# Passing member functions to std::thread [duplicate]



**Possible Duplicate:**
Start thread with member function

I have recently been playing around with the new std::thread library in c++11 and I came across a problem. When i try to pass a classes function into a new thread, it gives me an error (I dont have the exact error text right now since im away from home) I had a class like this

```
class A
{
    void FunctA();
    void FunctB();

    void run()
    {
        std::thread t(FunctA);
        std::thread r(FunctB);
    }
}
```

What am I doing wrong?

c++ multithreading c++11

edited May 23 at 12:17
Community♦
1 1

asked Jun 15 '12 at 20:32
Whyrusleeping
431 2 5 17

**marked** as duplicate by mfontanini, ildjarn, Christian Rau, Jerry Coffin, Evan Mulawski Jun 16 '12 at 21:05

This question has been asked before and already has an answer. If those answers do not fully address your question, please ask a new question.

1 have u tried making FunctA/FunctB static? – Anders K. Jun 15 '12 at 20:39

## 3 Answers

```
class A
{
    void FunctA();
    void FunctB();

    void run()
    {
        std::thread t(&A::FunctA, this);
        std::thread r(&A::FunctB, this);
    }
};
```

Pointers to member functions are different from pointers to functions, syntax of calling them is different, as well, and requires instance of class. You can just pass pointer to instance as second argument of `std::thread` constructor.

answered Jun 15 '12 at 20:36
Griwes
6,436 2 23 61

1 what if they had args? how would you pass them? – kdubs Jul 26 '15 at 21:10

@kdubs: ...as further arguments to the constructor call. Please do read the docs in the future. – Griwes Jul 27 '15 at 10:49

This code fails! even if you add the `public` keyword.. – towi_parallelism Feb 5 '16 at 16:56

1 I know this is old, but I've been looking for the solution to this. I've seen this answer posted in a few places now, and while this compiles it causes a segfault. Why is this? – Iron Attorney May 10 '16 at 16:18

@IronAttorney maybe you need to join or detach the thread. – Steve M Jun 18 '16 at 17:39



```
class A
{
  public:
    A( FunctA &fa ) : fa(fa) {}

    struct FunctA
    {
      virtual void operator()() = 0;
    }
    struct FunctB
    {
      void operator()()
      {
        // thread B
      }
    }
    FunctA &fa;
    FunctB fb;

    void run()
    {
        std::thread t(fa);
        std::thread r(fb);
    }
}
```

boost documentation

edited Jun 18 '12 at 8:30

answered Jun 15 '12 at 20:38

Naszta
**5,750** 2 19 41

It's dumb. As in: not smart. – Griwes Jun 15 '12 at 20:39

@Griwes : It's unnecessarily verbose, but it *would* work, so I don't think it warrants a downvote. – ildjarn Jun 15 '12 at 20:41

3 @ildjarn, right, maybe not worth a downvote, but still dumb. – Griwes Jun 15 '12 at 20:41

In this way the method could be virtual or pure virtual. – Naszta Jun 16 '12 at 8:03

I think, the problem is that you can't get pointer to member function in a way similar to functions. Here you will find more information about this.

Also, it would be much easier to answer, if you provided compipler error text.

answered Jun 15 '12 at 20:34

Spo1ler
**1,115** 1 11 21