```
 1: // $Id: mapleak.cpp,v 1.14 2015-01-26 14:35:42-08 - - $
 2:
 3: //
 4: // Illustrate how leaks may happen for a map.
 5: // Use argv to create a series of objects, each of which has a
 6: // string field and which is put in a map.
 7: // If the first argument is "-", erase all map entries.
 8: //
 9:
10: #include <iostream>
11: #include <map>
12:
13: using namespace std;
14:
15: int seqct = 0;
16: struct object {
17:    int seqnr;
18:    string value;
19:    explicit object (const string& val): seqnr(++seqct), value(val) {}
20: };
21:
22: int main (int argc, char** argv) {
23:    using strobjmap = map<string,object*>;
24:    strobjmap somap;
25:
26:    // Push each element of argv into map as object.
27:    for (int index = 1; index < argc; ++index) {
28:       string arg = argv[index];
29:       somap.emplace (strobjmap::value_type (arg, new object (arg)));
30:    }
31:
32:    // Iterate over the map, printing out the keys and values.
33:    for (auto itor = somap.cbegin(); itor != somap.cend(); ++itor) {
34:       cout << itor->first << " => (" << itor->second->seqnr << ", "
35:            << itor->second->value << ")" << endl;
36:    }
37:
38:    // If the first argument is "-", erase all map entries.
39:    if (argc > 1 && argv[1][0] == '-') {
40:       while (somap.size() > 0) {
41:          auto itor = somap.begin();
42:          object *second = itor->second;
43:          somap.erase (itor);
44:          delete second;
45:       }
46:    }
47:
48:    return 0;
49: }
50:
51: //TEST// alias grind='valgrind --leak-check=full --show-reachable=yes'
52: //TEST// grind mapleak arguments to check on leak >mapleak.out1 2>&1
53: //TEST// grind mapleak - arguments to check on leak >mapleak.out2 2>&1
54: //TEST// mkpspdf mapleak.ps mapleak.cpp* mapleak.out*
55:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting mapleak.cpp
    2: mapleak.cpp:
    3:       $Id: mapleak.cpp,v 1.14 2015-01-26 14:35:42-08 - - $
    4: g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++11 mapleak.cpp -o mapleak -
lglut -lGLU -lGL -lX11 -lrt -lm
    5: rm -f mapleak.o
    6: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished mapleak.cpp
```

```
    1: ==7338== Memcheck, a memory error detector
    2: ==7338== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
    3: ==7338== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright in
fo
    4: ==7338== Command: mapleak arguments to check on leak
    5: ==7338==
    6: arguments => (1, arguments)
    7: check => (3, check)
    8: leak => (5, leak)
    9: on => (4, on)
   10: to => (2, to)
   11: ==7338==
   12: ==7338== HEAP SUMMARY:
   13: ==7338==     in use at exit: 227 bytes in 10 blocks
   14: ==7338==   total heap usage: 16 allocs, 6 frees, 476 bytes allocated
   15: ==7338==
   16: ==7338== 147 bytes in 5 blocks are indirectly lost in loss record 1 of 2
   17: ==7338==    at 0x4A076A5: operator new(unsigned long) (in /opt/rh/devtoo
lset-2/root/usr/lib64/valgrind/vgpreload_memcheck-amd64-linux.so)
   18: ==7338==    by 0x3CC7C9C3C8: std::string::_Rep::_S_create(unsigned long,
 unsigned long, std::allocator<char> const&) (in /usr/lib64/libstdc++.so.6.0.13
)
   19: ==7338==    by 0x3CC7C9CDE4: ??? (in /usr/lib64/libstdc++.so.6.0.13)
   20: ==7338==    by 0x3CC7C9CF32: std::basic_string<char, std::char_traits<ch
ar>, std::allocator<char> >::basic_string(char const*, std::allocator<char> con
st&) (in /usr/lib64/libstdc++.so.6.0.13)
   21: ==7338==    by 0x4043BE: main (mapleak.cpp:28)
   22: ==7338==
   23: ==7338== 227 (80 direct, 147 indirect) bytes in 5 blocks are definitely
lost in loss record 2 of 2
   24: ==7338==    at 0x4A076A5: operator new(unsigned long) (in /opt/rh/devtoo
lset-2/root/usr/lib64/valgrind/vgpreload_memcheck-amd64-linux.so)
   25: ==7338==    by 0x4043D4: main (mapleak.cpp:29)
   26: ==7338==
   27: ==7338== LEAK SUMMARY:
   28: ==7338==    definitely lost: 80 bytes in 5 blocks
   29: ==7338==    indirectly lost: 147 bytes in 5 blocks
   30: ==7338==      possibly lost: 0 bytes in 0 blocks
   31: ==7338==    still reachable: 0 bytes in 0 blocks
   32: ==7338==         suppressed: 0 bytes in 0 blocks
   33: ==7338==
   34: ==7338== For counts of detected and suppressed errors, rerun with: -v
   35: ==7338== ERROR SUMMARY: 1 errors from 1 contexts (suppressed: 6 from 6)
```

```
    1: ==7339== Memcheck, a memory error detector
    2: ==7339== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
    3: ==7339== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright in
fo
    4: ==7339== Command: mapleak - arguments to check on leak
    5: ==7339==
    6: - => (1, -)
    7: arguments => (2, arguments)
    8: check => (4, check)
    9: leak => (6, leak)
   10: on => (5, on)
   11: to => (3, to)
   12: ==7339==
   13: ==7339== HEAP SUMMARY:
   14: ==7339==     in use at exit: 0 bytes in 0 blocks
   15: ==7339==   total heap usage: 19 allocs, 19 frees, 566 bytes allocated
   16: ==7339==
   17: ==7339== All heap blocks were freed -- no leaks are possible
   18: ==7339==
   19: ==7339== For counts of detected and suppressed errors, rerun with: -v
   20: ==7339== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
```