**上机作业4！对象传递与静态成员**

# include<iostream>

 using namespace std;

class Tr{  
Tr( int n)  
{

  i=n;

}

void set\_i( int n)  
{ i=n:}

int get\_i()

{return 0；}  
private:  
 int i;

}；  
  
void sqr\_ it( Tr ob)

{

ob. set\_i(ob.get\_i()\*ob.get\_i());

cout< < " 在函数sqr\_ it内，形参对象ob的数据成员i的值为：“ <<ob.get\_i;

 cout<<endl;

}

int main()

{ Tr obj(10);

cout<<”调用函数sqr\_it前，实参对象obj的数据成员i的值为:”；

cout<<obj.get\_i()<<endl;

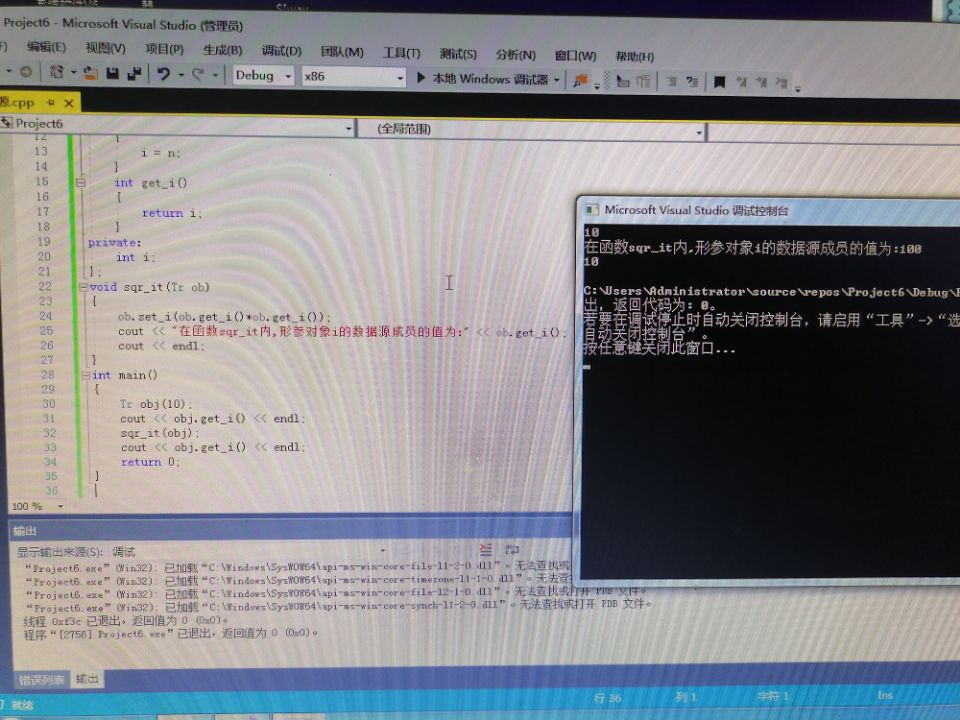
sqr\_it(obj);

cout<<”调用函数sqr\_it后，实参对象obj的数据成员i的值为:”;

cout<<ob.get\_i()<<endl;

return 0;

}

****

# include<iostream>

using namespace std;

class Ctest{

static int count;

public:

Ctest()

{

++count;

Cout<<””<<count<<endl;

}

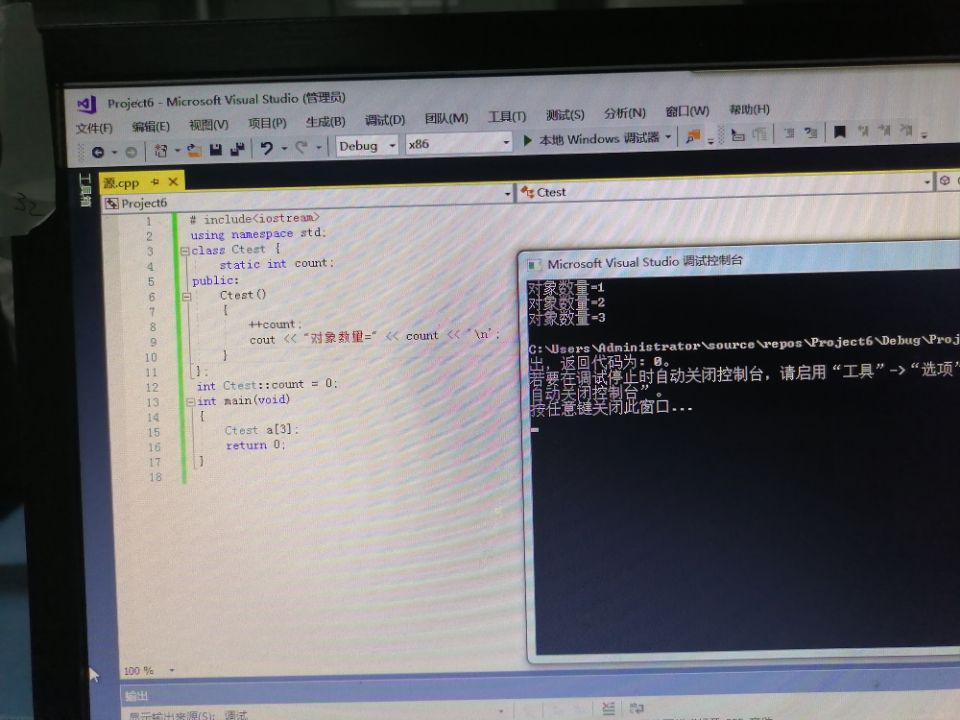
};

int Ctest::count=0;

int main()  
{  
 Ctest a[3];

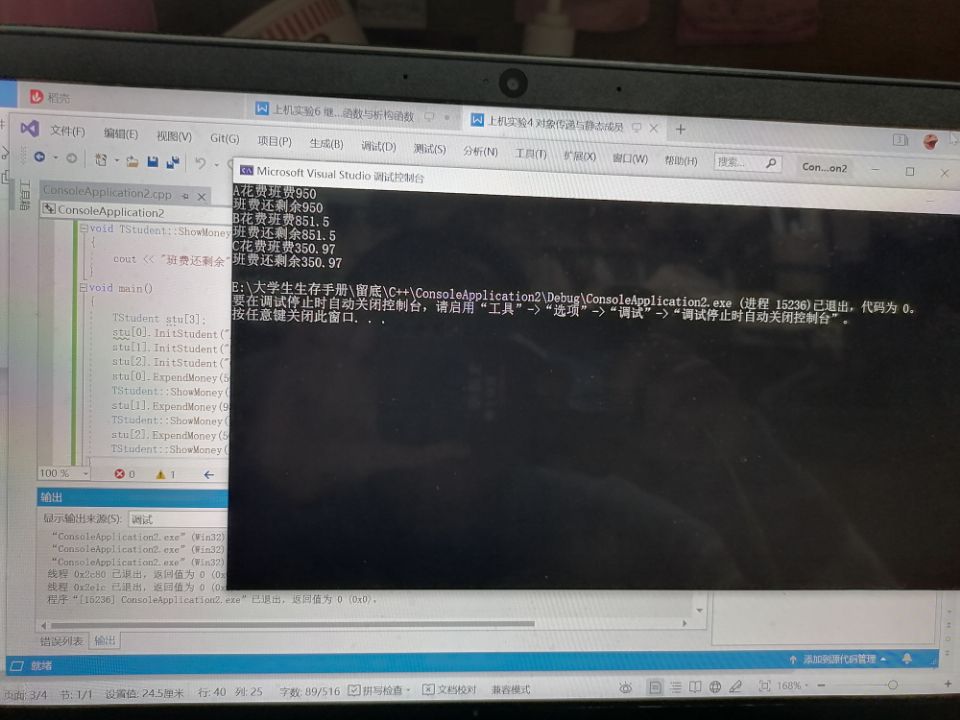
return 0;

}

****

**练习：**

#include <iostream>  
#include <string.h>  
using namespace std;  
class TStudent  
{  
private:  
char m\_Name[6];  
static float m\_ClassMoney;  
public:  
void InitStudent(const char name[6]);  
void ExpendMoney(float);  
static void ShowMoney();  
};  
float TStudent::m\_ClassMoney = 1000;  
void TStudent::InitStudent(const char name[6])  
{  
strcpy\_s(m\_Name,name);  
}  
void TStudent::ExpendMoney(float money)  
{  
m\_ClassMoney-=money;  
cout << m\_Name << "花费班费" << m\_ClassMoney << endl;  
}  
void TStudent::ShowMoney()  
{  
cout << "班费还剩余" << m\_ClassMoney << endl;  
}  
void main()  
{  
TStudent stu[3];  
stu[0].InitStudent("A");  
stu[1].InitStudent("B");  
stu[2].InitStudent("C");  
stu[0].ExpendMoney(50);  
TStudent::ShowMoney();  
stu[1].ExpendMoney(98.5);  
TStudent::ShowMoney();  
stu[2].ExpendMoney(500.53);  
TStudent::ShowMoney();  
}

****

**感想：**

**本次上机实验，我观察到了对象传递的方向，同时认识到了静态成员，它实现了多个对象之间的数据共享，但是我并没意识到程序出错的地方是变量未定义就已使用，同时注意型参与实参之间的传递和const 的使用。**