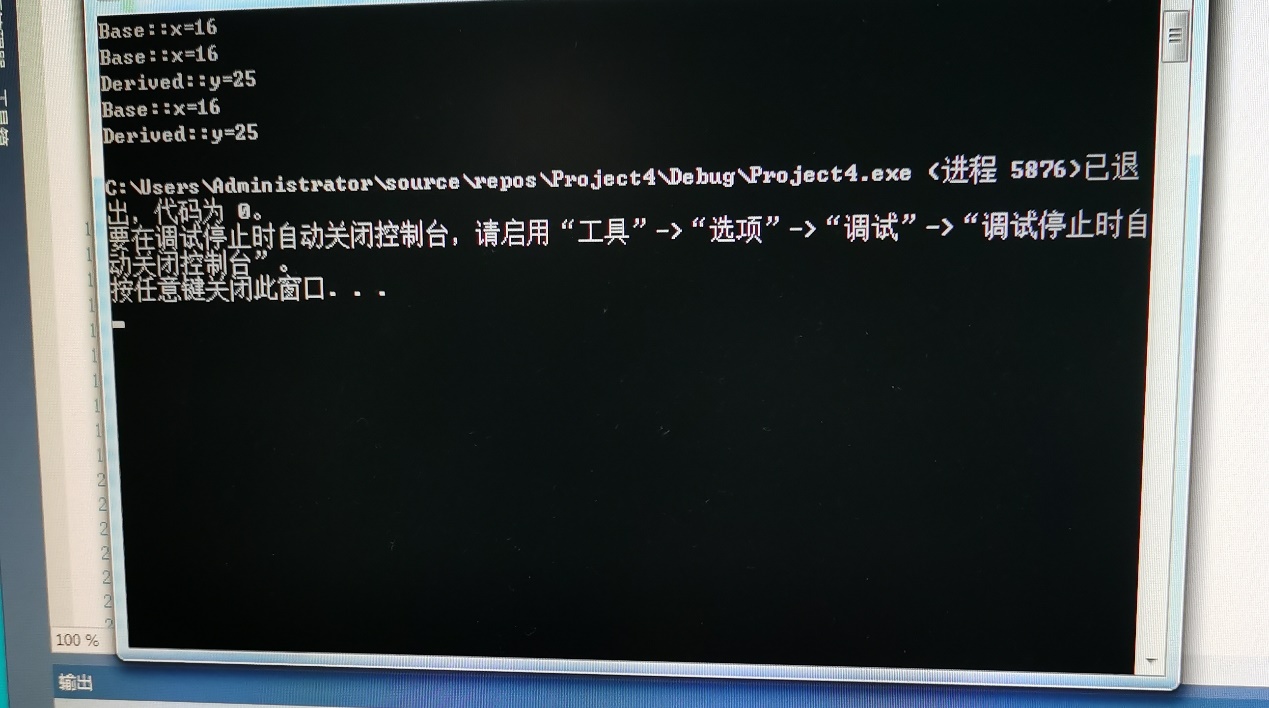
实验报告5

程序代码及结果：

#include <iostream>  
using namespace std;  
class Base {  
public:  
void setx(int i)  
{  
x = i;  
}  
int getx()  
{  
return x;  
}  
public:  
int x;  
};  
class Derived: public Base{  
public:  
void sety(int i)  
{  
y = i;  
}  
int gety()  
{return y; }  
void show()  
{  
cout << "Base::x=" << x << endl;  
}  
public:  
int y;  
};  
int main()  
{  
Derived bb;  
bb.setx(16);  
bb.sety(25);  
bb.show();  
cout << "Base::x=" << bb.x << endl;  
cout << "Derived::y=" << bb.y << endl;  
cout << "Base::x=" << bb.getx() << endl;  
cout << "Derived::y=" << bb.gety() << endl;  
return 0;  
}

改错后的结果



心得体会：

静态成员函数一般不访问普通数据成员，它的作用主要是访问和操作同类中的静态数据成员或全局变量。一个类的静态成员函数与非静态成员函数不同，它不需要创建任何该类的对象就可以被调用。