

Re: passing const ptr argument

Source: <http://coding.derkeiler.com/Archive/C/ CPP/comp.lang.cpp/2004-08/1570.html>

From: Gianni Mariani (*gi2nospam_at_mariani.ws*)

Date: 08/11/04

Date: 11 Aug 2004 04:33:41 EDT

Anthony wrote:

> *Hi,*
> *I want to write protect an argument that is passed by reference.*
> *But I still want to be able to call members of that argument.*
> *E.g. code (bottum) will gives the following compiler error:*
> *writeProtEvt.cpp: In member function `virtual void A::print(const B*):*
> *writeProtEvt.cpp:24: error: passing `const B' as `this' argument of `virtual*
> *void B::print()' discards qualifiers*
>
> *Can anyone explain how to do this the right way?*
>
> *thanks in advance,*
> *Anthony Lansbergen*
>
> //

> ---
> *#include <ostream.h>*
> *Class B*
> *{*
> *public:*
> *virtual void print()*

Try this :

```
virtual void print() const
{
    cout << "printB" << std::endl;
}
```

Note the const at the end of the function decl. This indicates that the "this" parameter to the function is const.

```
> {
> cout << "printB" << std::endl;
> }
> };
>
> class A
```

```
> {  
> public:  
> virtual void print(const B* b)  
> {  
> cout << "printA" << std::endl;  
> b->print();  
> }  
> };  
>  
> main()  
> {  
> cout << "test" << std::endl;  
> A* a;  
> B* b;  
> a = new A();  
> b = new B();  
> a->print(b);  
> }  
>  
>  
>
```