

Join the Stack Overflow Community

Stack Overflow is a community of 7.3 million programmers, just like you, helping each other.
Join them; it only takes a minute:

Sign up

THREAD ERROR: invalid use of non-static member function [duplicate]

We have 3 open jobs ▼



IMC – Explore. Engage. Expand.

Learn more

This question already has an answer here:
[Passing member functions to std::thread \[duplicate\]](#) 3 answers

I'm trying to understand threads in C++ but I don't know how to solve this problem.
I want to call two threads to run the function called "createS" but I get this error:

error: invalid use of non-static member function

I've read other questions about this topic but I really don't understand how to make my code works.
Could someone explain me what I'm doing wrong and try to help my find a solution?

test_class.cpp

```
void test_class::generateS(){  
    map1=new multimap<double,vector<int>>;  
    map2=new multimap<double,vector<int>>;  
  
    thread thread_1( createS, 0, nCells/2, map1 );  
    thread thread_2( createS, nCells/2, nCells, map2);  
  
    thread_1.join();  
    thread_2.join();  
}  
  
void test_class::createS(int startP, int endP, Costs *mapPointer){  
    //i do some stuff  
}
```

test_class.h

```
void createS(int start, int end, Costs *mapPointer);  
void generateS();
```

c++ multithreading compiler-errors non-static

edited Jan 5 at 1:59

soundslikeodd

1,057 1 6 24

asked Jan 5 at 1:45

Jack

29 7

marked as duplicate by [M.M](#) c++ Jan 5 at 2:04

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](#).

Try making CreateS() class static. – [brian beuning](#) Jan 5 at 2:01

1 Answer

```
thread thread_1(&test_class::createS, this, 0, nCells/2, map1);
```

26/06/2017

c++ - THREAD ERROR: invalid use of non-static member function - Stack Overflow

```
thread thread_2(&test_class::createS, this, nCells/2, nCells, map2);
```

Note: if `createS` does not depend on object state, better make it a `static` class member and call the way you did.

answered Jan 5 at 2:01



A.S.H

20.4k

3

12

32