```
   #include <iostream>
 2 using namespace std;

 4 class Base
   {
 6 public:
     Base(int newvalue);
 8   ~Base();
     virtual int foo(int a);
10   int count;
   private:
12   int value;
   };
14 Base::Base(int newvalue) : value(newvalue)
   {
16   count = 0;
   }
18 Base::~Base()
   {}
20 int Base::foo(int a)
   {
22   count = count+100; return 10*value + a;
   }
24

   class Deriv : public Base
26 {
   public:
28   Deriv(int newvalue);
     virtual int foo(int a);
30 private:
     int value;
32   Base* bp;
   };
34 Deriv::Deriv(int newvalue) : Base(newvalue − 2), value(newvalue), bp(new Base(newvalue − 1))
   {}
36 int Deriv::foo(int a)
   {
38   return 100*value + a;
   }
40
   int footwo(Base& b1, Deriv d2)
42 {
     return b1.foo(1) + d2.foo(2);
44 }

46 int main()
   {
48   Deriv d(9);
     Deriv* dp = new Deriv(5);
50   cout << footwo(d, *dp) << endl;
     cout << d.count << endl;
52   return 0;
   }
```

(handwritten annotations:) d — Deriv; value(9); value(5); Base(7); bp Base(8); Base(3); bp Base(4); Delete dp