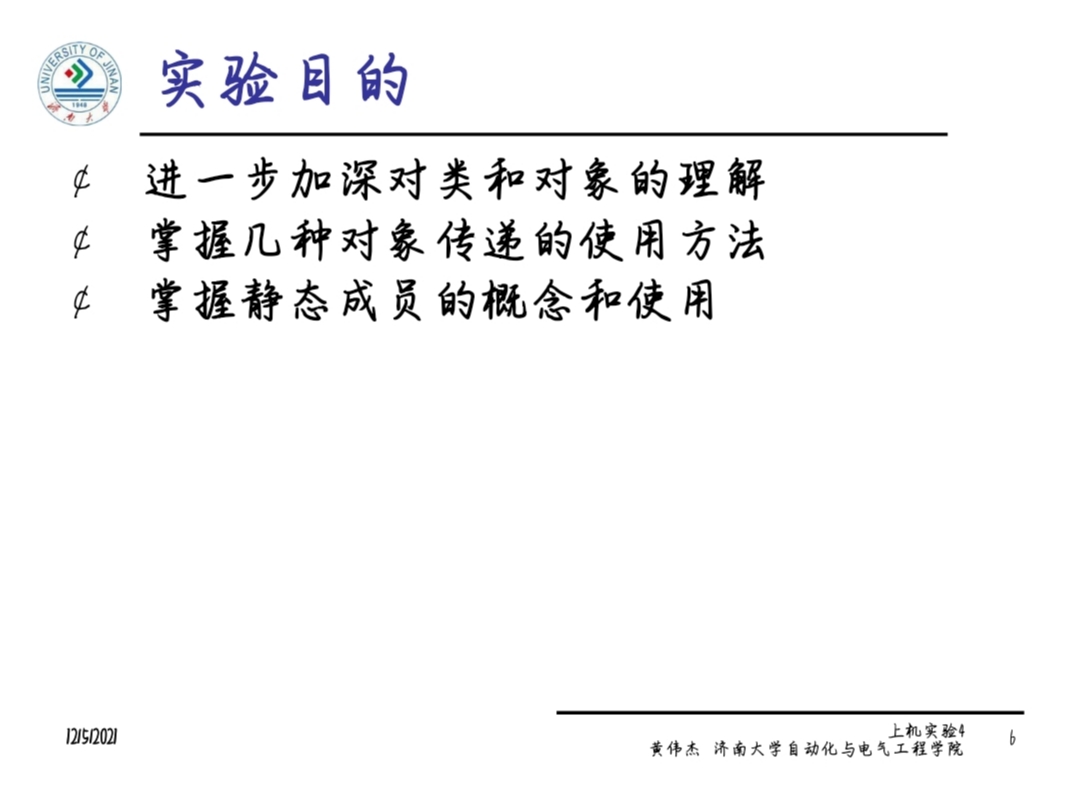
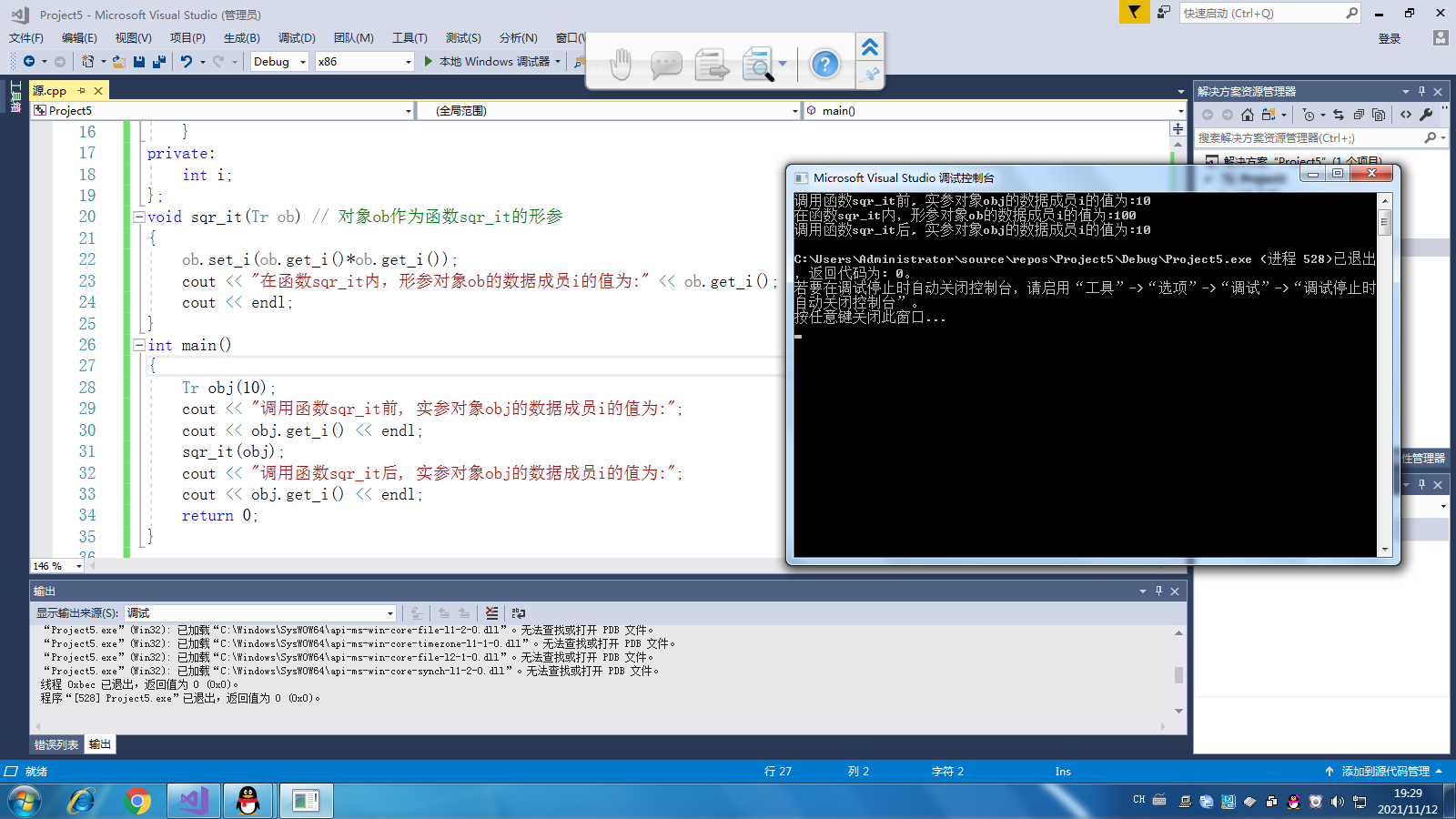
**实验四 11.12**

****

1. **操作**

****

# include<iostream>

using namespace std;

class Tr {

public:

Tr(int n)

{

i = n;

}

void set\_i(int n)

{

i = n;

}

int get\_i()

{

return i;

}

private:

int i;

};

void sqr\_it(Tr ob) // 对象ob作为函数sqr\_it的形参

{

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;

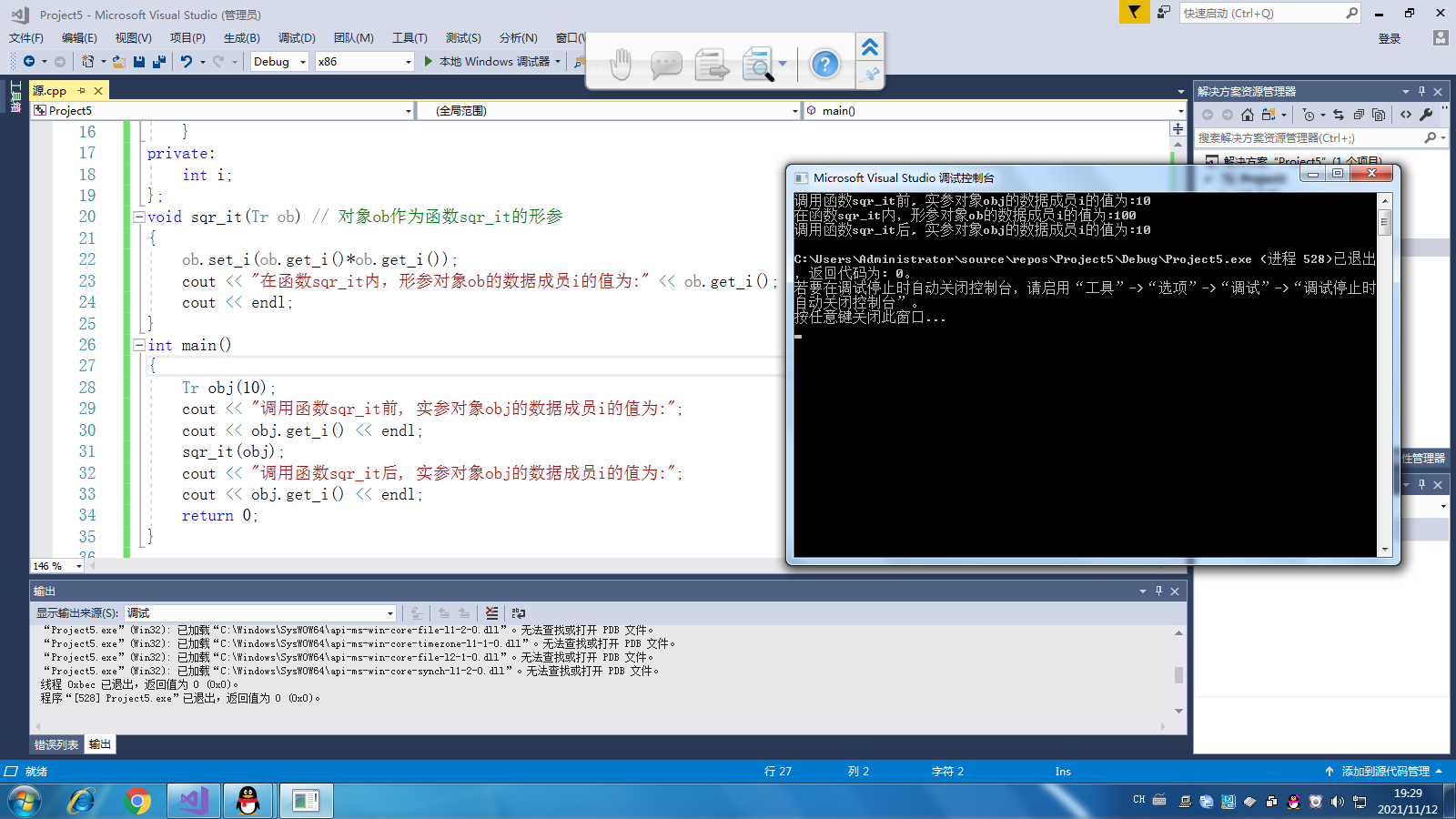
sqr\_it(obj);

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

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

return 0;

}

****

#include<iostream>

using namespace std;

class Ctest

{

static int count; //私有成员

public:

Ctest()

{

++count; cout << "对象数量=" << count << endl;

}

};

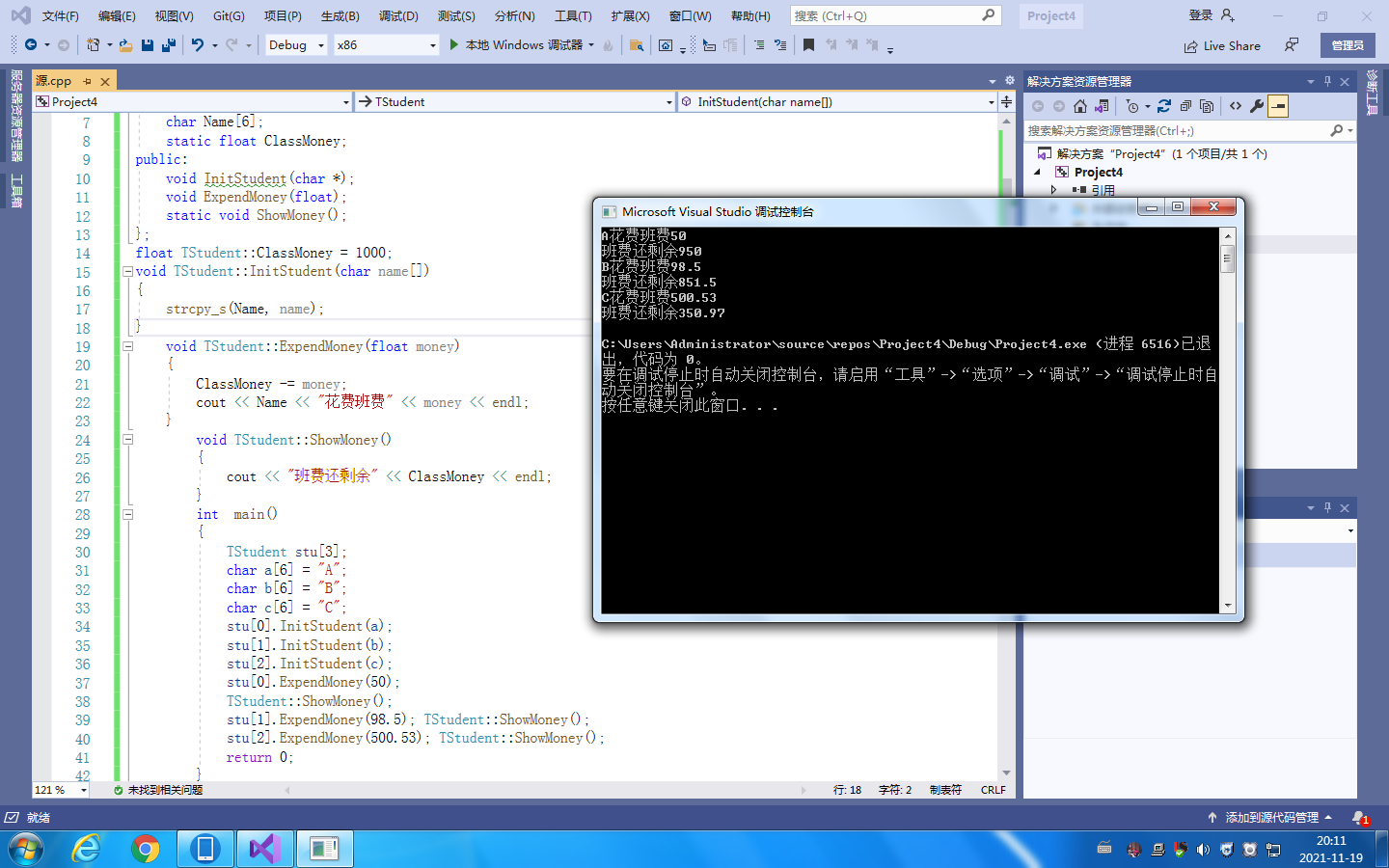
int Ctest:: count = 0;//A，对静态数据定义性说明

int main(void){

Ctest a[3];

return 0;

}



#include <iostream>

#include <string.h>

using namespace std;

class TStudent

{

private:

char Name[6];

static float ClassMoney;

public:

void InitStudent(char \*);

void ExpendMoney(float);

static void ShowMoney();

};

float TStudent::ClassMoney = 1000;

void TStudent::InitStudent(char name[])

{

strcpy\_s(Name, name);

}

void TStudent::ExpendMoney(float money)

{

ClassMoney -= money;

cout << Name << "花费班费" << money << endl;

}

void TStudent::ShowMoney()

{

cout << "班费还剩余" << ClassMoney << endl;

}

int main()

{

TStudent stu[3];

char a[6] = "A";

char b[6] = "B";

char c[6] = "C";

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();

return 0;

}

1. **问题**
2. 在C++中主函数用void main出现报错，让修改成int main。
3. Strcpy出现报错说，不安全，要改成strcpy\_s，比较安全。
4. stu[2].ExpendMoney(500.53);出现警告，从double到float截断。
5. 静态数据成员在定义或者说明时前面添加关键字static；
6. 初始化时使用作用域运算符来表明它所属类，因此，静态数据成员是类的成员，而不是对象的成员。
7. 初始化在类体外进行，而前面不加static，以免与一般静态变量或者对象相混肴。
8. **感受心得**

C++中有C语言没有的对象权限的问题，C++中数据成员如果是私有成员，那么类外对象就不能访问这个成员。

如果用对象作为形参，实际上就是值传递，在子函数内数值会改变，但是在子函数外，数值不会改变，仍是原来的数值。但是如果传递对象的地址，或者是形参是对象的引用，这是改变的是地址，数值就会发生改变。

对于静态成员，C++和C语言相类似，但是C++中静态成员属于类，不属于对象，所以直接用类进行调用。

Copyright©2021-2099 MengyuanWang.All rights reserved.