

```
1: // $Id: teststack.cpp,v 1.9 2014-04-09 17:29:51-07 - - $
2:
3: #include <iostream>
4: #include <string>
5: #include <vector>
6: using namespace std;
7:
8: #include "iterstack.h"
9:
10: typedef iterstack<string>::const_reverse_iterator stack_ritor;
11:
12: int main (int argc, char **argv) {
13:     vector<string> args (&argv[1], &argv[argc]);
14:
15:     iterstack<string> stk;
16:     cout << endl << "Pushing loop:" << endl;
17:     for (const auto& arg: args) {
18:         cout << arg << endl;
19:         stk.push_back (arg);
20:     }
21:     cout << endl << "Iteration loop:" << endl;
22:     for (const auto& elt: stk) cout << *itor << endl;
23:
24:     cout << endl << "Popping loop:" << endl;
25:     while (not stk.empty()) {
26:         cout << stk.back() << endl;
27:         stk.pop_back();
28:     }
29:     return 0;
30: }
31:
32: /*
33: //TEST// valgrind --leak-check=full --show-reachable=yes \
34: //TEST//      --log-file=teststack.out.grind \
35: //TEST//      teststack foo bar baz qux quux >teststack.out 2>&1
36: //TEST// mkpspdf teststack.ps teststack.cpp* iterstack.h teststack.out*
37: */
38:
```

```
1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting teststack.cpp
2: teststack.cpp: $Id: teststack.cpp,v 1.9 2014-04-09 17:29:51-07 - - $
3: g++ -g -O0 -Wall -Wextra -std=gnu++11 teststack.cpp -o teststack -lm
4: In file included from /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/ve
ctor:64:0,
5:         from teststack.cpp:5:
6: /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/bits/stl_vector.h:230:54
: error: 'typedef class std::reverse_iterator<__gnu_cxx::__normal_iterator<cons
t std::basic_string<char>*, std::vector<std::basic_string<char>, std::allocator
<std::basic_string<char> > > > std::vector<std::basic_string<char>, std::allo
cator<std::basic_string<char> > >::const_reverse_iterator' is inaccessible
7:         typedef std::reverse_iterator<const_iterator> const_reverse_iter
ator;
8:                                     ^
9: teststack.cpp:10:28: error: within this context
10:     typedef iterstack<string>::const_reverse_iterator stack_ritor;
11:                                     ^
12: In file included from /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/ve
ctor:64:0,
13:         from teststack.cpp:5:
14: /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/bits/stl_vector.h: In fu
nction 'int main(int, char**)':
15: /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/bits/stl_vector.h:901:7:
error: 'void std::vector<_Tp, _Alloc>::push_back(const value_type&) [with _Tp
= std::basic_string<char>; _Alloc = std::allocator<std::basic_string<char> >; s
td::vector<_Tp, _Alloc>::value_type = std::basic_string<char>]' is inaccessible
16:         push_back(const value_type& __x)
17:         ^
18: teststack.cpp:19:25: error: within this context
19:         stk.push_back (arg);
20:                     ^
21: teststack.cpp:22:40: error: 'itor' was not declared in this scope
22:         for (const auto& elt: stk) cout << *itor << endl;
23:                                     ^
24: teststack.cpp:22:21: warning: unused variable 'elt' [-Wunused-variable]
25:         for (const auto& elt: stk) cout << *itor << endl;
26:                     ^
27: In file included from /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/ve
ctor:64:0,
28:         from teststack.cpp:5:
29: /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/bits/stl_vector.h:855:7:
error: 'std::vector<_Tp, _Alloc>::reference std::vector<_Tp, _Alloc>::back() [
with _Tp = std::basic_string<char>; _Alloc = std::allocator<std::basic_string<c
har> >; std::vector<_Tp, _Alloc>::reference = std::basic_string<char>&]' is ina
ccessible
30:         back()
31:         ^
32: teststack.cpp:26:24: error: within this context
33:         cout << stk.back() << endl;
34:                     ^
35: In file included from /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/ve
ctor:64:0,
36:         from teststack.cpp:5:
37: /opt/rh/devtoolset-2/root/usr/include/c++/4.8.1/bits/stl_vector.h:937:7:
error: 'void std::vector<_Tp, _Alloc>::pop_back() [with _Tp = std::basic_strin
g<char>; _Alloc = std::allocator<std::basic_string<char> >]' is inaccessible
38:         pop_back()
39:         ^
```

04/11/14
12:22:00

\$cmpps109-wm/Assignments/asg2-dc-bigint/misc/
teststack.cpp.log

2/2

```
40: teststack.cpp:27:20: error: within this context
41:     stk.pop_back();
42:         ^
```

```
1: // $Id: iterstack.h,v 1.3 2014-04-09 17:26:07-07 - - $
2:
3: //
4: // The class std::stack does not provide an iterator, which is
5: // needed for this class. So, like std::stack, class iterstack
6: // is implemented on top of a container.
7: //
8: // We use private inheritance because we want to restrict
9: // operations only to those few that are approved. All functions
10: // are merely inherited from the container, with only ones needed
11: // being exported as public.
12: //
13: // No implementation file is needed because all functions are
14: // inherited, and the convenience functions that are added are
15: // trivial, and so can be inline.
16: //
17: // Any underlying container which supports the necessary operations
18: // could be used, such as vector, list, or deque.
19: //
20:
21: #ifndef __ITERSTACK_H__
22: #define __ITERSTACK_H__
23:
24: #include <vector>
25: using namespace std;
26:
27: template <typename value_type>
28: class iterstack: private vector<value_type> {
29:     private:
30:         using vector<value_type>::crbegin;
31:         using vector<value_type>::crend;
32:         using vector<value_type>::push_back;
33:         using vector<value_type>::pop_back;
34:         using vector<value_type>::back;
35:         typedef typename vector<value_type>::const_reverse_iterator
36:             const_iterator;
37:     public:
38:         using vector<value_type>::const_reference;
39:         using vector<value_type>::clear;
40:         using vector<value_type>::size;
41:         using vector<value_type>::empty;
42:         const_iterator begin() { return crbegin(); }
43:         const_iterator end() { return crend(); }
44:         inline void push (const value_type& value) { push_back (value); }
45:         inline void pop() { pop_back(); }
46:         inline const value_type& top() const { return back(); }
47: };
48:
49: #endif
50:
```

```
1:
2: Pushing loop:
3: teststack
4: foo
5: bar
6: baz
7: qux
8: quux
9:
10: Iteration loop:
11: quux
12: qux
13: baz
14: bar
15: foo
16: teststack
17:
18: Popping loop:
19: quux
20: qux
21: baz
22: bar
23: foo
24: teststack
```

```
1: ==29941== Memcheck, a memory error detector
2: ==29941== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al
.
3: ==29941== Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright i
nfo
4: ==29941== Command: teststack foo bar baz qux quux
5: ==29941== Parent PID: 29940
6: ==29941==
7: ==29941==
8: ==29941== HEAP SUMMARY:
9: ==29941==      in use at exit: 0 bytes in 0 blocks
10: ==29941==    total heap usage: 10 allocs, 10 frees, 295 bytes allocated
11: ==29941==
12: ==29941== All heap blocks were freed -- no leaks are possible
13: ==29941==
14: ==29941== For counts of detected and suppressed errors, rerun with: -v
15: ==29941== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
```