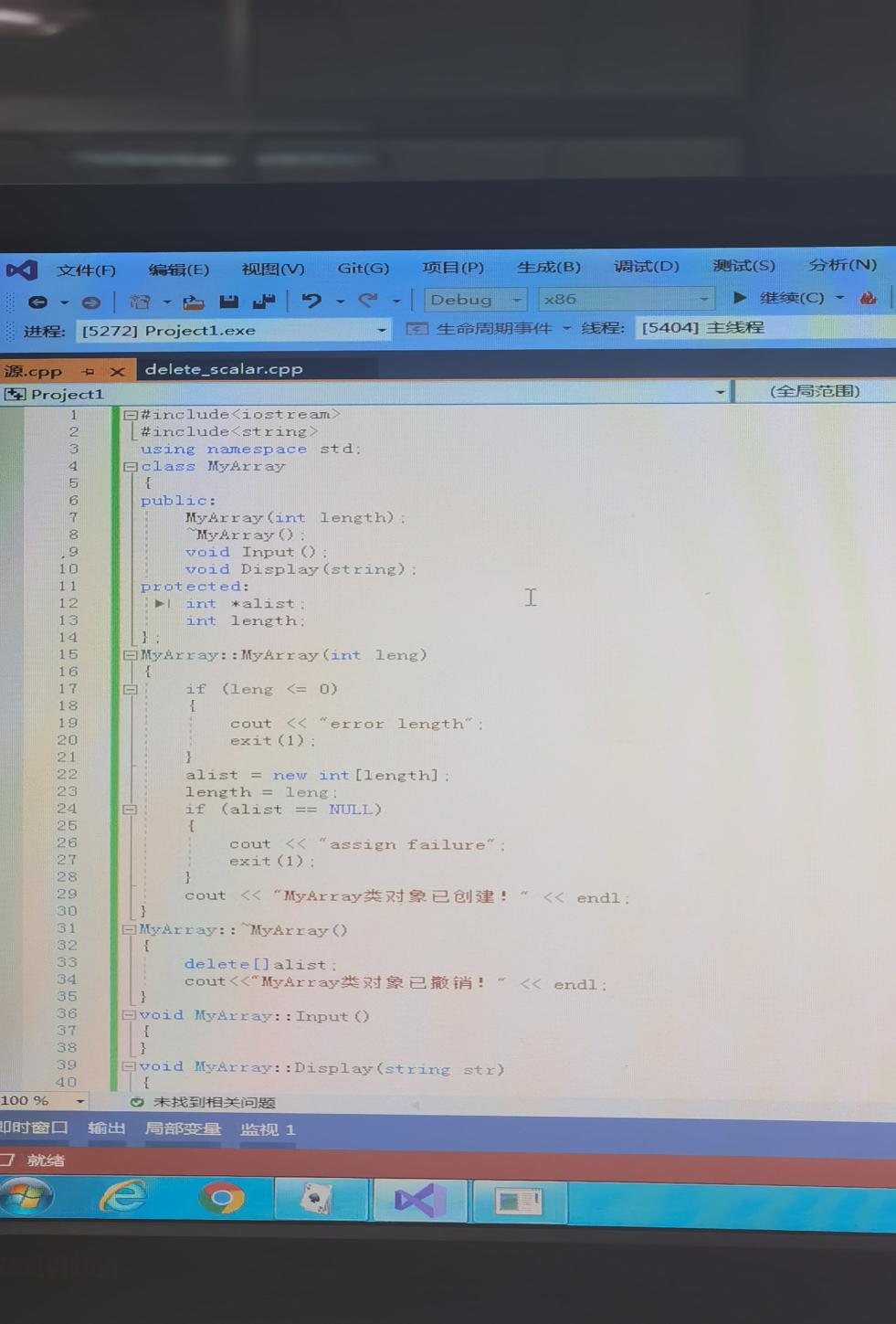
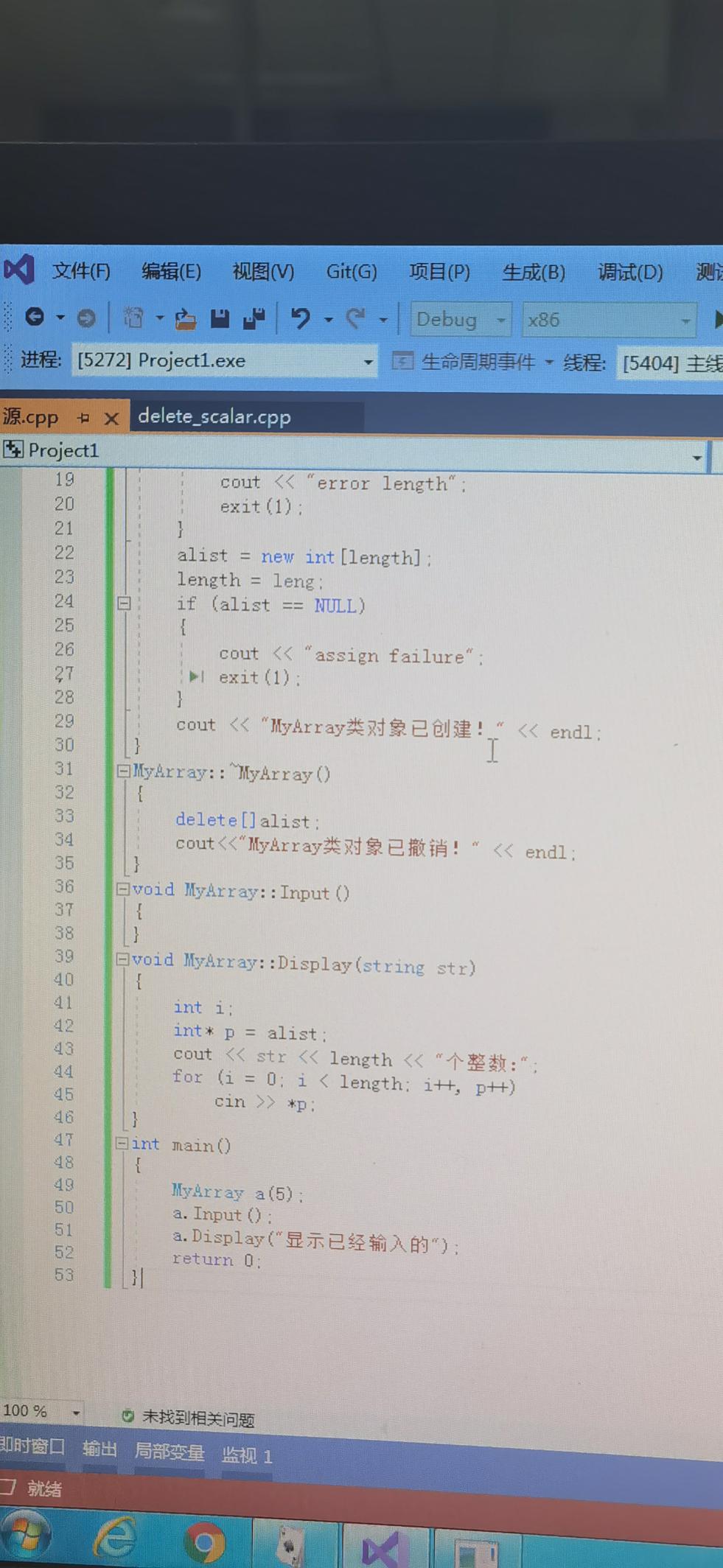


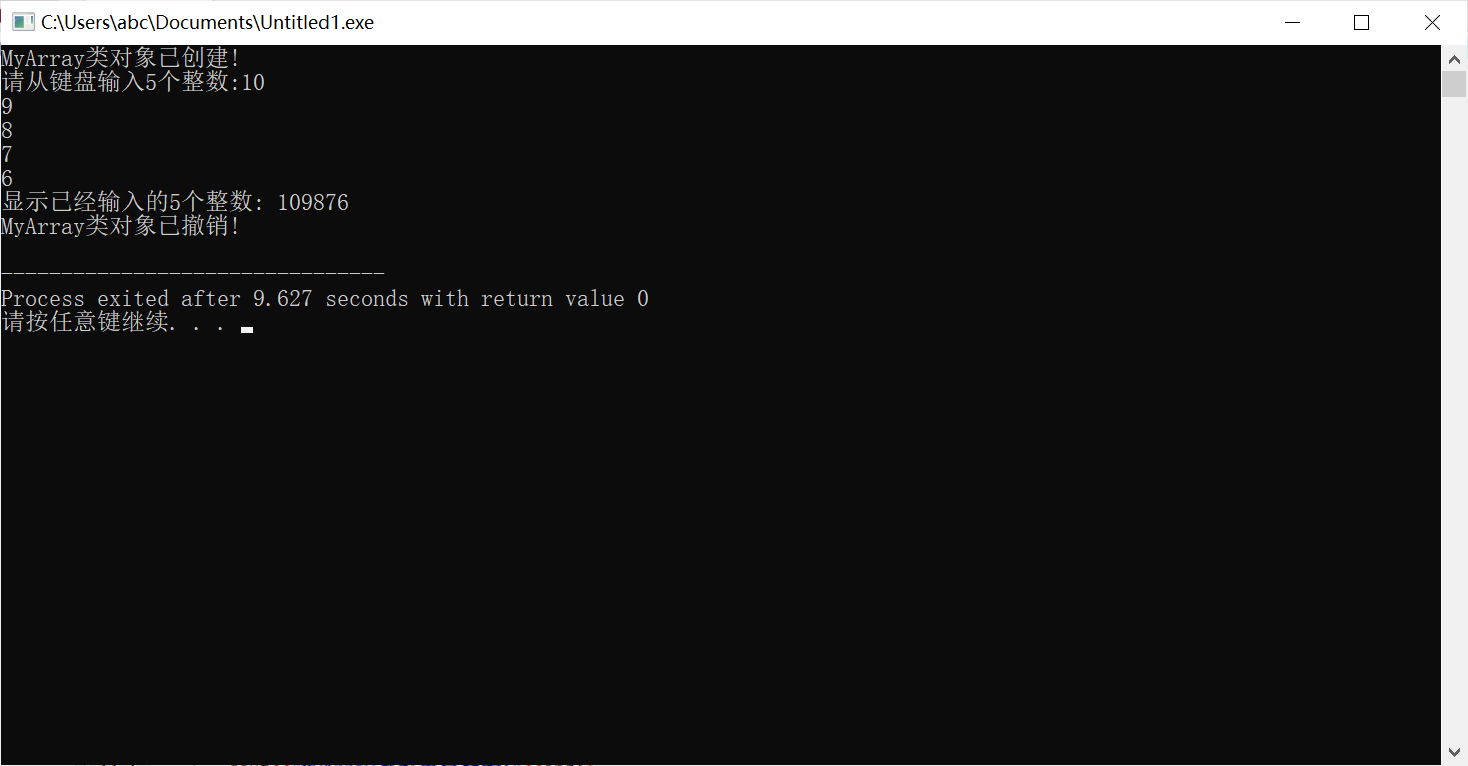
（1）

程序代码：





程序结果：



（2）

程序代码：

#include<iostream>

#include<string>

using namespace std;

class MyArray{

public:

MyArray(int length);

~MyArray();

void Input();

void Display(string);

protected:

int \*alist;

int length;

};

MyArray::MyArray(int leng)

{

if(leng<=0)

{

cout<<"error length";

exit(1);

}

alist = new int [length];

length = leng;

if(alist == NULL)

{

cout<<"assign failure";

exit(1);

}

cout<<"MyArray类对象已创建!"<<endl;

}

MyArray::~MyArray()

{

delete [ ] alist;

cout<<"MyArray类对象已撤销!"<<endl;

}

void MyArray::Display(string str)

{

int i;

int \*p=alist;

cout<<str<<length<<"个整数: ";

for(i=0;i<length;i++,p++)

cout<<\*p<<"";

cout<< endl;

}

void MyArray::Input()

{

cout<<"请从键盘输入"<<length<<"个整数:";

int i;

int \*p=alist;

for(i=0;i<length;i++,p++)

cin>>\*p;

}

class SortArray: public MyArray{

public:

void Sort();

SortArray(int leng):MyArray(leng)

{

cout<<"SortArray类对象已创建!"<<endl;

}

~SortArray();

};

SortArray::~SortArray()

{

cout<<"SortArray类对象已撤销!"<<endl;

}

void SortArray::Sort()

{

int i,j,temp;

for(i=0;i<length-1;i++)

for(j=0;j<length-i-1;j++)

{

if(alist[j]>alist[j+1])

{

temp=alist[j];

alist[j]=alist[j+1];

alist[j+1]=temp;

}

}

}

int main()

{

SortArray s(5);

s.Input();

s.Display("显示排序以前的");

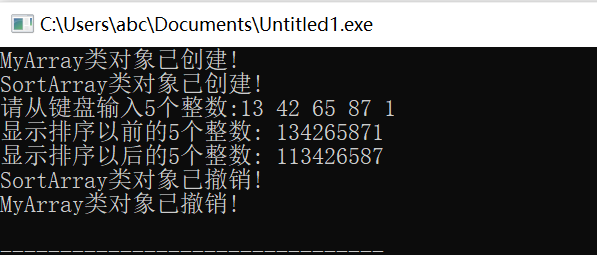
s.Sort();

s.Display("显示排序以后的");

return 0;

}

程序结果：



感想心得：

这节课的操作主要围绕派生类的声明方法和派生类构造函数的定义方法，以及在不同方式下，构造函数与析构函数的执行顺序与构造规则。在定义派生类对象时，构造函数的执行顺序是：先调用基类的构造函数，然后调用内嵌对象成员的构造函数，最后是执行派生类的构造函数体；析构函数与构造函数的执行顺序完全相反，从（2）的运行结果中都能体现出来。通过学习这节课，知识面从基类的构造函数扩展到了派生类的构造函数，完善了自己的知识体系，对于C++知识的掌握逐渐全面起来，通过这次上机课对构造规则和执行顺序的亲身体会，总体下来，收获非常大，但是对于复杂结构体的运行分析仍有很大的欠缺。

Copyright IMG_2562021-2099 HuixinLi. All rights reserved.