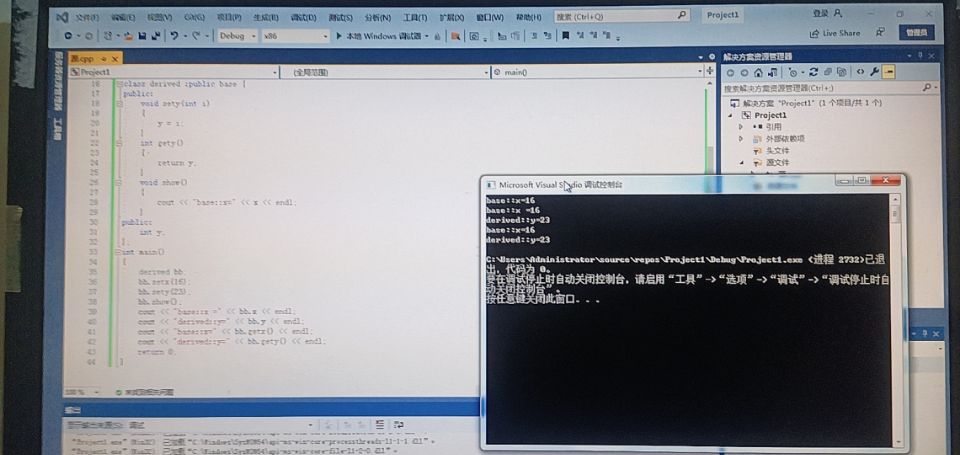
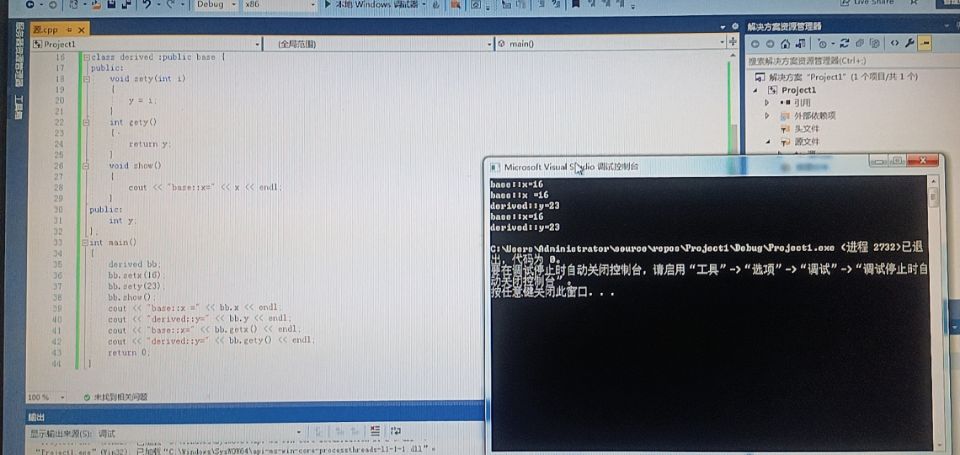
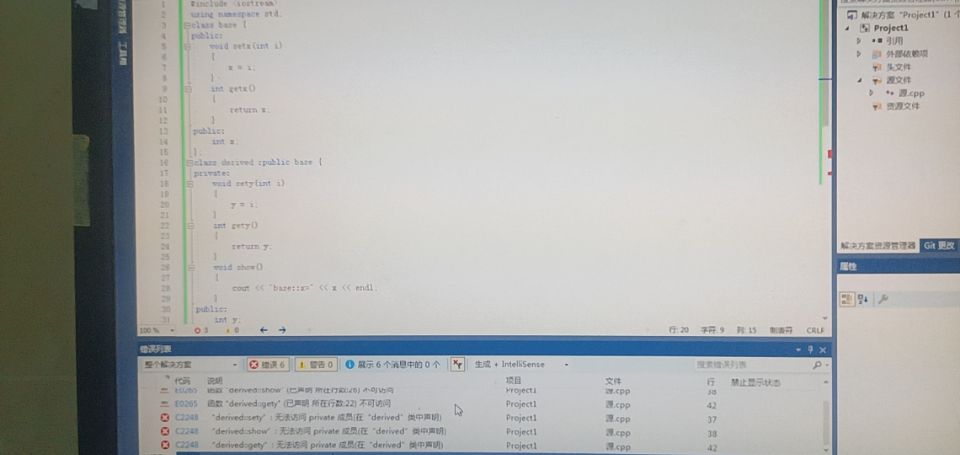
11.19日上机

#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; // 语句1  
 }  
public:  
    int y;  
};  
int main()  
{  
     Derived bb;                             // 语句2  
     bb.setx(16);                            // 语句3  
     bb.sety(25);                            // 语句4  
     bb.show();                              // 语句5  
    cout<<"Base::x="<<bb.x<<endl;       // 语句6  
    cout<<"Derived::y="<<bb.y<<endl;     // 语句7  
    cout<<"Base::x="<<bb.getx()<<endl;    // 语句8  
    cout<<"Derived::y="<<bb.gety()<<endl;   // 语句9  
    return 0;  
}



感想心得：

掌握了派生类的声明方法和派生类构造函数的定义方法。掌握了不同方式下，基类成员在派生类中的访问属性。

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。