

Home

PUBLIC

Questions

Tags

Users

Companies

COLLECTIVES

Explore Collectives

TEAMS

Create free Team

Why is copy constructor not allowed pass by value? [duplicate]

Asked 10 years, 4 months ago Modified 10 years, 4 months ago Viewed 19k times

Ask Question

This question already has answers here:

Closed 10 years ago.

Possible Duplicate:

[Why should the copy constructor accept its parameter by reference in C++?](#)
[Can a object be passed as value to the copy constructor](#)

Consider this piece of code:

```
class complex{
private:
    double re, im;
public:
    complex(double _re, double _im):re(_re),im(_im){}
    complex(complex c):re(c.re),im(c.im){}
};
```

When compiled, I got an error message: `invalid constructor; you probably meant 'complex (const complex&)'`

In the book `C++ Programming Language`, it is written that:

The copy constructor defines what copying means – including what copying an argument means – so writing

```
complex : complex(complex c) :re(c.re) , im(c.im) {} // error
```

is an error because any call would have involved an infinite recursion.

Why does this cause infinite recursion? It doesn't make sense.

c++ constructor

Share Edit Follow Flag

edited May 23, 2017 at 12:01
Community Bot
1 1

asked Dec 8, 2011 at 19:24
Amumu
16.6k 25 80 126

Add a comment

3 Answers

Sorted by: Highest score (default)

Passing by value means that the parameter is *copied* into the function. That calls the copy constructor.
If your copy constructor parameter is pass-by-value... It would call itself... over and over again...

Share Edit Follow Flag

answered Dec 8, 2011 at 19:26
Mystical
451k 45 326 325

So I got another question: which is the proper copy constructor function? I know const reference is the correct one, but if it is the correct one, shouldn't compiler exclude the pass by value case, since it's not the correct signature, thus no recursion should occur? – Amumu Dec 8, 2011 at 19:37

Can you clarify on what you mean by "exclude the pass by value case"? – Mystical Dec 8, 2011 at 19:41

Which means it should only pick const reference as the function for copying. Every thing else is excluded. Why does it apply the function with pass by value here? Or both are copy constructors, except the pass by value cannot be modified and is used to pass to other functions rather than its constructor? – Amumu Dec 8, 2011 at 19:44

Ah I think I see what you mean. You're saying that if you had two overloaded constructors: `complex(complex c)` and `complex(const complex &c)` ... That's an interesting corner case that I'm not sure about. – Mystical Dec 8, 2011 at 19:46

Add a comment

Passing by value (rather than by reference) means a copy needs to be made. So passing by value into your copy constructor means you need to make a copy before the copy constructor is invoked, but to make a [copy you first need to call the copy constructor](#).

Share Edit Follow Flag

answered Dec 8, 2011 at 19:26
Laurence Gonsalves
131k 32 230 283

Add a comment

When a function is passed with a parameter, local variable is created corresponding to the parameter and is copied with the argument passed on to the function invocation. Hence when the function is invoked, copy constructor of the function will be invoked to copy the argument passed to the invocation to the local variable created. This results in a endless loop.

Where as when a reference is passed on to the function , local variable is not created corresponding to the parameter.

Share Edit Follow Flag

answered Dec 8, 2011 at 19:45
notytony
982 2 10 13

Add a comment

Not the answer you're looking for? Browse other questions tagged `c++` `constructor` or [ask your own question](#).

The Overflow Blog

- Use Git tactically
- You should be reading academic computer science papers

Featured on Meta

- How might the Staging Ground & the new Ask Wizard work on the Stack Exchange...
- The Future of our Jobs Ad slots
- Review our technical responses for the 2022 Developer Survey
- Staging Ground Workflow: Listings, Filters, Quality Control, and Notifications

Hot Meta Posts

21 Too many [options]?

Linked

- 138 Why should the copy constructor accept its parameter by reference in C++?
- 6 Can a object be passed as value to the copy constructor
- 2 Copy constructors
- 1 C++ has two copy constructors, both by reference and by value?

Related

- 280 Why do we not have a virtual constructor in C++?
- 675 Why do this() and super() have to be the first statement in a constructor?
- 8 Templated copy-constructor fails with specific templated type
- 2 Object slicing with private copy constructor in derived class in C++
- 11 Copy Constructor Needed with temp object
- 1 C++ has two copy constructors, both by reference and by value?
- 27 How does Stroustrup take a non-const reference to a temporary?
- 5 what if C++ class contains both const reference and non-const reference copy constructor?
- 15 What is this strange copy constructor error complaining about?
- 154 How can I make my class immune to the "auto value = copy of proxy" landmine in C++?

Hot Network Questions

- When I burn off a very large number of calories, where does the weight go?
- ran relaxed MIP and got some integer results; constrain MIP to match those integers?
- Can a civilization develop Urban Centers without Agriculture?
- How do I count this rhythm? What is the name of this? It has a triplet but also an 8th note?
- How to write an equal sign above a letter

more hot questions



STACK OVERFLOW

Questions
Help

PRODUCTS

Teams
Advertising
Collectives
Talent

COMPANY

About
Press
Work Here
Legal
Privacy Policy
Terms of Service
Contact Us
Cookie Settings
Cookie Policy

STACK EXCHANGE NETWORK

Technology
Culture & recreation
Life & arts
Science
Professional
Business
API
Data

Blog Facebook Twitter LinkedIn Instagram