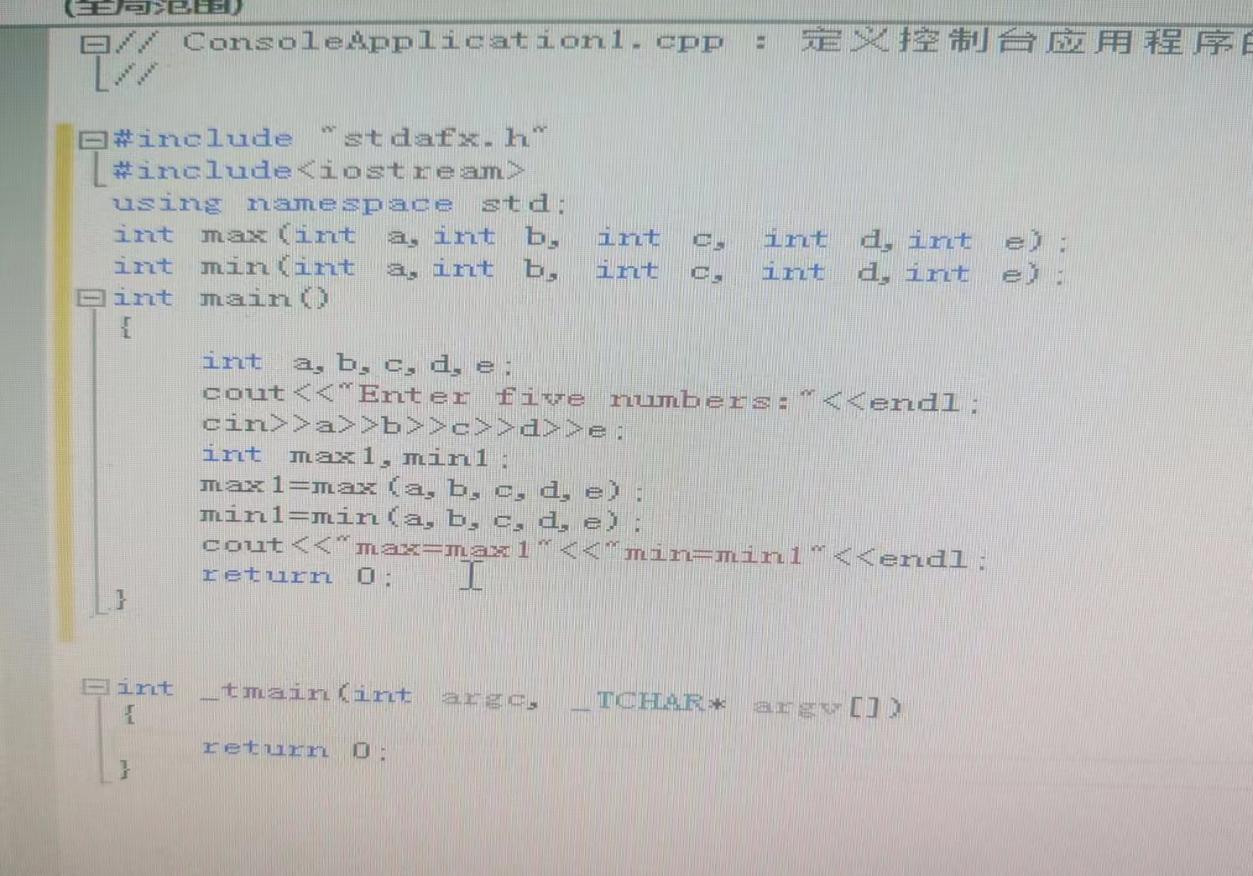
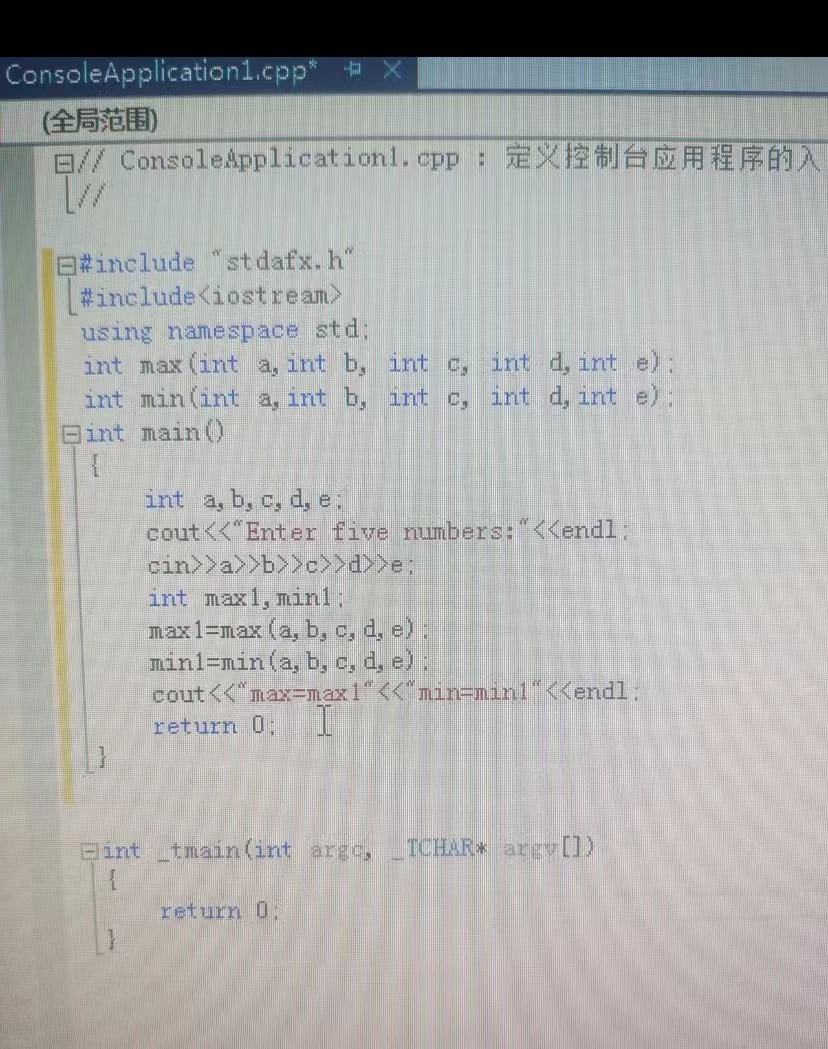
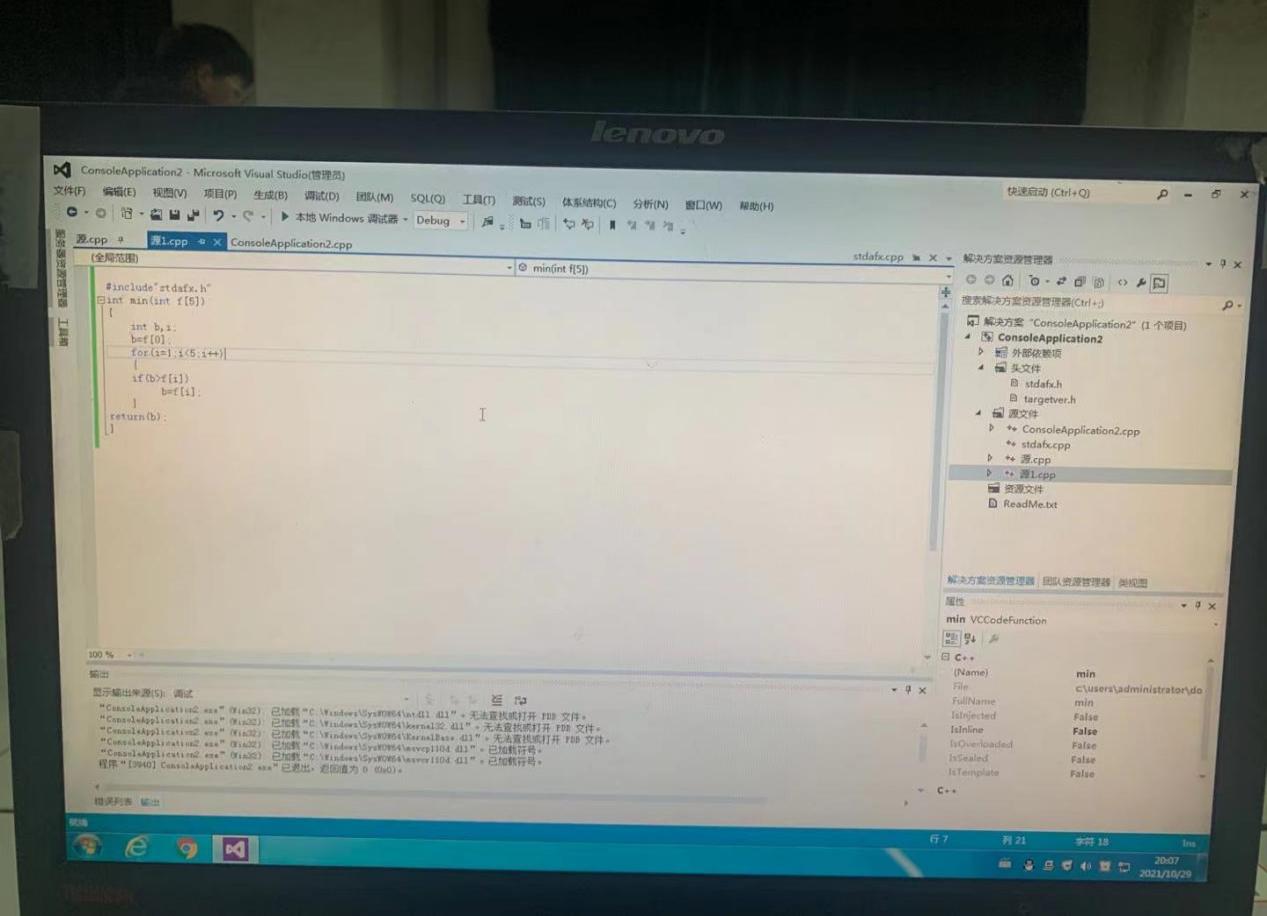
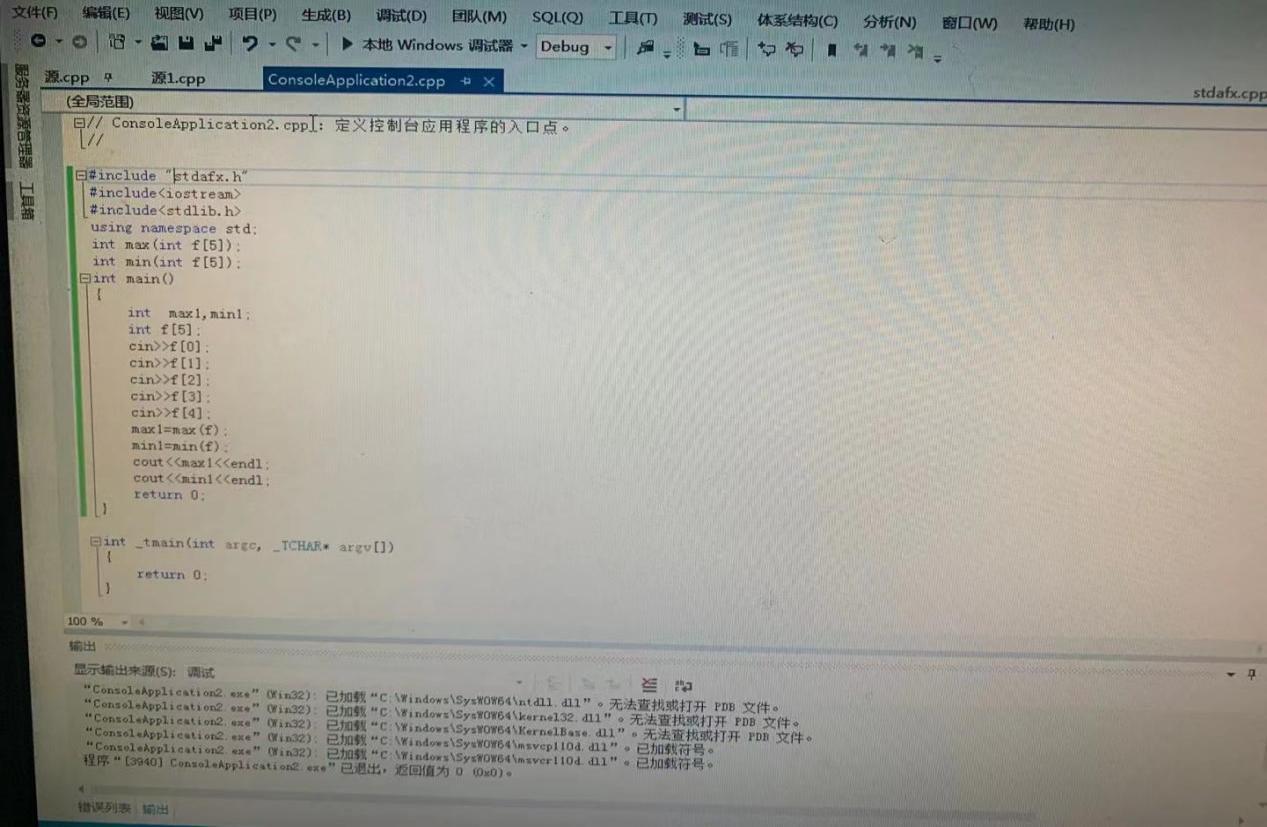
[[1]](#footnote-0)第一次上机（十月二十二日）

心得：第一次上机是对输入cout和输出cin以及一些基本语句的应用（注意endl结尾是必不可缺少的）。<<的运用也十分关键，第一次实验主要还是运用的和c语言相似的结构，用const代替了#define以此消除#define的危险性。

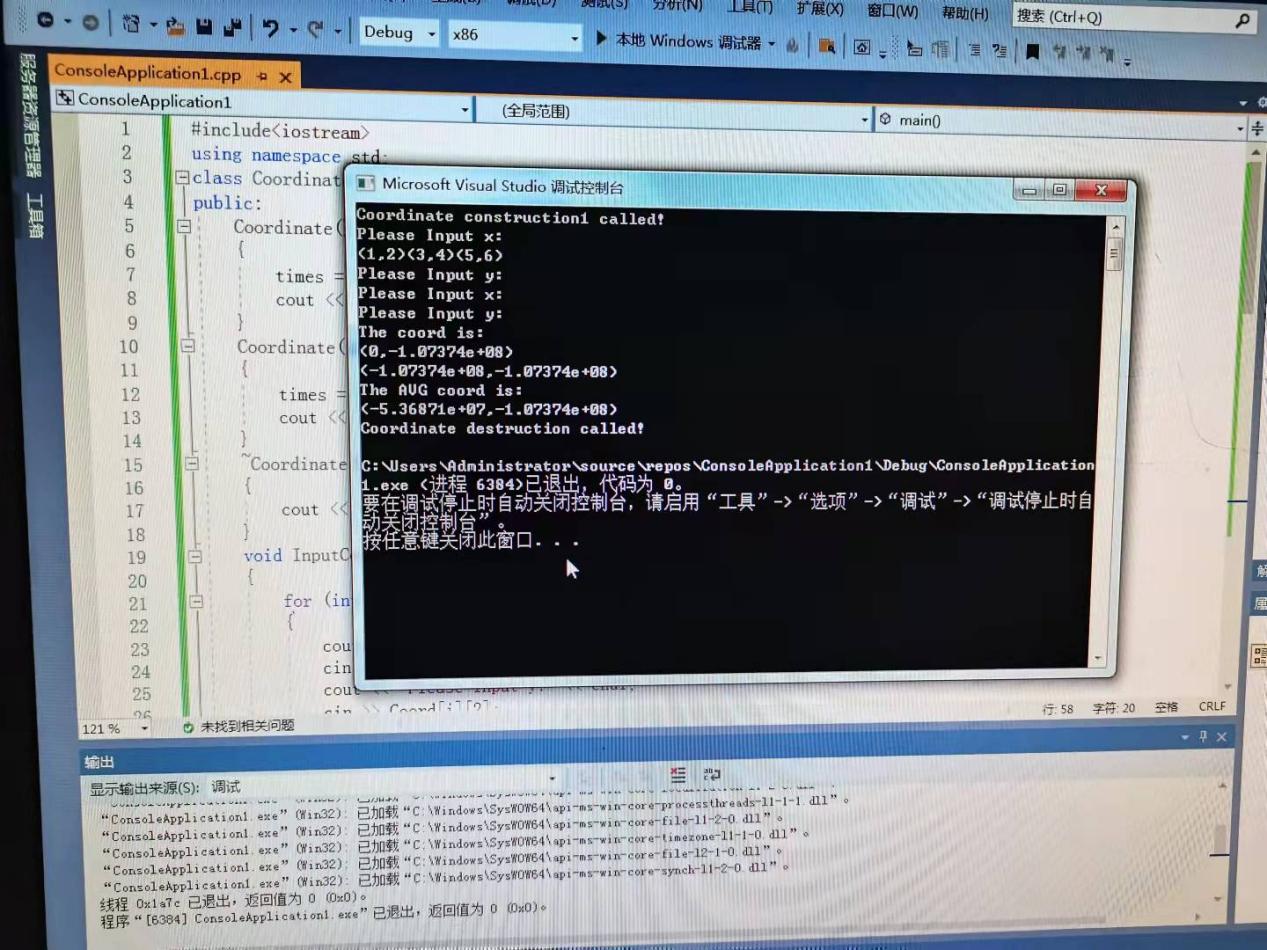
第二次上机（十月二十九日）





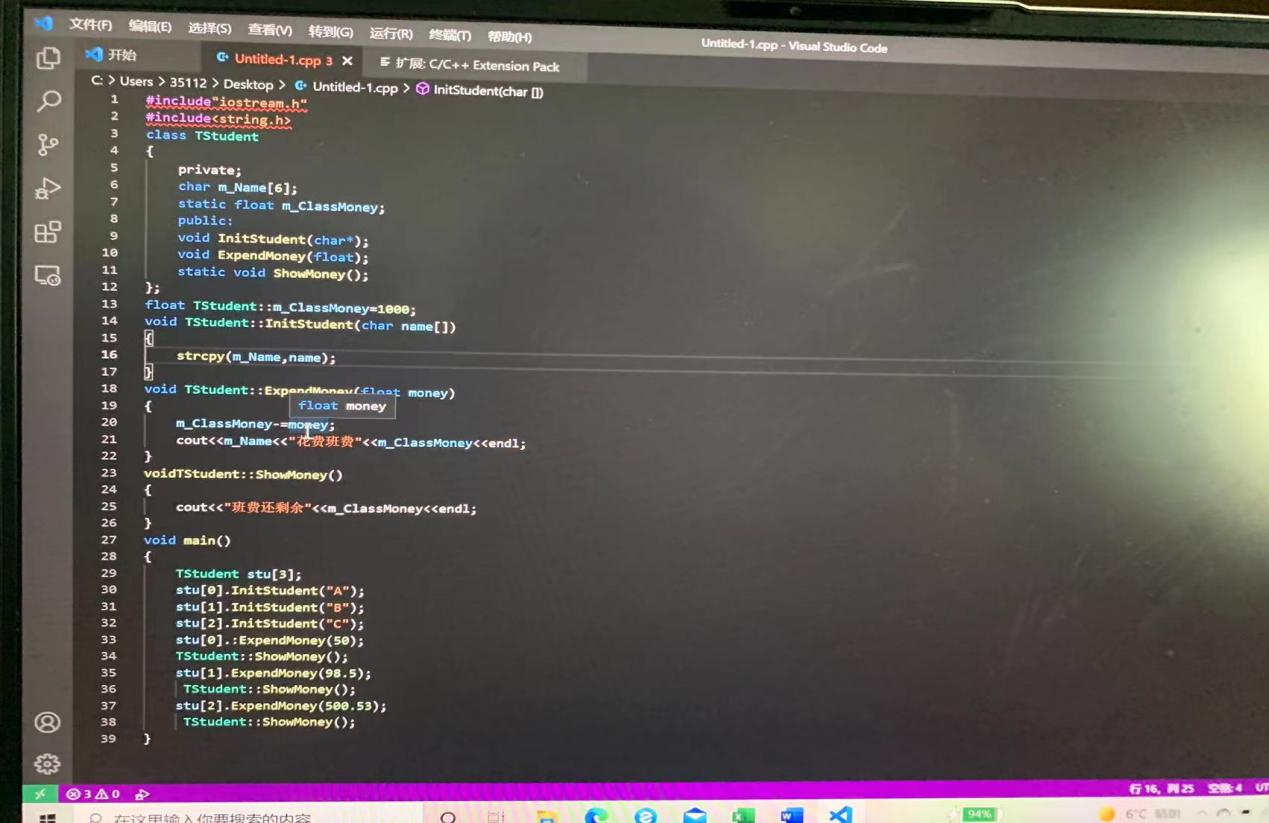
心得：第二次上机初步接触到建立和运行多文件程序，在我看来这次实验的方法和运行程序和c语言的函数部分基本相似，运用到了一维数组，其中输入和输出改变，开头的输入语也是数组selNumber[ ]；为保证程序正常运行，max变量没有被赋初值；需要在main函数前对子程序里的函数进行声明。[[2]](#footnote-1)

第三次上机（十一月五日）



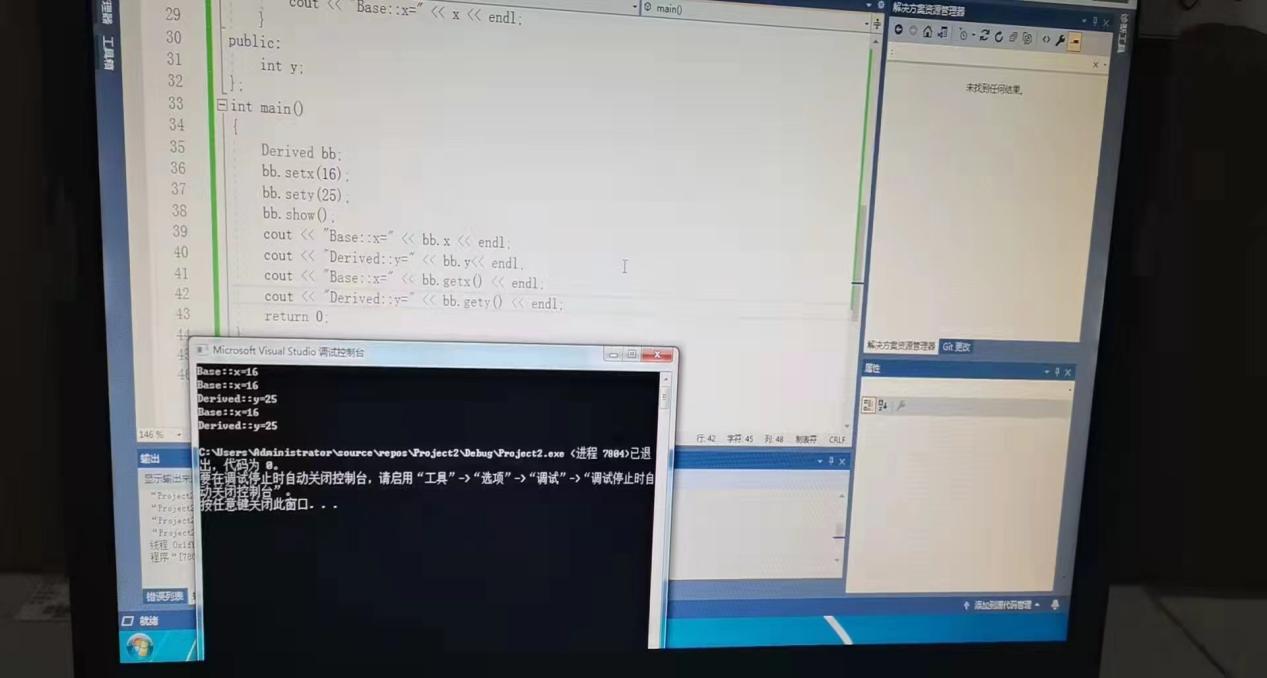
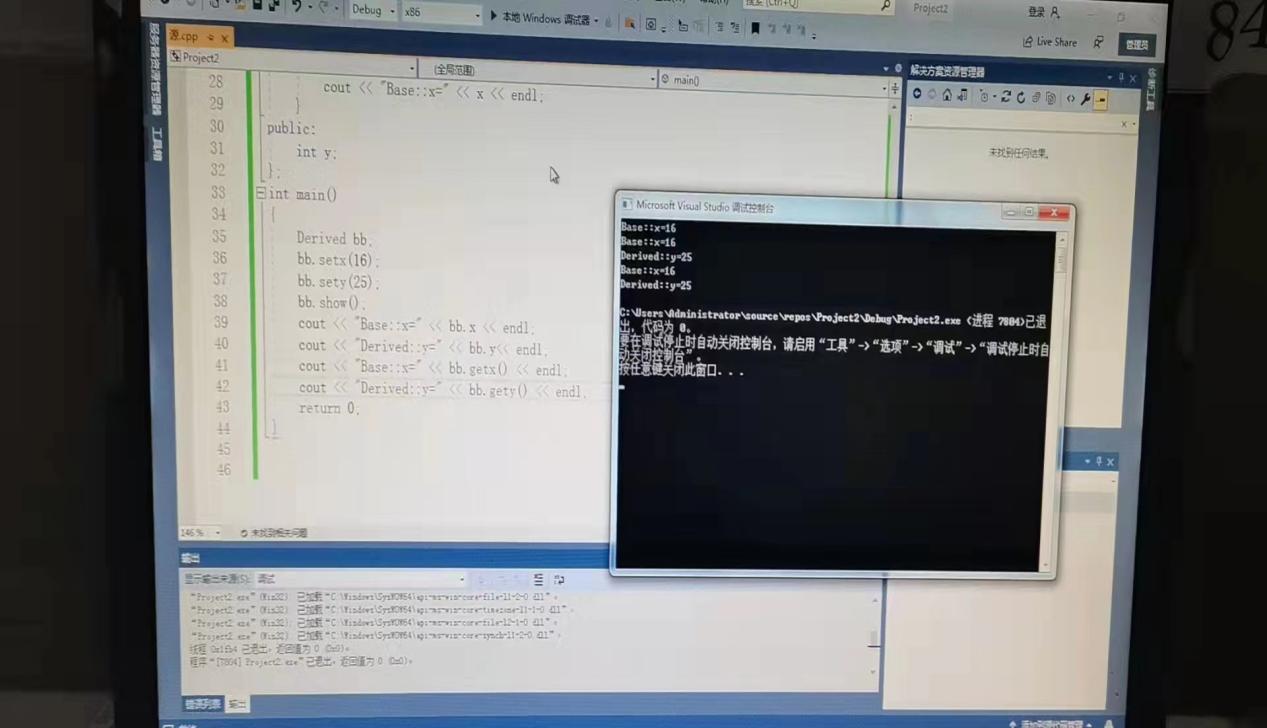
心得：实验中，创建一个Score类，连续输入一个二维坐标值，二位坐标值的数目可以由用户定义（默认为２个，最多为100组），显示用户输入的坐标值，显示用户输入坐标值的均值，最后对整个文件进行打包。第三次试验初步运用了析构函数和构造函数。初步掌握使用类和对象编制Ｃ＋＋程序。

第四次上机（十一月十二日）



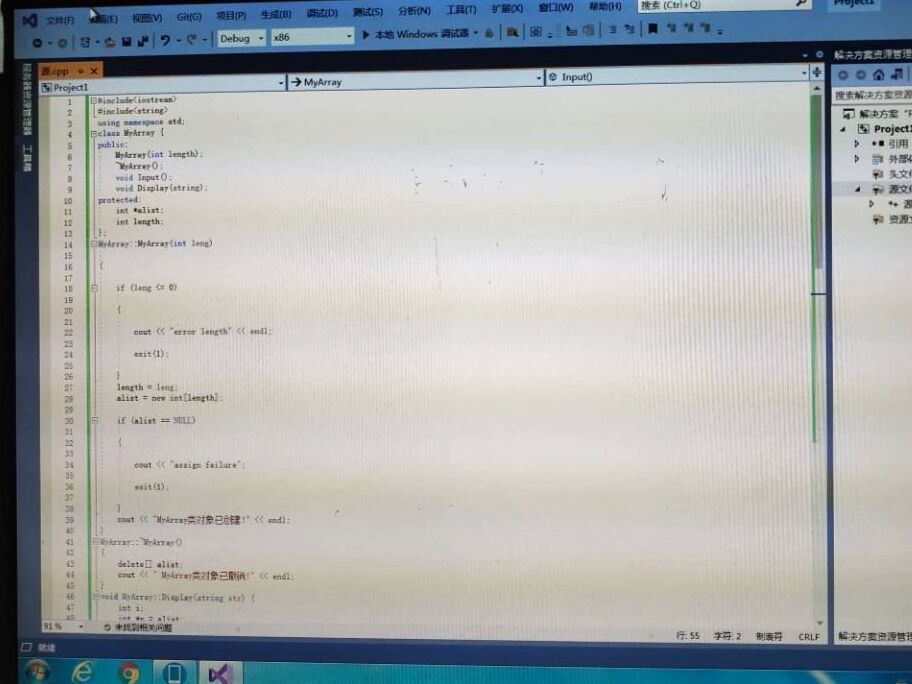
心得：第四次实验创建了一个TStudent类，该类包括三个函数：initStudent、ExpendMoney和Show Money；采用float m\_ClassMone变量作为静态变量，用于储存班级的班费，初始值设为1000；initStudent（char name[]）主要负责完成学生姓名的初始化；ExpendMoney（float money）主要是负责完成班费的花销计算；Show Money（）主要完成班费余额显示。

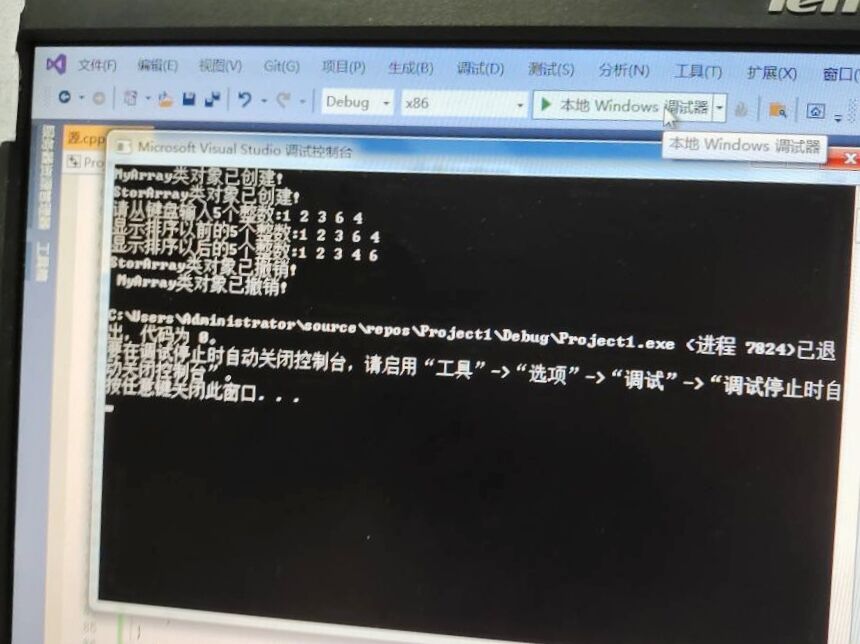
第五次上机（十一月十九日）



心得：１）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。２）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。

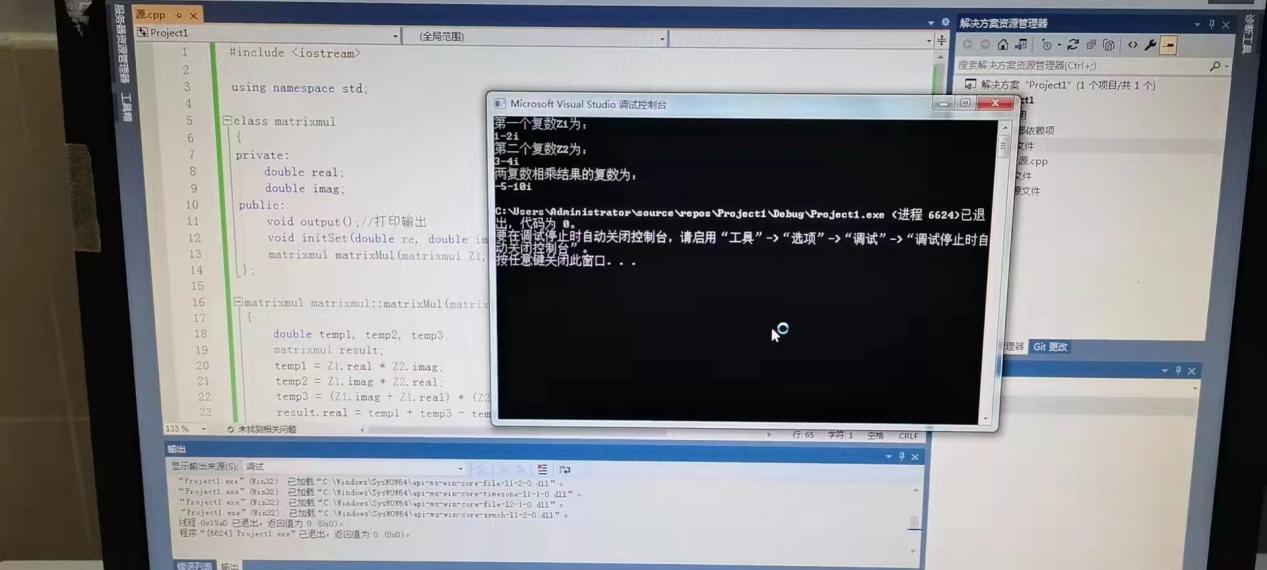
第六次上机（十一月二十六日）





心得；声明一个sottarray继承类myarray，在该类定义一个函数，具有奖输入的整数从小到大进行排序的功能。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，构造函数与析构函数的执行顺序与构造规则。

第七次上机（十二月三日）



心得：１）运算符重载的函数格式：函数类型 operator 运算符名称（形参表）{对运算符的重载处理}；２）对双目运算符而言，成员运算符重载函数的形参表中仅有一个参数，它作为运算符的右操作数；另一个操作数（左操作数）是隐含的，是该类的当前对象，他是通过this指针隐含传递给函数。

#include<iostream>  
using namespace std;  
class Complex {  
public:  
double real, img;  
Complex(double r = 0 ,double i = 0)  
{  
real = r;  
img = i;  
}  
};  
Complex operator\*(Complex a, Complex b)  
{  
Complex temp;  
temp.real = a.real \* b.real-a.img\*b.img;  
temp.img = a.real \* b.img + a.img \* b.real;  
return temp;  
}  
int main()  
{  
double a,b,c,d;  
cout << "请输入两个复数的实部和虚部" << endl;  
cin >> a >> b >> c >> d;  
Complex A(a, b);  
Complex B(c, d);  
Complex temp;  
temp = A \* B;  
cout << "乘积的实部是：" << temp.real << endl;  
cout << "乘积的虚部是：" << temp.img << endl;  
return 0;  
}

1. Copyright ©2021-2099 XuliangLV. All rights reserved [↑](#footnote-ref-0)
2. Copyright ©2021-2099 XuliangLV. All rights reserved [↑](#footnote-ref-1)