

```

1  #include <iostream>
2  using std::cout;
3  using std::endl;
4
5  class base
6  {
7  public:
8      int no;
9      int no1;
10
11     base()
12     {
13         cout << "base constructor\n";
14         no = 10;
15         no1 = 100;
16     }
17
18     ~base()
19     {
20         cout << "base destructor\n";
21     }
22 };
23
24 class derived1 : public virtual base
25 {
26 public:
27     int no;
28     int no2;
29
30     derived1()
31     {
32         cout << "derived1 constructor\n";
33         no = 20;
34         no2 = 200;
35     }
36
37     ~derived1()
38     {
39         cout << "derived1 destructor\n";
40     }
41 };
42
43 class derived2 : public virtual base
44 {
45 public:
46     int no;
47     int no3;
48
49     derived2()
50     {
51         cout << "derived2 constructor\n";
52         no = 30;
53         no3 = 300;
54     }
55
56     ~derived2()
57     {
58         cout << "derived2 destructor\n";
59     }
60
61 };
62
63
64 class derived : public derived1, public derived2
65 {
66 public:
67     derived()

```

```

68     {
69         cout << "derived constructor\n";
70     }
71
72     ~derived()
73     {
74         cout << "derived destructor\n";
75     }
76
77     void display()
78     {
79         cout << base::no << endl;
80         cout << derived1::no << endl;
81         cout << derived2::no << endl;
82
83         cout << base::no1 << endl;
84         cout << derived1::no1 << endl;
85         cout << derived2::no1 << endl;
86
87         cout << no2 << endl;
88         cout << no3 << endl;
89     }
90 };
91
92 int main(void)
93 {
94     derived obj;
95
96     cout << endl;
97     cout << sizeof(obj) << endl;
98
99     cout << endl;
100    obj.display();
101    cout << endl;
102
103    return 0;
104 }
105

```