```
// $Id: inherit.cpp,v 1.6 2016-04-08 19:51:13-07 - - $

// Simple example using inheritance.

#include <cmath>
#include <iostream>
#include <memory>
#include <sstream>
#include <vector>
using namespace std;

////////////////////////////////////////////////////////////////
// class shape
////////////////////////////////////////////////////////////////

class shape {
   private:
      static size_t next_id;
      size_t id {++next_id};
   public:
      virtual double area() const = 0;
      virtual double circum() const = 0;
      virtual ostream& show (ostream&) const;
      friend ostream& operator<< (ostream&, const shape&);
};

size_t shape::next_id {0};

ostream& shape::show (ostream& out) const {
   return out << "shape(" << id << ")";
}

ostream& operator<< (ostream& out, const shape& sh) {
   return sh.show (out);
}

////////////////////////////////////////////////////////////////
// class nothing
////////////////////////////////////////////////////////////////

class nothing: public shape {
   public:
      virtual double area() const override { return 0; }
      virtual double circum() const override { return 0; }
      virtual ostream& show (ostream&) const override;
};

ostream& nothing::show (ostream& out) const {
   return shape::show (out) << ": nothing ";
}

```

```cpp

////////////////////////////////////////////////////////////////////
// class circle
////////////////////////////////////////////////////////////////////

class circle: public shape {
   private:
      double radius {};
   public:
      circle (double r = 0): radius(r) {}
      virtual double area() const override {
         return M_PI * pow (radius, 2);
      }
      virtual double circum() const override {
         return 2 * M_PI * radius;
      }
      virtual ostream& show (ostream&) const override;
};

ostream& circle::show (ostream& out) const {
   return shape::show (out) << ": circle radius " << radius;
}

////////////////////////////////////////////////////////////////////
// class square
////////////////////////////////////////////////////////////////////

class square: public shape {
   private:
      double side {};
   public:
      square (double s): side(s) {}
      virtual double area() const override { return pow (side, 2); }
      virtual double circum() const override { return 4 * side; }
      virtual ostream& show (ostream&) const override;
};


ostream& square::show (ostream& out) const {
   return shape::show (out) << ": square side " << side;
}

```

```


////////////////////////////////////////////////////////////////////
// main function
////////////////////////////////////////////////////////////////////

using shape_ptr = shared_ptr<shape>;
int main (int argc, char** argv) {
   vector<string> args (&argv[1], &argv[argc]);
   vector<shape_ptr> vsp;
   for (const auto& arg: args) {
      cout << arg << ":" << endl;
      istringstream argstream (arg);
      char type {}; double num {};
      argstream >> type >> num;
      shape_ptr obj = nullptr;
      switch (arg[0]) {
         case 'c': obj = make_shared<circle> (num); break;
         case 's': obj = make_shared<square> (num); break;
         default:  obj = make_shared<nothing>(); break;
      }
      vsp.push_back (obj);
   }
   for (const auto& item: vsp) {
      cout << *item << ", area " << item->area() << ", circum "
           << item->circum() << endl;
   }
   return 0;
}

//TEST// alias grind='valgrind --leak-check=full --show-reachable=yes'
//TEST// grind inherit c c10 s s15 x >inherit.out 2>&1
//TEST// mkpspdf inherit.ps inherit.cpp* inherit.out

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting inherit.cpp
inherit.cpp:
     $Id: inherit.cpp,v 1.6 2016-04-08 19:51:13-07 - - $
g++ -g -O0 -Wall -Wextra -rdynamic -std=gnu++14 inherit.cpp
        -o inherit -lglut -lGLU -lGL -lX11 -lrt -lm
rm -f inherit.o
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished inherit.cpp
```

```
==29577== Memcheck, a memory error detector
==29577== Copyright (C) 2002-2013, and GNU GPL'd, by Julian Seward et al.
==29577== Using Valgrind-3.10.1 and LibVEX; rerun with -h for copyright info
==29577== Command: inherit c c10 s s15 x
==29577==
c:
c10:
s:
s15:
x:
shape(1): circle radius 0, area 0, circum 0
shape(2): circle radius 10, area 314.159, circum 62.8319
shape(3): square side 0, area 0, circum 0
shape(4): square side 15, area 225, circum 60
shape(5): nothing , area 0, circum 0
==29577==
==29577== HEAP SUMMARY:
==29577==     in use at exit: 0 bytes in 0 blocks
==29577==   total heap usage: 23 allocs, 23 frees, 870 bytes allocated
==29577==
==29577== All heap blocks were freed -- no leaks are possible
==29577==
==29577== For counts of detected and suppressed errors, rerun with: -v
==29577== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 1 from 1)
```