

CS246—Known C++20 Issues

November 22, 2024

1 Compilation errors

1. Compiling `<string>` before `<iostream>` and `<fstream>` causes a compilation error.

```
$ g++20h string iostream
In file included from /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/locale_classes.h:883,
                 from /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/ios_base.h:41,
                 from /usr/local/gcc-14.2.0/include/c++/14.2.0/ios:44,
                 from /usr/local/gcc-14.2.0/include/c++/14.2.0/ostream:40,
                 from /usr/local/gcc-14.2.0/include/c++/14.2.0/iostream:41:
/usr/local/gcc-14.2.0/include/c++/14.2.0/bits/locale_classes.tcc: In function 'const _Facet* std::_try_use_f
/usr/local/gcc-14.2.0/include/c++/14.2.0/bits/locale_classes.tcc:108:7: error: template argument 3 is invalid
 108 |         _GLIBCXX_STD_FACET(codecvt<char, char, mbstate_t>);
      |
...
$ g++20h iostream string
```

This is a known bug in the current compiler. Always compile `<string>` *last*!

2 Other issues

The `gdb` debugger doesn't work very well with user-defined modules compiled with `g++-14`. If all of your code is in a single C++20 file, with no user-defined modules, it works as expected.

As soon as the code is split so that classes are put into 1 or more user-defined modules, `gdb` stops in the system library code such as `std::string` allocators instead of the break point set in the class method. (Note as well that the `info skip` shows that nothing is currently being skipped.)

```
$ gdb ./a.out
[ gdb intro messages deleted ]
Reading symbols from ./a.out...
(gdb) info skip
Not skipping any files or functions.
(gdb) l main.cc:5
1      import <iostream>;
2      import student;
3      using namespace std;
4
5
6      int main() {
7          Student s{"Mackenzie Chao"};
8          cout << s << endl;
```

```

9         s.setAssns( -5 );
10        s.setMidterm( 110 );
(gdb) break 9
Breakpoint 1 at 0x402344: file main.cc, line 9.
(gdb) l Student-impl.cc:25
20        const string Student::getAssns() const { return gradeToReturn(assns); }
21        const string Student::getMidterm() const { return gradeToReturn(midterm); }
22        const string Student::getFinal() const { return gradeToReturn(final); }
23        const string Student::getName() const { return name; }
24
25        int Student::capGrade( int grade ) const {
26            if ( grade == NOT_YET_ASSIGNED ) return NOT_YET_ASSIGNED;
27            if ( grade < 0 ) return 0;
28            if ( grade > 100 ) return 100;
29            return grade;
(gdb) break 26
Breakpoint 2 at 0x402da2: /u1/cs246/teaching/handouts/C++20-Errors/gdb-V2/Student-impl.cc:26. (7 locations)
(gdb) run
Starting program: /u1/cs246/teaching/handouts/C++20-Errors/gdb-V2/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

Breakpoint 2, std::__new_allocator<char>::__new_allocator (this=0x7fffffff7cf) at Student-impl.cc:957143
(gdb) n
std::allocator_traits<std::allocator<char> >::select_on_container_copy_construction (__rhs=...) at /usr/local/gcc-1
584         { return __rhs; }
(gdb)
__gnu_cxx::__alloc_traits<std::allocator<char>, char>::_S_select_on_copy (__a=...) at /usr/local/gcc-14.2.0/include
98         { return _Base_type::select_on_container_copy_construction(__a); }
(gdb)
std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string (this=0x7fffffff850,
553         : _M_dataplus(_M_local_data(),
...
(gdb) n

Breakpoint 2, std::allocator<char>::~~allocator (this=<optimized out>, __in_chrg=<optimized out>) at Student-impl.cc:
(gdb)
std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::basic_string (this=0x7fffffff7d0,
556         _M_construct(__str._M_data(), __str._M_data() + __str.length(),
(gdb)
558         }
(gdb)
_ZNKW7student7Student8getAssnsB5cxx11Ev (this=0x7fffffff850) at Student-impl.cc:957155
(gdb)
_ZW7student1sRSoRKS_7Student (out=..., s=...) at Student-impl.cc:957145
(gdb)
...

```

<https://stackoverflow.com/questions/6219223/can-i-prevent-debugger-from-stepping-into-boost-42721326#42721326> suggests creating a .gdbinit file in your root directory that contains the following Python code in order to skip the system library STL code.

```

# skip all STL source files
define skipstl
python
# get all sources loadable by gdb

```

```

def GetSources():
    sources = []
    for line in gdb.execute('info sources', to_string=True).splitlines():
        if line.startswith("/"):
            sources += [source.strip() for source in line.split(",")]
    return sources

# skip files of which the (absolute) path begins with 'dir'
def SkipDir(dir):
    sources = GetSources()
    for source in sources:
        if source.startswith(dir):
            gdb.execute('skip file %s' % source, to_string=True)

# apply only for c++
if 'c++' in gdb.execute('show language', to_string=True):
    SkipDir("/usr")
end
end

define hookpost-run
    skipstl
end

```

If we do this and try running `gdb` again, once we've set the break points and have typed the `run` command, `info skip` shows that we're skipping all of the STL system library code. Unfortunately, if there are STL method/function calls that coincide with our break points, we'll have to explicitly disable the break points associated with the STL calls (e.g. 2.1, 2.2, 2.4-2.6, 3.1, 3.2, 3.4-3.6) and leave enabled the break point associate with our own implementation (e.g. 2.3, 3.3). We still won't be able to see what code we're stepping through, as shown in the following example, but we *can* print/display variable values, which can help us at least somewhat.

In the worst case scenario, you could concatenate the files (in the correct dependency order) and remove the module declaration code in order to be able to debug fully.

```

$ gdb ./a.out
[ gdb intro messages deleted]
Reading symbols from ./a.out...
(gdb) break 9
Breakpoint 1 at 0x402344: file main.cc, line 9.
(gdb) l Student-impl.cc:25
20     const string Student::getAssns() const { return gradeToReturn(assns); }
21     const string Student::getMidterm() const { return gradeToReturn(midterm); }
22     const string Student::getFinal() const { return gradeToReturn(final); }
23     const string Student::getName() const { return name; }
24
25     int Student::capGrade( int grade ) const {
26         if ( grade == NOT_YET_ASSIGNED ) return NOT_YET_ASSIGNED;
27         if ( grade < 0 ) return 0;
28         if ( grade > 100 ) return 100;
29         return grade;
(gdb) break 26
Breakpoint 2 at 0x402da2: /u1/cs246/teaching/handouts/C++20-Errors/gdb-V2/Student-impl.cc:26. (7 locations)
(gdb) l
30         } // Student::capGrade

```

```

31
32     const string Student::gradeToReturn( int grade ) const {
33         if ( grade == NOT_YET_ASSIGNED ) return TBD;
34         ostringstream oss;
35         oss << grade;
36         return oss.str();
37     } // Student::setGradeToReturn
(gdb) break 33
Note: breakpoint 2 also set at pc 0x402da2.
Note: breakpoint 2 also set at pc 0x402dca.
Note: breakpoint 2 also set at pc 0x402de7.
Note: breakpoint 2 also set at pc 0x4032b2.
Note: breakpoint 2 also set at pc 0x4032bf.
Note: breakpoint 2 also set at pc 0x40347b.
Note: breakpoint 2 also set at pc 0x4034a0.
Breakpoint 3 at 0x402da2: /u1/cs246/teaching/handouts/C++20-Errors/gdb-V2/Student-impl.cc:33. (7 locations)
(gdb) run
Starting program: /u1/cs246/teaching/handouts/C++20-Errors/gdb-V2/a.out
[Thread debugging using libthread_db enabled]
Using host libthread_db library "/lib/x86_64-linux-gnu/libthread_db.so.1".

```

Breakpoint 2, std::__new_allocator<char>::__new_allocator (this=0x7fffffff7cf) at Student-impl.cc:957143

```

(gdb) info skip
Num  Enb Glob File                                RE Function
1    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/new n <none>
2    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/type_traits n <none>
3    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/move.h n <none>
4    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/ptr_traits.h n <none>
5    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/stl_construct.h n <none>
6    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/basic_string.h n <none>
7    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/alloc_traits.h n <none>
8    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/x86_64-linux-gnu/bits/c++config.h n <none>
9    y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/allocator.h n <none>
10   y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/basic_string.tcc n <none>
11   y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/new_allocator.h n <none>
12   y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/char_traits.h n <none>
13   y    n /usr/local/gcc-14.2.0/include/c++/14.2.0/bits/stl_iterator_base_types.h n <none>
--Type <RET> for more, q to quit, c to continue without paging--
Quit
(gdb) c
Continuing.

```

Breakpoint 2, std::__new_allocator<char>::__new_allocator (this=0x7fffffff850) at Student-impl.cc:957143

```

(gdb) info break 2
Num  Type          Disp Enb Address          What
2    breakpoint    keep y  <MULTIPLE>
      breakpoint already hit 2 times
2.1          y  0x000000000402da2 in std::allocator<char>::~~allocator() at Student-impl.cc:957143
2.2          y  0x000000000402dca in std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>>::basic_string(const char*, const char*, const std::allocator<char>&) at Student-impl.cc:957143
2.3          y  0x000000000402de7 in _ZW7studentlsRSorKS_7Student at Student-impl.cc:957145
2.4          y  0x0000000004032b2 in std::__new_allocator<char>::__new_allocator(std::__new_allocator<char> const&) at Student-impl.cc:957143
2.5          y  0x0000000004032bf in std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char>>::basic_string(const char*, const char*, const std::allocator<char>&) at Student-impl.cc:957143
2.6          y  0x00000000040347b in std::__new_allocator<char>::__new_allocator(std::__new_allocator<char> const&) at Student-impl.cc:957143
2.7          y  0x0000000004034a0 in std::allocator<char>::~~allocator() at Student-impl.cc:957143
(gdb) disable break 2.1

```

```
(gdb) disable break 2.2
(gdb) disable break 2.4
(gdb) disable break 2.5
(gdb) disable break 2.6
(gdb) disable break 2.7
(gdb) c
```

Continuing.

Breakpoint 3, std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_Alloc_hider::_Alloc

```
(gdb) info break 3
```

Num	Type	Disp	Enb	Address	What
3	breakpoint	keep y		<MULTIPLE>	
breakpoint already hit 3 times					
3.1		y		0x0000000000402da2	in std::allocator<char>::~~allocator() at Student-impl.cc:957143
3.2		y		0x0000000000402dca	in std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_Alloc_hider::_Alloc at Student-impl.cc:957143
3.3		y		0x0000000000402de7	in _ZW7student1sRSoRKS_7Student at Student-impl.cc:957145
3.4		y		0x00000000004032b2	in std::__new_allocator<char>::~__new_allocator(std::__new_allocator<char> &) at Student-impl.cc:957145
3.5		y		0x00000000004032bf	in std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >::_Alloc_hider::_Alloc at Student-impl.cc:957143
3.6		y		0x000000000040347b	in std::__new_allocator<char>::~__new_allocator(std::__new_allocator<char> &) at Student-impl.cc:957145
3.7		y		0x00000000004034a0	in std::allocator<char>::~~allocator() at Student-impl.cc:957143

```
(gdb) disable break 3.1
(gdb) disable break 3.2
(gdb) disable break 3.4
(gdb) disable break 3.5
(gdb) disable break 3.6
(gdb) disable break 3.7
(gdb) c
```

Continuing.

Breakpoint 2, _ZW7student1sRSoRKS_7Student (out=..., s=...) at Student-impl.cc:957145

```
(gdb) s
```

```
(gdb) p s
```

```
$1 = (const Student@student &) @0x7fffffff850: {static TBD = {
  _M_dataplus = {<std::allocator<char> > = {<No data fields>},
  _M_p = 0x41a2b0 "to be determined", _M_string_length = 16, {
  _M_local_buf = "\020", '\000' <repeats 14 times>,
  _M_allocated_capacity = 16}}, name = "Mackenzie Chao", assns = -1,
  midterm = -1, final = -1}
```

```
(gdb) s
```

```
_ZNKW7student7Student7getNameB5cxx11Ev (this=0x7fffffff850) at Student-impl.cc:957158
```

```
(gdb)
```

```
_ZW7student1sRSoRKS_7Student (out=..., s=...) at Student-impl.cc:957145
```

```
(gdb)
```

```
(gdb)
```

```
_ZNKW7student7Student8getAssnsB5cxx11Ev (this=0x7fffffff850) at Student-impl.cc:957155
```

```
(gdb)
```

```
_ZNKW7student7Student13gradeToReturnB5cxx11Ei (this=0x7fffffff850, grade=-1) at Student-impl.cc:957168
```

```
(gdb) s
```

```
_ZNKW7student7Student8getAssnsB5cxx11Ev (this=0x7fffffff850) at Student-impl.cc:957155
```

```
(gdb)
```

```
_ZW7student1sRSoRKS_7Student (out=..., s=...) at Student-impl.cc:957145
```

```
(gdb) c
```

Continuing.

```
Student{Mackenzie Chao (to be determined, to be determined, to be determined)}
```

```

Breakpoint 1, main () at main.cc:9
9          s.setAssns( -5 );
(gdb) s
_ZNW7student7Student8setAssnsEi (this=0x7fffffff850, a=-5) at Student-impl.cc:957152
(gdb) s
_ZNKW7student7Student8capGradeEi (this=0x7fffffff850, grade=-5) at Student-impl.cc:957161
(gdb)
(gdb)
(gdb)
main () at main.cc:10
10          s.setMidterm( 110 );
(gdb) c
Continuing.

Breakpoint 2, _ZW7student1sRSorKS_7Student (out=..., s=...) at Student-impl.cc:957145
(gdb)
Continuing.
Student{Mackenzie Chao (0, 100, 99)}
[Inferior 1 (process 21821) exited normally]
(gdb) q

```

CSCF has provided the following information on the error:

“I think there is a problem with g++ and how it’s importing <string> and <iostream> together.

If I comment out `import <string>` from `Collection-impl.cc`, and set breakpoint at `Collection-impl.cc:34`, then gdb stops at the line `"if ( !root ) {"` which is what you’d expect. I can also comment out only `import <iostream>` and it works properly. I just cannot have both <string> and <iostream> imported for some reason.

In a simpler example I made, importing <string> before <iostream> in my `*-impl.cc` file apparently makes the problem go away too. But this doesn’t work in the graph example.

I downloaded the latest g++ code from GitHub (a dev copy of g++-15, not yet released) and it works a bit better. If I compile the graph example like this with g++ v15:

```

g++ -std=c++20 -fmodules-ts -c -x c++-system-header string iostream fstream
g++ -std=c++20 -fmodules-ts -Wall -g *.cc

```

Then setting breakpoint `Collection-impl.cc:34` will stop at `"if ( !root ) {"` and gdb reports correct line number 34. Instead of the 990765 line number.

Note I had to put string first in the g++ command. If I put string later, then g++ 15 still imports <string> in odd way and gdb can’t debug it properly. Putting string first is not possible in g++ 14 because of a bug Brad already reported:

https://gcc.gnu.org/bugzilla/show_bug.cgi?id=116671

This bug report is what led me to trying out g++ 15, which is still in development.”