

```

1  once per thread, move to therad, ICE API from get data, lambda copt
2
3  // copy this object
4  auto l1 = [thisCopy=*this] { std::cout << thisCopy.name << '\n'; };//C++14
5  auto l1 = [*this] { std::cout << name << '\n'; };// C++17
6
7  //lang/inlinethreadlocal.hpp
8  #include <string>
9  #include <iostream>
10 struct MyData {
11     inline static std::string gName = "global"; // unique in program
12     inline static thread_local std::string tName = "tls"; // unique per thread
13     std::string lName = "local"; // for each object
14     ...
15     void print(const std::string& msg) const {
16         std::cout << msg << '\n';
17         std::cout << "- gName: " << gName << '\n';
18         std::cout << "- tName: " << tName << '\n';
19         std::cout << "- lName: " << lName << '\n';
20     }
21 };
22 inline thread_local MyData myThreadData; // one object per thread
23
24 //lang/inlinethreadlocal1.cpp
25 #include "inlinethreadlocal.hpp"
26 #include <thread>
27 void foo();
28 int main()
29 {
30     myThreadData.print("main() begin:");
31     myThreadData.gName = "thread1 name";
32     myThreadData.tName = "thread1 name";
33     myThreadData.lName = "thread1 name";
34     myThreadData.print("main() later:");
35     std::thread t(foo);
36     t.join();
37     myThreadData.print("main() end:");
38 }
39
40

```