#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;

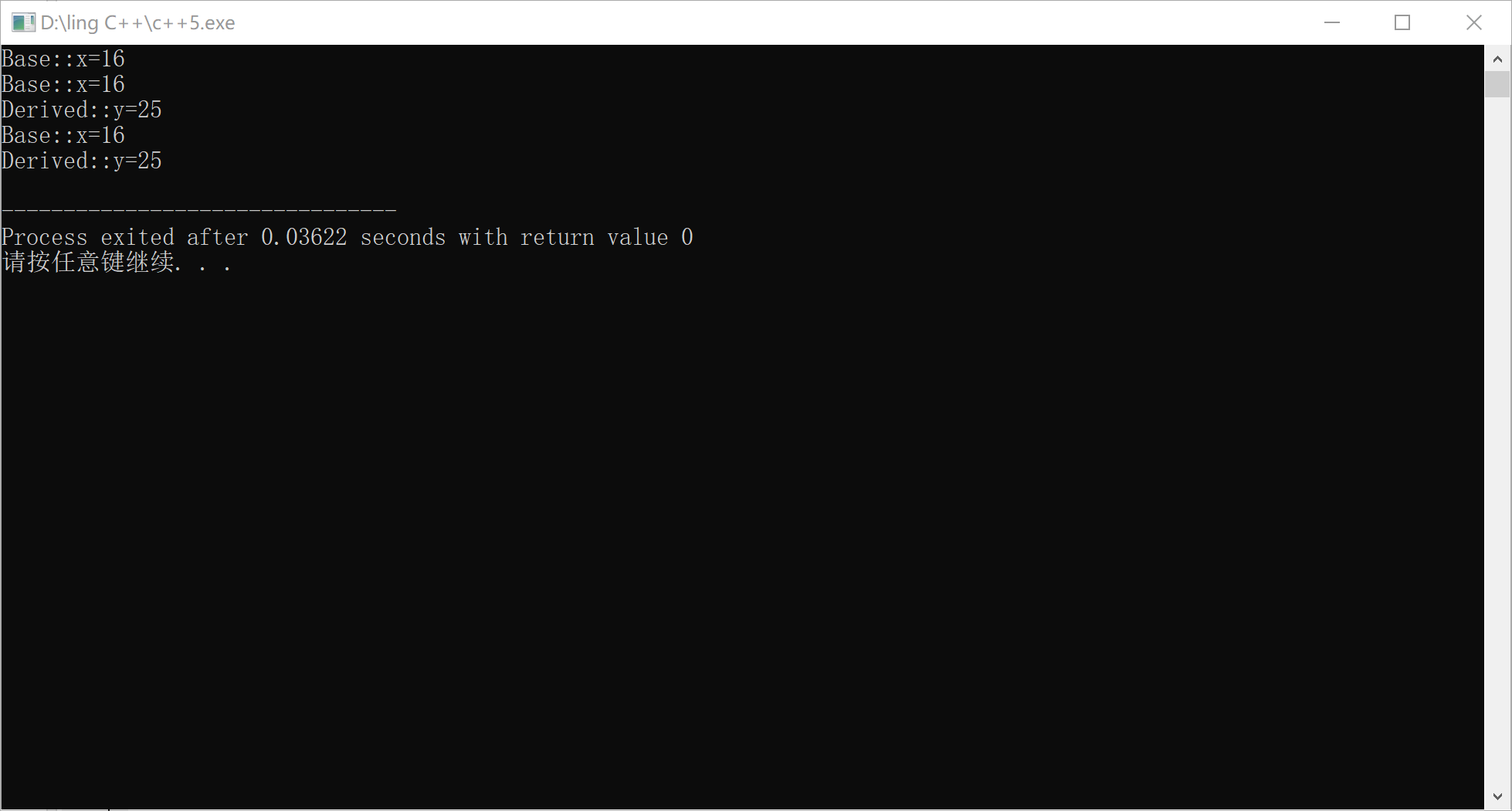
cout<<"Derived::y="<<bb.y<<endl;

cout<<"Base::x="<<bb.getx()<<endl;

cout<<"Derived::y="<<bb.gety()<<endl;

return 0;

}



感想：本次上级内容是有关继承下的构造函数和析构函数，代码量不多，也并不是很复杂，但是本题较为反常的点为平常为private的量改为public，是因为如果为private的话，在类外无法访问，造成程序运行出错，虽然这样可以执行，但在现实中这样的程序有一点的危险。