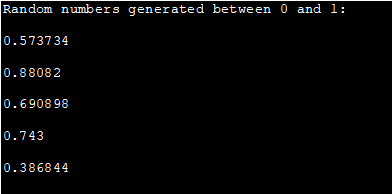
We can use srand () and rand () function to generate random numbers between 0 and 1. Note that we have to cast the output of rand () function to the decimal value either float or double.

The default return value of rand () function i.e. integer is inadequate to display random numbers between 0 and 1 which are fractions.

**C++ program given below displays the first five random numbers between 0 and 1.**

|  |
| --- |
| #include <time.h>  #include <iostream>  using namespace std;  int main()  {      cout<<"Random numbers generated between 0 and 1:"<<endl;      srand( (unsigned)time( NULL ) );        for (int i = 0; i < 5; i++)      {          cout << (float) rand()/RAND\_MAX << endl;      }      return 0;  } |

**Output**:

[](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/09/3-6.png)

We see that the output of the program is the random number between 0 and 1 which are fractions.

If we don’t cast the return value of rand () function to float or double, then we will get 0 as the random number.

C++ Random Number Between 1 And 10

The next example is to generate random numbers between 1 and 10. Following is the C++ program that generates random numbers.

We call the srand function with the system clock and then call the rand function with module 10 operators.

|  |
| --- |
| #include <iostream>  #include <ctime>  #include <cstdlib>  using namespace std;  int main()  {      srand(time(0));  // Initialize random number generator.        cout<<"Random numbers generated between 1 and 10:"<<endl;      for(int i=0;i<10;i++)          cout << (rand() % 10) + 1<<" ";      return 0;  } |

**Output:**

[Random numbers generated between 1 and 10](https://www.softwaretestinghelp.com/wp-content/qa/uploads/2019/09/4-5.png)

In the above program, we generate the first 10 random numbers between 1 and 10. Note that every time the program is run, it will generate different sets of numbers because of the srand function being called.

Frequently Asked Questions

**Q #1) What is the header file for Random function in C++?**

**Answer:** The functions to generate random numbers, rand and srand are defined in <**cstdlib**> header of C++.

**Q #2) What is Rand\_max in C++?**

**Answer: RAND\_MAX** is a constant in header <cstdlib> generally set to value 32767. The pseudo-random number generator (PRNG) generates random numbers between 0 to RAND\_MAX.

**Q #3) How does the random function work?**

**Answer:**C++ supports two random functions i.e. **srand () and rand ( ).** The function srand () seeds the random number generator used by rand () function which generates the random number sequence depending on the initial seed provided.

**Q #4) How do you srand with time?**

**Answer:**The srand function seeds the pseudo-random number generator (PRNG) used by the rand () function. It is a standard practice to use the result of a call to time (0) as seed. This time function returns the value, a number of seconds since 00:00 hours, Jan 1, 1970, UTC (current UNIX timestamp).

Thus the value of seed changes every second. Hence every time when srand is called with time function, a new set of the random numbers is generated.