Subject: Re: |SOLVED] Vector of object: cast to inherited class
Posted by Xemuth on Tue, 17 Sep 2019 07:55:00 GMT
View Forum Message <> Reply to Message

OK, I got it, If we don't define Hello() as Virtual then if we do this kind of thing :

```
 B b;
 A* a = &b;
 a->Hello(); //Printing "Hello from A"
```

Since Hello() is static defined C++ will only use type of declared variable to know what function he must call.
if "a" was B* then we would have call B::Hello() etc...
However, if we set Hello() as virtual in A and redifine it in B then the function C++ must call will not be define at compilation but during the runtime, for each call of Hello() C++ will test the real type of Object (I don't know how he do it) then call the right definition (based on type of object) of the function.

```
 B b;
 A* a = &b; //here Hello() is virtual in A  and redefined in B
 a->Hello();//C++ try to know real type of "a" and see it's a B object (or maybe a B* even if I store it in A*) then Printing "Hello from B"
```

:d :d