

```

1 #include <iostream>
2 using namespace std;
3
4 class Base
5 {
6 public:
7     Base(int newvalue);
8     ~Base();
9     virtual int foo(int a);
10    int count;
11 private:
12    int value;
13 };
14 Base::Base(int newvalue) : value(newvalue)
15 {
16    count = 0;
17 }
18 Base::~Base()
19 {
20    int Base::foo(int a)
21    {
22        count = count+100; return 10*value + a;
23    }
24
25 class Deriv : public Base
26 {
27 public:
28    Deriv(int newvalue);
29    virtual int foo(int a);
30 private:
31    int value;
32    Base* bp;
33 };
34 Deriv::Deriv(int newvalue) : Base(newvalue - 2), value(newvalue), bp(new Base(newvalue - 1))
35 {
36    int Deriv::foo(int a)
37    {
38        return 100*value + a;
39    }
40    int footwo(Base& b1, Deriv d2)
41    {
42        return b1.foo(1) + d2.foo(2);
43    }
44 }
45
46 int main()
47 {
48    Deriv d(9);
49    Deriv* dp = new Deriv(5);
50    cout << footwo(d, *dp) << endl;
51    cout << d.count << endl;
52    return 0;
53 }
```