

```

1
2 #include <iostream>
3
4 class Base {
5 protected:
6     int data;
7
8 public:
9     Base(int d) : data(d) {}
10
11     // Virtual function to get data
12     virtual int getData() const {
13         return data;
14     }
15
16     // Virtual function to set data
17     virtual void setData(int d) {
18         data = d;
19     }
20
21     // Function to swap data of two objects of the same class type
22     static void swapData(Base* obj1, Base* obj2) {
23         int temp = obj1->getData();
24         obj1->setData(obj2->getData());
25         obj2->setData(temp);
26     }
27 };
28
29 class Derived : public Base {
30 public:
31     Derived(int d) : Base(d) {}
32
33     // Override virtual function to get data
34     int getData() const override {
35         return data;
36     }
37
38     // Override virtual function to set data
39     void setData(int d) override {
40         data = d;
41     }
42 };
43
44 int main() {
45     Base* obj1 = new Derived(10);
46     Base* obj2 = new Derived(20);
47
48     std::cout << "Before swapping:" << std::endl;
49     std::cout << "obj1 data: " << obj1->getData() << std::endl;
50     std::cout << "obj2 data: " << obj2->getData() << std::endl;
51
52     // Swapping data values using the swapData function
53     Base::swapData(obj1, obj2);
54
55     std::cout << "\nAfter swapping:" << std::endl;
56     std::cout << "obj1 data: " << obj1->getData() << std::endl;
57     std::cout << "obj2 data: " << obj2->getData() << std::endl;
58
59     delete obj1;
60     delete obj2;
61
62     return 0;
63 }

```