

Forum:[Hack Forums](#)

Thread: [Random number to an int \(c++\)](#)

Maskin - April 11, 2021, 8:53 pm

I want to give my "int bValue" a random number from 1-50, and have that number stored so that the next time I print "bValue" I get that same number which was generated before.

I tried doing this on my own and using google but no success. I haven't been coding for very long as you can tell.

Grateful for any help.

Lynx - April 12, 2021, 1:09 am

Can you show us the code you currently have?

Sajuuk - April 12, 2021, 10:46 am

Code // C style pseudo-random

```
#include <cstdlib>
```

```
#include <ctime>
```

```
// seed current time
```

```
    srand(time(0));
```

```
// generate number
```

```
    int bValue = rand() % 51;
```

```
// print number
```

```
    printf("Pseudo-random number: ", bValue);
```

Your phrasing is weird to me. Sounds like you just want to be able to print it later on. If that's the case, store it in an int (or "unsigned int" to eliminate the possibility of somehow getting a negative) and print the int whenever you need. IIRC the return value of rand() is an int, but otherwise just typecast it.

MHC-1 - April 12, 2021, 12:30 pm

Another solution. Usually I use a Mersenne-Twister. Another way of generating a random number.

Code #include <iostream>

```
#include <random>
```

```
using namespace std;
```

```
int main()
```

```
{
```

```
    // C++ - Generate rndm. num
```

```
    random_device seed_generator;
```

```
    mt19937_64 random_device(seed_generator());
```

```
    uniform_int_distribution<int> distribution(1, 50);
```

```
    // stored in bValue1
```

```
    const int bValue1 = distribution(random_device);
```

```

// print first
cout << bValue1 << endl;

// do sth.

// print sec.
cout << bValue1 << endl;
}

```

I think this follows the steps you already mentioned in your question.

Maskin - April 13, 2021, 3:26 pm

(Apr 12, 2021 10:46 AM)Sajuuk Wrote: [\[CITE\]](#) Code // C style pseudo-random

```

#include <cstdlib>
#include <ctime>

```

```

// seed current time
srand(time(0));
// generate number
int bValue = rand() % 51;
// print number
printf("Pseudo-random number: ", bValue);

```

Your phrasing is weird to me. Sounds like you just want to be able to print it later on. If that's the case, store it in an int (or "unsigned int" to eliminate the possibility of somehow getting a negative) and print the int whenever you need. IIRC the return value of rand() is an int, but otherwise just typecast it.

Sorry about that, but thanks now it works.

The issue I had was that the code below would always run the 'else' regardless if 'bValue == gValue'. So I assumed the random generator kept giving 'int bValue' a new number or something. I'm new at this just trying to learn.

```

Code if(bValue == gValue) {
cout << dock << endl;
} else {
cout << "The number was incorrect." << endl;
}

```

Vector Dynamics - April 21, 2021, 1:52 am

(Apr 12, 2021 12:30 PM)MHC-1 Wrote: [\[CITE\]](#) Another solution. Usually I use a Mersenne-Twister. Another way of generating a random number.

```

Code #include <iostream>
#include <random>

```

```

using namespace std;

```

```

int main()
{
// C++ - Generate rndm. num
random_device seed_generator;
mt19937_64 random_device(seed_generator());
uniform_int_distribution<int> distribution(1, 50);

```

```
// stored in bValue1
const int bValue1 = distribution(random_device);

// print first
cout << bValue1 << endl;

// do sth.

// print sec.
cout << bValue1 << endl;
}
```

I think this follows the steps you already mentioned in your question.

This is the way you're supposed to do it in C++, the other suggested way in earlier post is for C.