Should I use `this` within a class?

Asked 9 years, 11 months ago Active 2 years, 11 months ago Viewed 34k times

35

Within a member function of a class in C++, does it make a difference, if I use `this->dataMember` or just `dataMember`? What is considered better style? Is there any performance difference?

(I am not talking about the case where a local variable has the same name as the data member, in which case you must, to my knowledge, use `this->` to distinguish between them.)

`c++` `class` `this`

Share Edit Follow Flag

asked Mar 6, 2012 at 19:52
Ben
14.2k • 16 • 82 • 123

7 Answers

Active | Oldest | Votes

44

As a general rule, it's a question of local conventions. Most of the places I've seen do not use `this->` except when necessary, and that's the convention I prefer as well, but I've heard of people who prefer to use it systematically.

There are two cases when it is necessary. The first is if you've hidden the name with the same name in local scope; if e.g. you have a member named `toto`, and you also named your function argument `toto`. Many coding conventions mark either the member or argments to avoid this case, e.g. all member names start with `my` or `m_`, or a parameter name will start with `the`.

The other case is that `this->` can be used in a template to make a name dependent. This is relevant if a template class inherits from a dependent type, and you want to access a member of the base, e.g.:

```
template <typename T>
class Toto : public T
{
public:
    int f()
    {
        return this->g();
    }
};
```

Without the `this->` here, `g()` would be a non-dependent name, and the compiler would look it up in the context of the template definition, without taking the base class into consideration.

Share Edit Follow Flag

answered Mar 6, 2012 at 20:06
James Kanze
146k • 15 • 173 • 324

2 This is a great answer, I hadn't considered the template case. – Joshua Hedges Aug 8, 2017 at 20:23

11

I always use `this` when calling member *functions*.

1. It turns the function name into a dependent name so that base class member functions are found within a class template.
2. It suppresses argument-dependent lookup. ADL has its advantages, but it can lead to surprising behavior, and I like it if it's not getting in my way.
3. It has no real disadvantages, and so I use it for all member function calls for consistency reasons.
4. I program in Python a lot where an explicit `self` is mandatory, so it's not a real burden for me.

But for data members I use it only when necessary because there is no ADL taking place. To answer your specific questions:

> Within a member function of a class in C++, does it make a difference, if I use this->dataMember or just dataMember?

Yes, if this is within a class template. Then `dataMember` is considered a non-dependent name, which can lead to semantic differences. For example:

```
#include <iostream>

int i = 1;

struct R {
  int i;
  R(): i(2) { }
};

template<typename T>
struct S: T {
  void f() {
    std::cout << i << ' '     // selects ::i
              << this->i      // selects R::i
              << std::endl;
  }
};

int main() {
  S<R>().f();
}
```

> What is considered better style?

I don't think that there is a strong opinion within the community about this. Use either style, but be consistent.

> Is there any performance difference?

I'm pretty sure there isn't.

Share Edit Follow Flag

answered Mar 6, 2012 at 21:07
Philipp
45.3k • 12 • 80 • 107

8

This is a matter of style. Some people like the extra `this->` to make it more obvious that you are accessing a class member. But if you feel it's obvious enough without it, there will be no difference in the generated code or performance.

(Besides the case you mentioned with overlapping scopes, `this->` can also be mandatory in a template when trying to name a member of a type-dependent base class.)

Share Edit Follow Flag

answered Mar 6, 2012 at 20:04
aschepler
68.2k • 8 • 100 • 150

6

it's simply redundant to use `this->` to call members, unless you want to semantically distinguish between locals and members quickly. a lot of people use the `m_` prefix for class members, to avoid writing `this->` all the time.

Share Edit Follow Flag

answered Mar 6, 2012 at 19:54
Not_a_Golfer
43k • 10 • 122 • 88

2

use this when you have a hidden/private member =) in any other case it does not make a difference =)

from the IBM information center i quote the following

> Unless a class member name is hidden, using the class member name is equivalent to using the class member name with the this pointer and the class member access operator (->).

Share Edit Follow Flag

edited Mar 10, 2019 at 1:10
chepe263
2,694 • 20 • 38

answered Mar 6, 2012 at 19:55
Sebastian Flückiger
5,368 • 8 • 31 • 68

0

If a template function makes a call to a member function such that the call does not depend on any template parameters, `this->` can be used to help the compiler as an alternative to `MyUtopicClass<int, double, double>::vin()`.

Share Edit Follow Flag

answered Mar 6, 2012 at 20:02
perreal
89k • 19 • 143 • 171

3 Careful: if `vin` is a virtual function, then `this->vin()` and `MyUtopicClass<int, double, double>::vin()` are different. – aschepler Mar 6, 2012 at 20:05

0

using "this->" is better (you are sure it's the members) but it's doesn't make a difference

Share Edit Follow Flag

answered Mar 6, 2012 at 19:57
Pben
1,061 • 7 • 12