```
 1: // $Id: makeunique.cpp,v 1.11 2019-05-08 11:33:31-07 - - $
 2:
 3: //
 4: // Example of using a unique_ptr to point at dynamically
 5: // allocated local data.  C++ does not allow local arrays
 6: // to be given a dynamic size, as in "char c[n]".
 7: //
 8:
 9: #include <cstring>
10: #include <iostream>
11: #include <memory>
12: using namespace std;
13:
14: int main() {
15:    const char* const data = "$RCSfile: makeunique.cpp,v $";
16:    const size_t datalen = strlen (data) + 1;
17:    auto buffer = make_unique<char[]> (strlen (data) + 1);
18:    strncpy (buffer.get(), data, datalen);
19:    cout << "buffer=\"" << buffer.get() << "\"" << endl;
20: }
21:
22: //TEST// alias grind='valgrind --leak-check=full --show-reachable=yes'
23: //TEST// grind makeunique >makeunique.out 2>&1
24: //TEST// mkpspdf makeunique.ps makeunique.cpp* makeunique.out
25:
```

```
    1: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting makeunique.cpp
    2: checksource makeunique.cpp
    3: ident makeunique.cpp
    4: makeunique.cpp:
    5:      $Id: makeunique.cpp,v 1.11 2019-05-08 11:33:31-07 - - $
    6:      $RCSfile: makeunique.cpp,v $
    7: cpplint.py.perl makeunique.cpp
    8: Done processing makeunique.cpp
    9: g++ -Wall -Wextra -Wpedantic -Wshadow -fdiagnostics-color=never -std=gnu
++2a -Wold-style-cast -g -O0 makeunique.cpp -o makeunique -lm
   10: rm -f makeunique.o
   11: @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished makeunique.cpp
```

```
    1: ==7595== Memcheck, a memory error detector
    2: ==7595== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
    3: ==7595== Using Valgrind-3.14.0 and LibVEX; rerun with -h for copyright i
nfo
    4: ==7595== Command: makeunique
    5: ==7595==
    6: buffer="$RCSfile: makeunique.cpp,v $"
    7: ==7595==
    8: ==7595== HEAP SUMMARY:
    9: ==7595==     in use at exit: 0 bytes in 0 blocks
   10: ==7595==   total heap usage: 1 allocs, 1 frees, 29 bytes allocated
   11: ==7595==
   12: ==7595== All heap blocks were freed -- no leaks are possible
   13: ==7595==
   14: ==7595== For counts of detected and suppressed errors, rerun with: -v
   15: ==7595== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```