代码：

#include<iostream>

#include<string>

#include<cmath>

using namespace std;

class myarray {

protected:

int a[5];

int\* p = a;

public:

myarray();

~myarray();

void setmyarray();

void show();

};

myarray::myarray()

{

cout << "MyArray 类对象已创建！" << endl;

}

myarray::~myarray()

{

cout << "MyArray 类对象已撤销！" << endl;

}

void myarray::setmyarray()

{

int i;

cout << "请从键盘输入5个数：";

for (i = 0; i < 5; i++)

{

cin >> a[i];

}

}

void myarray::show()

{

int q;

cout << "显示排序以前的5个整数：";

for (q = 0; q < 5; q++)

{

cout << a[q] << " ";

}

cout << endl;

}

class sortarray :public myarray {

private:

int b[5];

int\* t = b;

public:

sortarray();

~sortarray();

void sort();

void show();

};

sortarray::sortarray()

{

cout << "SortArray 类对象已创建！" << endl;

}

sortarray::~sortarray()

{

cout << "SortArray 类对象已撤销！" << endl;

}

void sortarray::sort()

{

int n, m, w;

for (n = 0; n < 5; n++, p++)

{

b[n] = \*p;

}

for (n = 0; n < 5; n++)

{

for (w = 0; w < 4 - n; w++)

if (b[w] > b[w + 1])

{

m = b[w];

b[w] = b[w + 1];

b[w + 1] = m;

}

}

}

void sortarray::show()

{

int i;

cout << "显示排序之后的5个整数：";

for (i = 0; i < 5; i++, t++)

cout << \*t << " ";

cout << endl;

}

int main()

{

sortarray aa;

aa.setmyarray();

aa.sort();

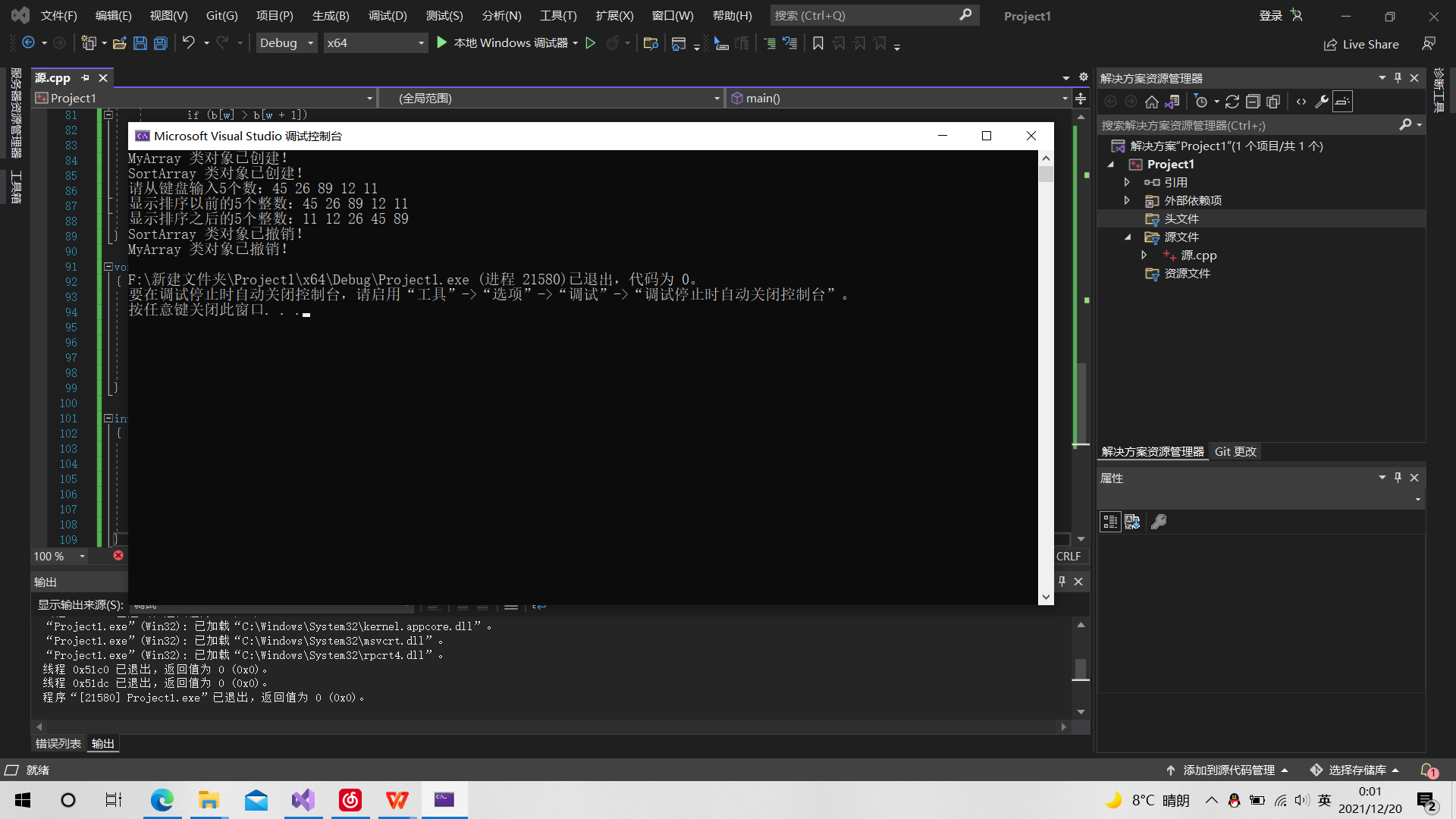
aa.myarray::show();

aa.show();

return 0;

}

结果：



总结：初步掌握派生类的声明方法和派生类构造函数的定义方法，掌握不同方式下，构造函数与析构函数的执行顺序与构造规则，继承方式只会提升访问权限低于该继承方式的成员变量的访问权限。派生类公有继承基类为优解，并且基类中的指针变量最好声明为保护类型，以便基类进行访问。