```
 1: // $Id: fixarray.cpp,v 1.8 2014-05-30 16:43:27-07 - - $
 2:
 3: //
 4: // fixarray - implementation of an int vector, using templates,
 5: // with the array inline in the struct, so size can not be changed.
 6: // also shows checked operator[] instead of get/put.
 7: //
 8:
 9: #include <iostream>
10: #include <stdexcept>
11:
12: using namespace std;
13:
14: ////////////////////////////////////////////////////////////////////
15: // fixarray.h
16: ////////////////////////////////////////////////////////////////////
17:
18: template <typename item_t, size_t size_>
19: class fixarray {
20:    private:
21:       item_t data_[size_];
22:       void range_check (size_t index) const;
23:    public:
24:       fixarray();
25:       fixarray (const fixarray &) = default;             // copy ctor
26:       fixarray &operator= (const fixarray &) = default; // copy op=
27:       ~fixarray() = default;
28:       size_t size() const;
29:       item_t operator[] (size_t index) const; // get: x = a[i];
30:       item_t& operator[] (size_t index); // put: a[i] = x;
31: };
32:
```

```
33:
34: //////////////////////////////////////////////////////////////////
35: // fixarray.cpp
36: //////////////////////////////////////////////////////////////////
37:
38: template <typename item_t, size_t size_>
39: fixarray<item_t,size_>::fixarray() {
40:    for (size_t index = 0; index < size_; ++index) data_[index] = int();
41: }
42:
43: template <typename item_t, size_t size_>
44: void fixarray<item_t,size_>::range_check (size_t index) const {
45:    if (index >= size_) throw out_of_range ("fixarray::range_check");
46: }
47:
48: template <typename item_t, size_t size_>
49: size_t fixarray<item_t,size_>::size() const {
50:    return size_;
51: }
52:
53: template <typename item_t, size_t size_>
54: item_t fixarray<item_t,size_>::operator[] (size_t index) const {
55:    range_check (index);
56:    return data_[index];
57: }
58:
59: template <typename item_t, size_t size_>
60: item_t& fixarray<item_t,size_>::operator[] (size_t index) {
61:    range_check (index);
62:    return data_[index];
63: }
64:
```

```
65:
66: //////////////////////////////////////////////////////////////////////
67: // main.cpp
68: //////////////////////////////////////////////////////////////////////
69:
70: using tenvec = fixarray<int,10>;
71: int main () {
72:    tenvec v1;
73:    v1[3] = 99;
74:    int x = v1[3];
75:    cout << x << endl;
76:    try {
77:       cout << "v1[999] = " << v1[999] << endl;
78:    }catch (out_of_range error) {
79:       cout << error.what() << endl;
80:    }
81:    tenvec v2 = v1;
82:    v2[3] = 1234;
83:    cout << v1[3] << " " << v2[3] << endl;
84:    v2 = v1;
85:    cout << v1[3] << " " << v2[3] << endl;
86:    for (size_t i = 0; i < v1.size(); ++i) cout << v1[i];
87:    cout << endl;
88:    return 0;
89: }
90:
91: //TEST// alias grind='valgrind --leak-check=full --show-reachable=yes'
92: //TEST// grind fixarray >fixarray.out 2>&1
93: //TEST// mkpspdf fixarray.ps fixarray.cpp* fixarray.out*
94:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting fixarray.cpp
    2: fixarray.cpp:
    3:       $Id: fixarray.cpp,v 1.8 2014-05-30 16:43:27-07 - - $
    4: g++ -g -O0 -Wall -Wextra -std=gnu++11 fixarray.cpp -o fixarray -lglut -l
GLU -lGL -lX11 -lm -lrt
    5: rm -f fixarray.o
    6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished fixarray.cpp
```

```
    1: ==21588== Memcheck, a memory error detector
    2: ==21588== Copyright (C) 2002-2012, and GNU GPL'd, by Julian Seward et al
.
    3: ==21588== Using Valgrind-3.8.1 and LibVEX; rerun with -h for copyright i
nfo
    4: ==21588== Command: fixarray
    5: ==21588==
    6: 99
    7: fixarray::range_check
    8: 99 1234
    9: 99 99
   10: 00099000000
   11: ==21588==
   12: ==21588== HEAP SUMMARY:
   13: ==21588==     in use at exit: 0 bytes in 0 blocks
   14: ==21588==   total heap usage: 3 allocs, 3 frees, 199 bytes allocated
   15: ==21588==
   16: ==21588== All heap blocks were freed -- no leaks are possible
   17: ==21588==
   18: ==21588== For counts of detected and suppressed errors, rerun with: -v
   19: ==21588== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
```