```
// $Id: virtual.cpp,v 1.14 2015-02-10 17:54:20-08 - - $

#include <iostream>
#include <typeinfo>

using namespace std;
#define TRACE(STMT) cout << "TRACE (" << #STMT << ")" << endl; STMT

class nonbase {
   public:
      nonbase() { cout << "nonbase()" << endl; }
      ~nonbase() { cout << "~nonbase()" << endl; }
      void show () { cout << typeid (*this).name() << endl; }
};

class nonderived: public nonbase {
   public:
      nonderived() { cout << "nonderived()" << endl; }
      ~nonderived() { cout << "~nonderived()" << endl; }
      void show () { cout << typeid (*this).name() << endl; }
};

class virtbase {
   public:
      // Inheritance classes must have virtual destructors.
      virtbase() { cout << "virtbase()" << endl; }
      virtual ~virtbase() { cout << "~virtbase()" << endl; }
      virtual void show () { cout << typeid (*this).name() << endl; }
};

class virtderived: public virtbase {
   public:
      virtderived() { cout << "virtderived()" << endl; }
      virtual ~virtderived() { cout << "~virtderived()" << endl; }
      virtual void show () { cout << typeid (*this).name() << endl; }
};

int main () {
   TRACE (nonbase *nb = new nonbase();)
   TRACE (nb->show();)
   TRACE (nonderived *nd = new nonderived ();)
   TRACE (nd->show();)
   TRACE (delete nb;)
   TRACE (nb = nd;)
   TRACE (nb->show();)
   TRACE (nd = static_cast<nonderived*> (nb);)
   TRACE (nd->show();)
   TRACE (delete nd;)

   TRACE (virtbase *vb = new virtbase();)
   TRACE (vb->show();)
   TRACE (virtderived *vd = new virtderived ();)
   TRACE (vd->show();)
   TRACE (delete vb;)
   TRACE (vb = vd;)
   TRACE (vb->show();)
   TRACE (vd = dynamic_cast<virtderived*> (vb);)
   TRACE (vd->show();)
   TRACE (delete vd;)
   TRACE (return 0;)
}

//TEST// grind="valgrind --leak-check=full --show-reachable=yes"
//TEST// $grind virtual 1>virtual.out 2>&1
//TEST// mkpspdf virtual.ps virtual.cpp* virtual.out*

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting virtual.cpp
virtual.cpp:
     $Id: virtual.cpp,v 1.14 2015-02-10 17:54:20-08 - - $
g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++11 virtual.cpp -o virtual -lglut -lGLU -lGL -lX11 -lrt -lm
rm -f virtual.o
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished virtual.cpp
```

```
==8927== Memcheck, a memory error detector
==8927== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==8927== Using Valgrind-3.9.0 and LibVEX; rerun with -h for copyright info
==8927== Command: virtual
==8927==
TRACE (nonbase *nb = new nonbase();)
nonbase()
TRACE (nb->show();)
7nonbase
TRACE (nonderived *nd = new nonderived ();)
nonbase()
nonderived()
TRACE (nd->show();)
10nonderived
TRACE (delete nb;)
~nonbase()
TRACE (nb = nd;)
TRACE (nb->show();)
7nonbase
TRACE (nd = static_cast<nonderived*> (nb);)
TRACE (nd->show();)
10nonderived
TRACE (delete nd;)
~nonderived()
~nonbase()
TRACE (virtbase *vb = new virtbase();)
virtbase()
TRACE (vb->show();)
8virtbase
TRACE (virtderived *vd = new virtderived ();)
virtbase()
virtderived()
TRACE (vd->show();)
11virtderived
TRACE (delete vb;)
~virtbase()
TRACE (vb = vd;)
TRACE (vb->show();)
11virtderived
TRACE (vd = dynamic_cast<virtderived*> (vb);)
TRACE (vd->show();)
11virtderived
TRACE (delete vd;)
~virtderived()
~virtbase()
TRACE (return 0;)
==8927==
==8927== HEAP SUMMARY:
==8927==     in use at exit: 0 bytes in 0 blocks
==8927==   total heap usage: 5 allocs, 5 frees, 27 bytes allocated
==8927==
==8927== All heap blocks were freed -- no leaks are possible
==8927==
==8927== For counts of detected and suppressed errors, rerun with: -v
==8927== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 6 from 6)
```