```
#include <iostream>
using std::cout;
using std::endl;

class base
{
public:
    int no;
    int no1;

    base()
    {
        cout << "base constructor\n";
        no = 10;
        no1 = 100;
    }

    ~base()
    {
        cout << "base destructor\n";
    }
};

class derived1 : public virtual base
{
public:
    int no;
    int no2;

    derived1()
    {
        cout << "derived1 constructor\n";
        no = 20;
        no2 = 200;
    }

    ~derived1()
    {
        cout << "derived1 destructor\n";
    }
};

class derived2 : public virtual base
{
public:
    int no;
    int no3;

    derived2()
    {
        cout << "derived2 constructor\n";
        no = 30;
        no3 = 300;
    }

    ~derived2()
    {
        cout << "derived2 destructor\n";
    }

};


class derived : public derived1, public derived2
{
public:
    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
```