

Forum:[Hack Forums](#)

Thread: [c++ random number generator](#)

pcanningg - November 14, 2016, 5:59 am

Write a program that will randomly generate a number between 1 and 100. The program should allow the user to guess the number, narrowing down the range of values based on users' input, eg:

Number generated: 43

Computer output: Please enter a number between 1 and 100

Guess entered: 73

Computer output: Your number lies between 0 and 73

The game should allow 6 attempts

My code so far

```
#include "stdafx.h"
#include <ctime>
#include <conio.h>
#include < Windows.h>
#include <iomanip>
#include <iostream>
#include <string>
```

```
using namespace std;
```

```
int main()
```

```
{
    int randnum, num = 0;
```

```
    randnum = (rand() % 100 + 1);
```

```
    for (int i = 0; i <= 5; i++)
```

```
    {
        cout << "\n\n\tPlease enter a number between 0 and 100: ";
        cin >> num;
```

```
        if (randnum == num)
        {
            cout << "You have guessed the random number!!!";
        }
```

```
    else
```

```
    {
        if (randnum != num)                <Some help with this part
        {
```

```
}  
}
```

```
}
```

Dostoevsky - November 14, 2016, 7:58 am

You'd be better off narrowing the choices by checking if the option is greater, or lesser than, the random number. Once you've guessed the random number, you just need to break your for.

While we're at it, you should maybe take advantage of C++11's `<random>` library but I suppose that may be overkill for this simple project.

Your conditions are also weird. If the number isn't equal, it implies it's not equal so your second condition doesn't have to be so explicit. As I said, you should check against their scale using the greater than and lesser than conditional operators to provide helpful output to narrow down the user's suggestions.

EDIT: Why are you including the Windows header? There's nothing in here that requires WinAPI. Don't do what I think you're gonna do and use `system("pause");` or some `getch();` bullshit. Use `std::ignore`. Also, you should probably output `std::endl` rather than `"\n"`.

pcanningg - November 14, 2016, 8:14 am

(Nov 14, 2016 08:58 AM)Dostoevsky Wrote: [\[CITE\]](#) You'd be better off narrowing the choices by checking if the option is greater, or lesser than, the random number. Once you've guessed the random number, you just need to break your for.

While we're at it, you should maybe take advantage of C++11's `<random>` library but I suppose that may be overkill for this simple project.

Your conditions are also weird. If the number isn't equal, it implies it's not equal so your second condition doesn't have to be so explicit. As I said, you should check against their scale using the greater than and lesser than conditional operators to provide helpful output to narrow down the user's suggestions.

EDIT: Why are you including the Windows header? There's nothing in here that requires WinAPI. Don't do what I think you're gonna do and use `system("pause");` or some `getch();` bullshit. Use `std::ignore`. Also, you should probably output `std::endl` rather than `"\n"`.

I cant use the greater or smaller thing that you were talking about as the computer must output what the random is between ie

random number = 42

user guesses 10

computer: the number is between 10 and 100

Also my code is going to look like shit because Ive been programming for like 4 weeks and Im only going off of what my teacher has taught me so i dont know a lot.

Dostoevsky - November 14, 2016, 8:22 am

(Nov 14, 2016 09:14 AM)pcanningg Wrote: [\[CITE\]](#) I cant use the greater or smaller thing that you were talking about as the computer must output what the random is between ie

random number = 42

user guesses 10

computer: the number is between 10 and 100

Also my code is going to look like shit because Ive been programming for like 4 weeks and Im only going off of what my teacher has taught me so i dont know a lot.

You.. you.. have to use the greater and lesser than operators to do what you want to do.. You can't tell if the random number is between 2 points unless you tell that it's: $\text{min} < x < \text{max}$.

You're not showing off using shit you should already know (and that you have used in your for loop).

pcanningg - November 14, 2016, 8:44 am

(Nov 14, 2016 09:22 AM)Dostoevsky Wrote: [\[CITE\]](#) You.. you.. have to use the greater and lesser than operators to do what you want to do.. You can't tell if the random number is between 2 points unless you tell that it's: $\text{min} < x < \text{max}$.

You're not showing off using shit you should already know (and that you have used in your for loop).

Bro what are you talking about showing off of shit, your saying that I have to use the greater and lesser than operators for the program to output where the number is?

Dostoevsky - November 14, 2016, 8:56 am

(Nov 14, 2016 09:44 AM)pcanningg Wrote: [\[CITE\]](#) Bro what are you talking about showing off of shit, your saying that I have to use the greater and lesser than operators for the program to output where the number is?

Sorry, I'm exhausted and read "going off" as showing off. As more evidence of my exhaustion, I just sliced half of the tip of my finger off with a bread knife by accident.

I can't type that easily rn but yeah u need the operators: As said: $\text{min} < x < \text{max}$

pcanningg - November 14, 2016, 9:10 am

(Nov 14, 2016 09:56 AM)Dostoevsky Wrote: [\[CITE\]](#) Sorry, I'm exhausted and read "going off" as showing off. As more evidence of my exhaustion, I just sliced half of the tip of my finger off with a bread knife by accident.

I can't type that easily rn but yeah u need the operators: As said: $\text{min} < x < \text{max}$
lol worries man, and thank you for the help

Infinity. - November 14, 2016, 9:39 am

(Nov 14, 2016 09:56 AM)Dostoevsky Wrote: [\[CITE\]](#) Sorry, I'm exhausted and read "going off" as showing off. As more evidence of my exhaustion, I just sliced half of the tip of my finger off with a bread knife by accident.

I can't type that easily rn but yeah u need the operators: As said: $\text{min} < x < \text{max}$
RIP half of the tip of your finger. ?

RDCA - November 14, 2016, 9:42 pm

(Nov 14, 2016 08:58 AM)Dostoevsky Wrote: [\[CITE\]](#) Use `std::ignore`. Also, you should probably output `std::endl` rather than `"\n"`.

No... do not use `std::endl` rather than `"\n"` unless you're explicitly trying to flush the buffer. It's extremely expensive relative to a newline.

Sajuuk - November 14, 2016, 9:49 pm

(Nov 14, 2016 10:42 PM)RDCA Wrote: [\[CITE\]](#) No... do not use `std::endl` rather than `"\n"` unless you're explicitly trying to flush the buffer. It's extremely expensive relative to a newline.

I was about to type up half of a college essay's worth of paragraphs on why not to use `std::endl` for every newline, but I just remembered that in the class I'm taking, the professor hasn't even told us about escape characters yet, so everyone except me is using `\n` for newlines. Thus, the couple hundred words I would have typed would have zero first-hand support as far as education goes.

From personal experience, do what RDCA said: don't use `std::endl` for every newline character you need to print.

Dostoevsky - November 14, 2016, 10:25 pm

(Nov 14, 2016 10:42 PM)RDCA Wrote: [\[CITE\]](#) No... do not use `std::endl` rather than `"\n"` unless you're explicitly trying to flush the buffer. It's extremely expensive relative to a newline.

Thanks for clarifying that. My mistake, I was labouring under the delusion that `"\n"` wasn't cross platform (my confusion, of course, deriving from the new lines in actual files and how they differ per platform - whoops).

Thanks again.

Sajuuk - November 14, 2016, 10:35 pm

(Nov 14, 2016 11:25 PM)Dostoevsky Wrote: [\[CITE\]](#) Thanks for clarifying that. My mistake, I was labouring under the delusion that `"\n"` wasn't cross platform (my confusion, of course, deriving from the new lines in actual files and how they differ per platform - whoops).

Thanks again.

Going back up and rereading your first posts on this thread, I now understand what you meant by the late night altering perception:

(Nov 14, 2016 10:49 PM)Sajuuk Wrote: [\[CITE\]](#) the professor hasn't even told us about escape characters yet, so everyone except me is using `\n` `std::endl` for newlines.

I expect things to only get worse from here on out.

0xDEAD10CC - November 15, 2016, 8:54 pm

(Nov 14, 2016 10:42 PM)RDCA Wrote: [\[CITE\]](#) No... do not use `std::endl` rather than `"\n"` unless you're explicitly trying to flush the buffer. It's extremely expensive relative to a newline.

Most implementations are line-buffered anyways, so the same thing ends up happening essentially, where `endl` is probably just a bit more overhead because of a second write and an added function call. The difference comes in when the buffering changes, or when things like `std::cout.sync_with_stdio(false)` are used, but `stdout` is fully buffered by default. "Extremely" is also subjective -- how many times does one write to the console anyways in a program like above? Surely it's an small "performance hit" in this case.

RDCA - November 16, 2016, 10:58 am

(Nov 15, 2016 09:54 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Most implementations are line-buffered anyways, so the same thing ends up happening essentially, where `endl` is probably just a bit more overhead because of a second write and an added function call. The difference comes in when the buffering changes, or when things like `std::cout.sync_with_stdio(false)` are used, but `stdout` is fully buffered by default. "Extremely" is also subjective -- how many times does one write to the console anyways in a program like above? Surely it's an small "performance hit" in this case.

Measured performance on Linux systems tends to be about 9x more expensive, and it's even worse on Windows. Hence it's relatively expensive to a newline. Also, clearly I'm not talking about the specific code the OP has, I'm talking about a rule of thumb.

0xDEAD10CC - November 16, 2016, 11:17 am

(Nov 16, 2016 11:58 AM)RDCA Wrote: [\[CITE\]](#) Measured performance on Linux systems tends to be

about 9x more expensive, and it's even worse on Windows. Hence it's relatively expensive to a newline. Also, clearly I'm not talking about the specific code the OP has, I'm talking about a rule of thumb.

Relatively, but say something takes 1 nanosecond, 9x is still only just 9 nanoseconds. If "\n" took 1 second, then I'm sure more people would be concerned, but unless you're looping it, nobody will ever see or care to measure the difference. As a rule of thumb, sure though.