

Forum:[Hack Forums](#)

Thread: [Simple program.. Am I developing any bad habits?](#)

asillychilly - March 2, 2014, 4:14 pm

Code #include <iostream>

#include <time.h>

#include <stdlib.h>

using namespace std;

int generateFunction()

{

int intGen;

srand(time(NULL));

intGen = rand()%10;

return intGen;

}

int main()

{

char decision;

//int number;

int guess;

int counter = 0;

cout << "Generating a random number, please wait.." << endl;

cout << endl;

int number = generateFunction();

cout << endl;

cout << "The number is between 1-10, you have 3 chances.. type in your guess please. \n";

for(int i=0; i<3; i++)

{

cin >> guess;

counter++;

if(guess != number) {

cout << "Sorry that's wrong. \n";

}

else {

cout << "You win! \n";

return 0;

}

}

```
return 0;
```

```
}
```

I'm a beginner, I just started learning C++11 a few days ago, I do have backgrounds in Java and VB.NET. Do you see any bad habits being developed? I'm reading along with Stanford C++ handouts, as well as reading along with the C++ Primer (5th Edition) so I'm getting influenced by a lot of different instructors.

Sajuuk - March 2, 2014, 4:28 pm

Yes:

1. You can seed random with `srand(rand())`. It's a hell of a lot quicker to do, and it works.
2. You're utilizing a namespace specifier. It's best to use something like `std::wcout` or the like because as you advance, you'll most likely have to use multiple namespaces, which could or could not use functions with the same name in different namespaces. Boost is an example:

```
Code std::wcout << L"Hello there.";
boost::boostFunctionHere();
/* etc */
```

asillychilly - March 2, 2014, 4:30 pm

(Mar 2, 2014 05:28 PM)Sajuuk Wrote: [\[CITE\]](#) Yes:

1. You can seed random with `srand(rand())`. It's a hell of a lot quicker to do, and it works.
2. You're utilizing a namespace specifier. It's best to use something like `std::wcout` or the like because as you advance, you'll most likely have to use multiple namespaces, which could or could not use functions with the same name in different namespaces. Boost is an example:

```
Code std::wcout << L"Hello there.";
boost::boostFunctionHere();
/* etc */
```

isn't the `rand()` function only pseudo-random? Wouldn't that mean the seed is going to develop into a pattern, rendering the `rand()` after the `srand()` predictable? As for #2 thanks, I'll look into that.

Sajuuk - March 2, 2014, 4:34 pm

(Mar 2, 2014 05:30 PM)asillychilly Wrote: [\[CITE\]](#) isn't the `rand()` function only pseudo-random? Wouldn't that mean the seed is going to develop into a pattern, rendering the `rand()` after the `srand()` predictable? As for #2 thanks, I'll look into that.

I use `srand(rand())` only when I'm fucking around with C/C++. I suppose for more sophisticated application code you'd have to use `srand(time(0))` because `rand()` is indeed pseudo-random.

g4143 - March 2, 2014, 4:42 pm

If you find that your if branches return the same type then you may consider using the ternary operator.

```
Code #include <iostream>
```

```
int main()
{
    int x = 123;
    int y = 0;
```

```
std::cout << "Enter a number->";
std::cin >> y;

std::cout << ((y != x) ? "Not equal" : "Equal") << std::endl;
}
```

Skyla Blue - March 2, 2014, 5:00 pm

(Mar 2, 2014 05:14 PM)asillychilly Wrote: [\[CITE\]](#) Code int generateFunction()

```
{
    int intGen;
    srand(time(NULL));
    intGen = rand()%10;
    return intGen;
}
```

You could just do return rand()%10, instead of storing it in a variable.

nramsey - March 2, 2014, 5:07 pm

Quote:Do you see any bad habits being developed?

The explicit qualification of entities and the disuse of C Standard Library functions for pseudo-random number generation should be preferred.

<http://en.cppreference.com/w/cpp/numeric/random>
<http://channel9.msdn.com/Events/GoingNative/2013/rand-Considered-Harmful>

0xDEAD10CC - March 3, 2014, 6:37 am

(Mar 2, 2014 05:28 PM)Sajuuk Wrote: [\[CITE\]](#) Yes:

1. You can seed random with srand(rand()). It's a hell of a lot quicker to do, and it works.
2. You're utilizing a namespace specifier. It's best to use something like std::wcout or the like because as you advance, you'll most likely have to use multiple namespaces, which could or could not use functions with the same name in different namespaces. Boost is an example:

```
Code std::wcout << L"Hello there.";
boost::boostFunctionHere();
/* etc */
```

1. That's a stupid idea... If srand() was not called previously then the prng is not seeded, and will generate all of the values default to the algorithm it uses, which destroys entropy. If this is the first call, you might as well just put:

```
Code srand(41);
```

Seeding with the time is common, not the best, but much better than seeding immediately with rand(). If it was never previously seeded, one could guess the entire array of numbers that successive calls to rand() will generate.

Code 172

```
22420
5
5992
5768
8953
4904
10571
32065
```

7917 ...

If you needed something better than time, seed with the current mouse position.

The other thing: If this is C++, then it's common to prefix with C instead of suffix with ".h" for header files...

Code `#include <time.h>`

Becomes:

Code `#include <ctime>`

Mouse? - March 3, 2014, 8:57 am

Touching up on the last two posts, they're basically telling you to stop mixing C and C++ so much. C++11 gives you `<random>` which allows you to get truly random numbers with `std::random_device` without having to explicitly give it a seed.

http://www.cplusplus.com/reference/random/random_device/

You're doing this.

Code `int intGen;`

`// random num seed`

`intGen = rand()%10;`

`return intGen;`

Now look at this extreme example of what you did.

Code `int a=5,b,c,d,e,f,g;`

`b=a;`

`c=b;`

`d=c;`

`e=d;`

`f=e;`

`g=f;`

`return g;`

when you could just

Code `return 5;`

Also post-increment operator (`variable++`;). When you're not doing anything with the predecessor value, this just wastes the cpu cycles to copy the variable, return it and then increment it when by using pre-increment operator, (`++variable`;) it simply increments the value and then returns it. No copying happens.

Your code:

Code `for(int i=0; i<3; i++)`

`{`

`cin >> guess;`

`counter++;`

`// ...Example of what I'm saying:`

Code `for(int i=0; i<3; ++i)`

`{`

`cin >> guess;`

`++counter;`

`//...`

asillychilly - March 3, 2014, 11:33 am

(Mar 3, 2014 09:57 AM)Rodaxoleaux Wrote: [\[CITE\]](#) Touching up on the last two posts, they're basically telling you to stop mixing C and C++ so much. C++11 gives you `<random>` which allows

you to get truly random numbers with `std::random_device` without having to explicitly give it a seed.
http://www.cplusplus.com/reference/random/random_device/

You're doing this.

```
Code int intGen;  
// random num seed  
intGen = rand()%10;  
return intGen;
```

Now look at this extreme example of what you did.

```
Code int a=5,b,c,d,e,f,g;  
b=a;  
c=b;  
d=c;  
e=d;  
f=e;  
g=f;  
return g;  
when you could just
```

```
Code return 5;
```

Also post-increment operator (`variable++`). When you're not doing anything with the predecessor value, this just wastes the cpu cycles to copy the variable, return it and then increment it when by using pre-increment operator, (`++variable`;) it simply increments the value and then returns it. No copying happens.

Your code:

```
Code for(int i=0; i<3; i++)  
{  
    cin >> guess;  
    counter++;  
// ...Example of what I'm saying:  
Code for(int i=0; i<3; ++i)  
{  
    cin >> guess;  
    ++counter;  
//...
```

Thanks for the super helpful post. As for mixing C/C++, well like I said I'm using various mediums to learn and some of them emphasize C basics first and others emphasize C++11 first so I'm getting mixed up from different influences. Thanks!

Sajuuk - March 3, 2014, 12:56 pm

(Mar 3, 2014 07:37 AM)TxTriton Wrote: [\[CITE\]](#) 1. That's a stupid idea... If `srand()` was not called previously then the prng is not seeded, and will generate all of the values default to the algorithm it uses, which destroys entropy. If this is the first call, you might as well just put:

```
Code srand(41);
```

Seeding with the time is common, not the best, but much better than seeding immediately with `rand()`. If it was never previously seeded, one could guess the entire array of numbers that successive calls to `rand()` will generate.

```
Code 172
```

```
22420
```

```
5
```

```
5992
```

5768
8953
4904
10571
32065
7917 ...

If you needed something better than time, seed with the current mouse position.

The other thing: If this is C++, then it's common to prefix with C instead of suffix with ".h" for header files...

Code `#include <time.h>`

Becomes:

Code `#include <ctime>`

I stated that I seed with `rand()` "when I'm fucking with shit", meaning when I'm not doing real work, just fiddling around with functions. I guess I failed to tell him to only use `rand()` to seed when doing stuff just for fun. Sorry.

0xDEAD10CC - March 3, 2014, 2:00 pm

(Mar 3, 2014 01:56 PM)Sajuuk Wrote: [\[CITE\]](#) I stated that I seed with `rand()` "when I'm fucking with shit", meaning when I'm not doing real work, just fiddling around with functions. I guess I failed to tell him to only use `rand()` to seed when doing stuff just for fun. Sorry.

Quote:I stated that I seed with `rand()` "when I'm fucking with shit"

That's a great excuse, but you're not fooling people here.

Here's what you initially wrote:

Quote:1. You can seed random with `srand(rand())`. It's a hell of a lot quicker to do, and it works.

"quicker", and "it works"... *On a sidenote, you know what else is even quicker and works just as well?

Code `srand(41);`

If this was used when you're just fooling around, I don't see why you'd be suggesting it... Especially since he's already written a better method of seeding the prng--in addition to the fact that it's not much more difficult to write one over the other (a difference of 4 characters?). The fact that you'd even back it up with "when I'm fucking with shit" is silly, but it hints to me that you don't understand `rand()` and `srand()`.

Quote:I guess I failed to tell him to only use `rand()` to seed when doing stuff just for fun. Sorry.

This is almost like saying "I forgot to tell him that he should only smoke if he wants to smoke, but I'll tell him that he should be smoking first, even though being a non-smoker, as his original decision, was the better one."

BTW - Here was his original question, if you didn't see it:

Quote:Do you see any bad habits being developed?

I don't even know why you'd post such a ridiculous suggestion, then try to back it up with illogical reasons. What he initially had was perfectly fine for a beginner, and wasn't really classified as a bad habit at all. Your suggestion is ironic because it would've made his code even worse.

Sajuuk - March 3, 2014, 2:06 pm

(Mar 3, 2014 03:00 PM)TxTriton Wrote: [\[CITE\]](#) That's a great excuse, but you're not fooling people

here.

That's great, but for what reason was I supposed to fool you all? Whenever I first started programming C++, I used to seed it with `rand()`, but no longer need to use it because I haven't had any way to utilize it in the last eight months (I won't explain, it's a hell of a long explanation). Truly... my idiocy.

DavidProHQ - March 3, 2014, 6:38 pm

That's pretty good and that's how I generate random numbers. There are other ways, but for learning purposes, that's just fine.

Like someone said, you're listing your header files in C format instead of C++. And I don't see anything wrong with 'using namespace std'. Some older programs might not use that so it's good to be aware that's not always used.

Also, that will generate a number 0-9 not 1-10. No biggie, but just something to be aware of.

Lastly, awesome of you for making a post like this. Too many people make post like "Where I learn C++?!?!" Or they clearly post some copy and pasted material none of which they know nothing about. You learn by doing, just like what you're doing. And then ask for guidance to make sure you're on the right track, just like you're doing.