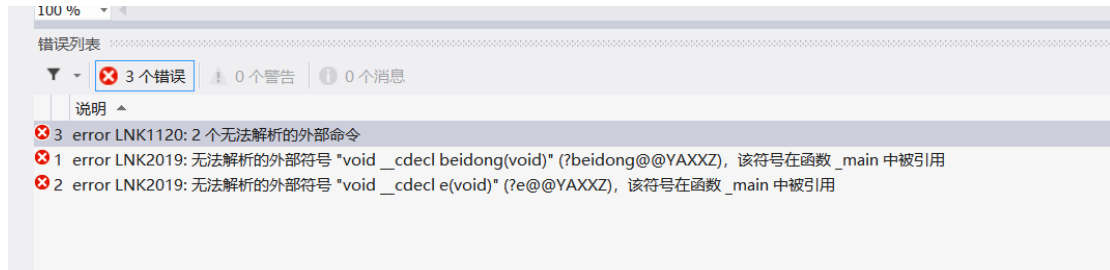


错误整理

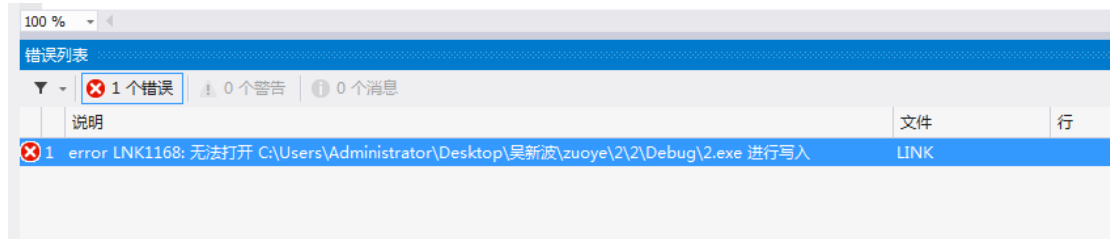
1

产生的错误及解决办法

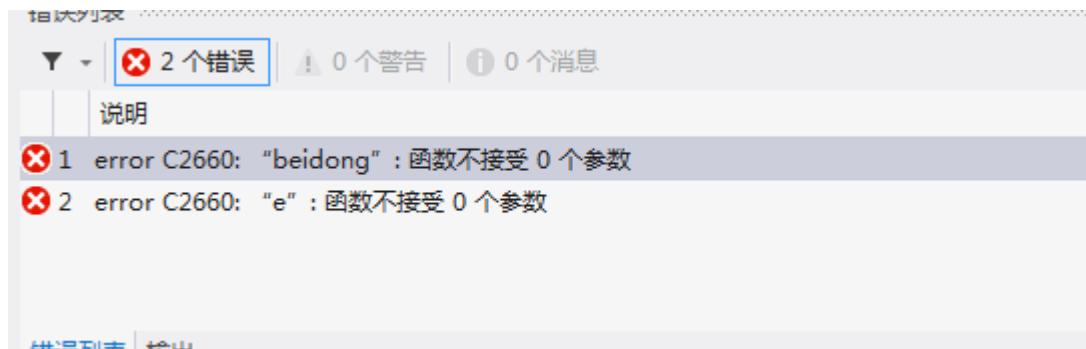
由于在函数定义的时候没有定义参数,在后面调用函数的时候定义了参数造成下面的错误以后在定义函数时及时定义参数及参数类型。



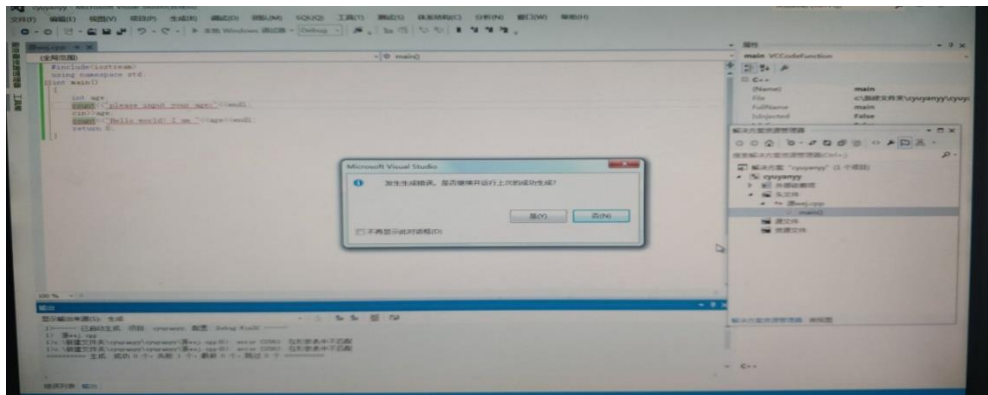
上次调试时的窗口没有关上, 再进行下次调试的时候造成的错误。



在调用函数时没有定义参数个数造成的错误。



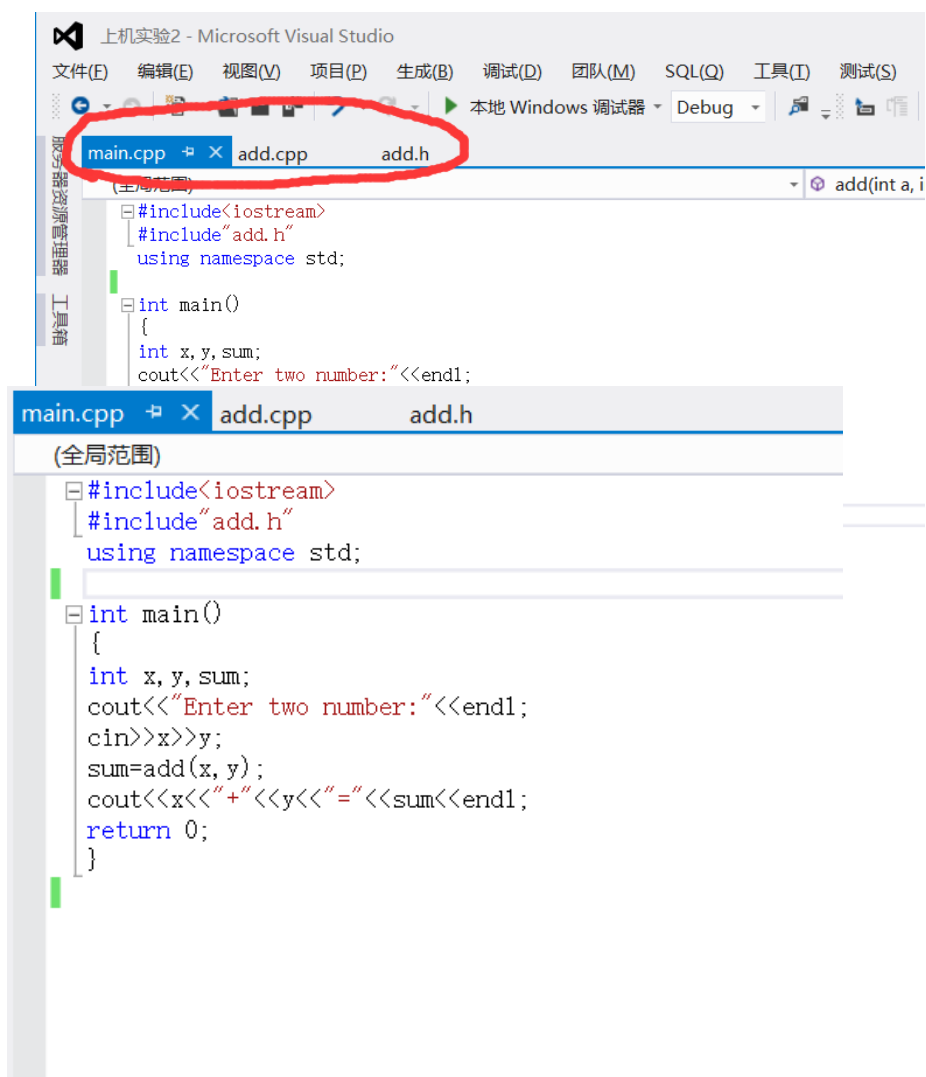
2



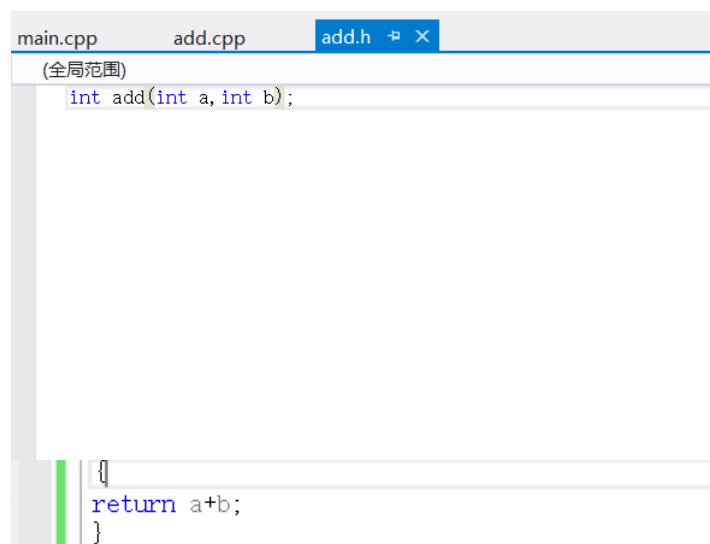
由于对 cout 与 cin 的不熟悉导致 cout 在输入到程序中时多写了一个 n 进而使得程序无法运行, 在对程序进行修改后, 得到了程序运行后的具体界面。

在设计程序时产生的错误及解决方法

接下来，我们使用头文件（.h）的方式写这个简单的加法程序



Add.h



```
return a+b;
}
```

程序运行结果如下图所示：

```
main.cpp add.cpp add.h
(全局范围)
#include <iostream>
#include "add.h"
using namespace std;

int main()
{
    int x, y, sum;
    cout << "Enter two number:" << endl;
    cin >> x >> y;
    sum = add(x, y);
    cout << x << " + " << y << " = " << sum << endl;
    return 0;
}
```

```
C:\WINDOWS\system32\cmd.exe
Enter two number:
3 2
3+2=5
请按任意键继续. . .
```

.h 文件作用:

1. .h 中一般放的是同名.c 文件中定义的变量、数组、函数的声明，需要让.c 外部使用的声明。
2. 方便开发: 包含一些文件需要的共同的常量、结构、类型定义、函数、变量申明；
3. 使函数的作用域从函数声明的位置开始，而不是函数定义的位置(实践总结)
4. 提供接口: 对一个软件包来说可以提供一个给外界的接口(例如: stdio.h)。

4

在编程中遇到的困难：出现下图这种情况是一个对语句应用的失误，在 switch 语句中，每一个分支语句 case 后都要有 break，因为没有跳出，所以输出结果和预想值产生了偏差。解决办法是在每一条 case 语句的结尾加上了 break。

```
上机作业3.cpp yx.h yx.cpp
(全局范围)
#include <iostream>
#include "yx.h"
using namespace std;

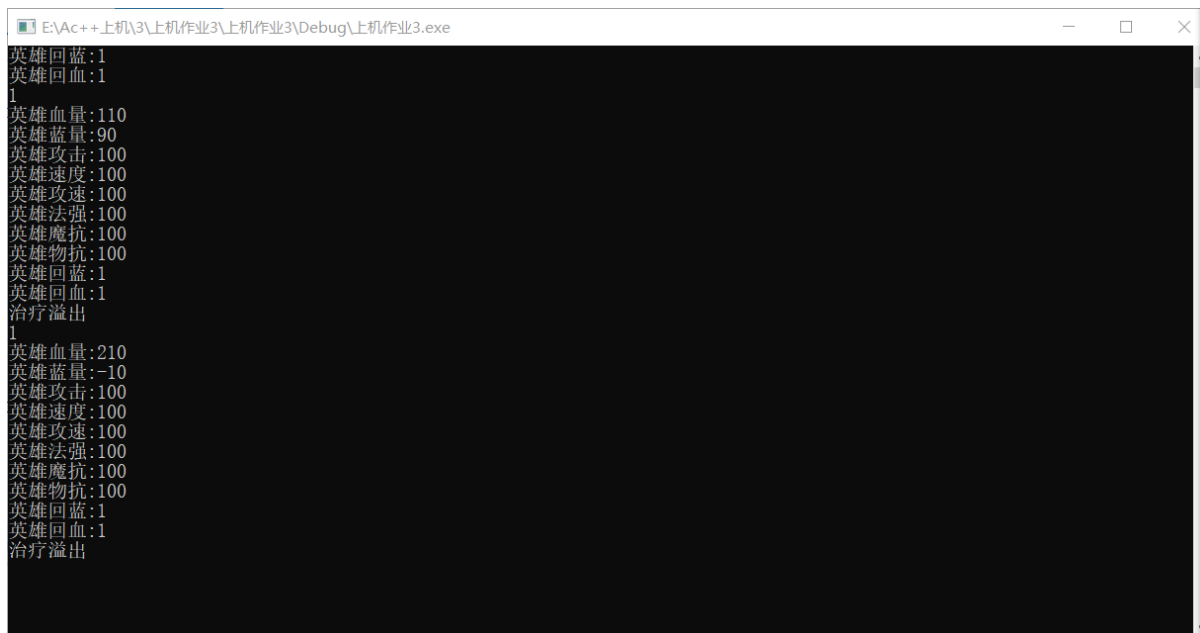
int main()
{
    cout << "欢迎来到王者荣耀--召唤师峡谷" << endl;
    cout << "开始游戏，您的英雄为胡茄离愁蔡文姬，id为连我所爱\n您正在我方水晶。" << endl;
    int xz;
    cout << "1. 胡茄离愁：" << endl;
    cout << "2. 查看英雄属性：" << endl;
    cout << "3. 退出游戏：" << endl;
    cin >> xz;
    while (xz != 3)
    {
        if (xz == 2)
        {
            yx caiwenji(100, 100, 100, 100, 100, 100, 1, 1);
            caiwenji.showd();
        }
        else
        {
            switch (xz)
            {
                case 1: cout << "游戏结束" << endl;
                case 3: cout << "游戏结束" << endl;
                default: cout << "禁止乱按" << endl;
            }
            cin >> xz;
        }
    }
}
```

```
E:\Ac++\上机\上机作业3\上机作业3\Debug\上机作业3.exe
欢迎来到王者荣耀--召唤师峡谷
开始游戏，您的英雄为胡茄离愁蔡文姬，id为连我所爱
您正在我方水晶。
1. 胡茄离愁：
2. 查看英雄属性：
3. 退出游戏：
1
游戏结束
游戏结束
禁止乱按
1
游戏结束
游戏结束
禁止乱按
11
禁止乱按
这是你的英雄的属性
英雄血量:100
英雄蓝量:100
英雄攻击:100
英雄速度:100
英雄攻速:100
英雄法强:100
英雄魔抗:100
```

错误 2：在对蓝量的判断语句 `if (caiwenji.d2=0)` //魔法值是否为 0

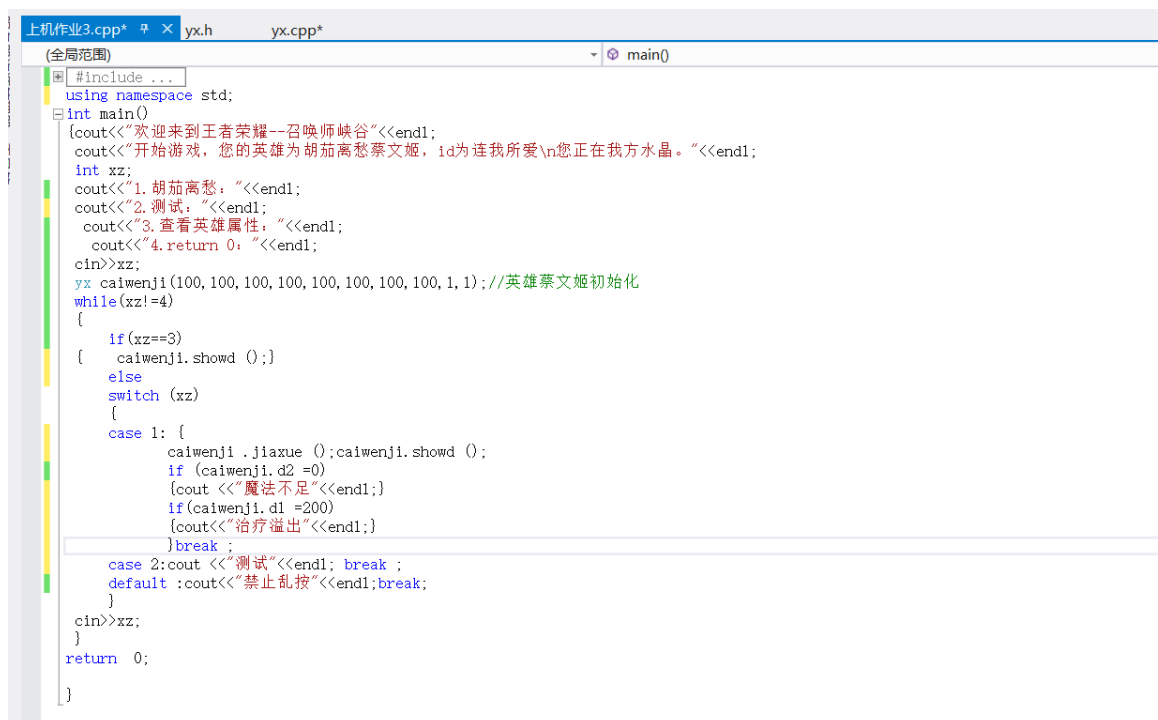
犯了一个原则性错误，`=`在 `c++`中是赋值语句，而判断语句是`==`，所以在执行过程中发生了如下所示的错误。

改为 `if (caiwenji.d2==0)` 后，程序实现了功能。



```
E:\Ac++\上机\3\上机作业3\上机作业3\Debug\上机作业3.exe
英雄回蓝:1
英雄回血:1
1
英雄血量:110
英雄蓝量:90
英雄攻击:100
英雄速度:100
英雄攻速:100
英雄法强:100
英雄魔抗:100
英雄物抗:100
英雄回蓝:1
英雄回血:1
治疗溢出
1
英雄血量:210
英雄蓝量:-10
英雄攻击:100
英雄速度:100
英雄攻速:100
英雄法强:100
英雄魔抗:100
英雄物抗:100
英雄回蓝:1
英雄回血:1
治疗溢出
```

如下图，最终实现了蔡文姬胡茄离愁技能的加血和耗蓝操作。



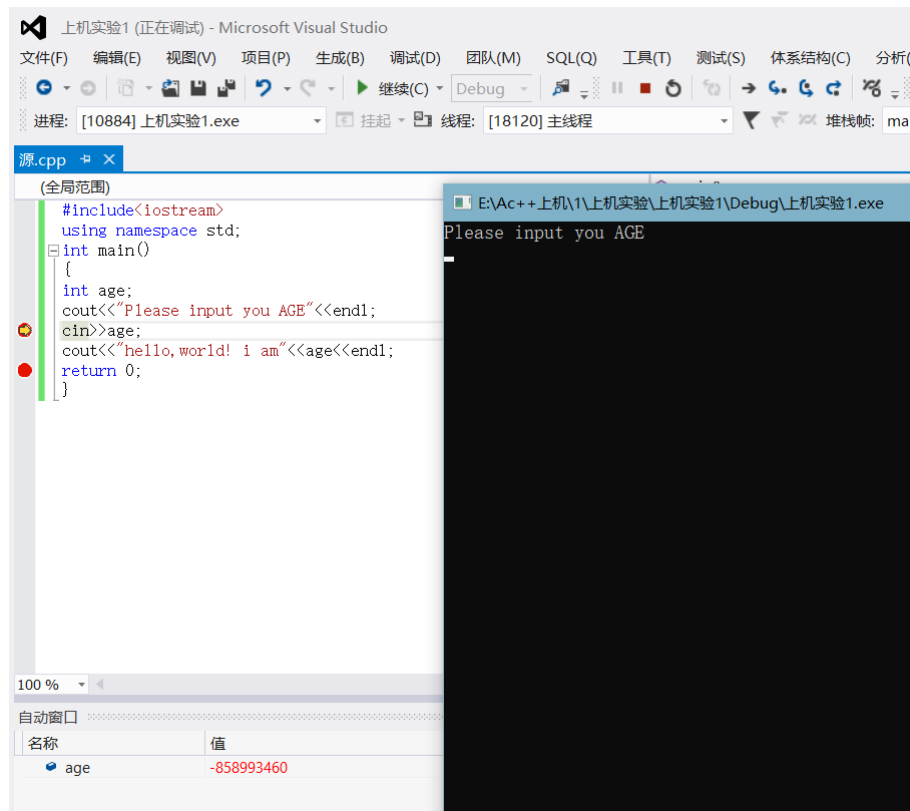
```
上机作业3.cpp*  x yx.h  yx.cpp*
(全局范围)  main()
#include ...
using namespace std;
int main()
{
    cout<<"欢迎来到王者荣耀--召唤师峡谷"<<endl;
    cout<<"开始游戏，您的英雄为胡茄离愁蔡文姬，id为连我所爱\n您正在我方水晶。"<<endl;
    int xz;
    cout<<"1. 胡茄离愁："<<endl;
    cout<<"2. 测试："<<endl;
    cout<<"3. 查看英雄属性："<<endl;
    cout<<"4. return 0;"<<endl;
    cin>>xz;
    yx caiwenji(100,100,100,100,100,100,100,100,1,1); //英雄蔡文姬初始化
    while(xz!=4)
    {
        if(xz==3)
        {
            caiwenji.showd();
        }
        else
        {
            switch(xz)
            {
                case 1: {
                    caiwenji.jiaxue();caiwenji.showd();
                    if(caiwenji.d2==0)
                    {cout<<"魔法不足"<<endl;}
                    if(caiwenji.d1==200)
                    {cout<<"治疗溢出"<<endl;}
                    break;
                }
                case 2: cout<<"测试"<<endl; break;
                default: cout<<"禁止乱按"<<endl; break;
            }
        }
        cin>>xz;
    }
    return 0;
}
```

其次，实现了攻击韩信，若韩信死亡，显示死亡状态，并析构假人韩信这个析构函数。

```
E:\Ac++\上机\3\上机作业3\上机作业3\Debug\上机作业3.exe
欢迎来到王者荣耀--召唤师峡谷
开始游戏，您的英雄为胡茄离愁蔡文姬，id为连我所爱
您正在我方水晶。
1. 胡茄离愁:
2. 攻击假人韩信/庄周:
3. 查看英雄属性:
4. return 0:
1
英雄血量:110
英雄蓝量:90
英雄攻击:100
英雄速度:100
英雄攻速:100
英雄法强:100
英雄魔抗:100
英雄物抗:100
英雄回蓝:1
英雄回血:1
1
英雄血量:120
英雄蓝量:80
英雄攻击:100
英雄速度:100
英雄攻速:100
英雄法强:100
英雄魔抗:100
英雄物抗:100
英雄回蓝:1
英雄回血:1
```

```
E:\Ac++\上机\3\上机作业3\上机作业3 E:\Ac++\上机\3\上机作业3\Debug\上机作业3.exe
1. 胡茄离愁: 您正在我方水晶。
2. 攻击假人韩信/庄周: 1. 胡茄离愁:
3. 查看英雄属性: 2. 攻击假人韩信/庄周:
4. return 0: 3. 查看英雄属性:
2 4. return 0:
英雄血量:50 3
英雄蓝量:100 英雄血量:100
英雄攻击:0 英雄蓝量:100
英雄速度:0 英雄攻击:100
英雄攻速:0 英雄速度:100
英雄法强:0 英雄攻速:100
英雄魔抗:0 英雄法强:100
英雄物抗:0 英雄魔抗:100
英雄回蓝:0 英雄物抗:100
英雄回血:0 英雄回蓝:1
2 英雄回血:1
英雄血量:0 1
英雄蓝量:100 英雄血量:110
英雄攻击:0 英雄蓝量:90
英雄速度:0 英雄攻击:100
英雄攻速:0 英雄速度:100
英雄法强:0 英雄攻速:100
英雄魔抗:0 英雄法强:100
英雄物抗:0 英雄魔抗:100
英雄回蓝:0 英雄物抗:100
英雄回血:0 英雄回蓝:1
韩信死亡 英雄回血:1
韩信已经死了 3
我们要摧毁它的函数 英雄血量:110
英雄蓝量:90
英雄攻击:100
英雄速度:100
英雄攻速:100
英雄法强:100
英雄魔抗:100
英雄物抗:100
英雄回蓝:1
英雄回血:1
```

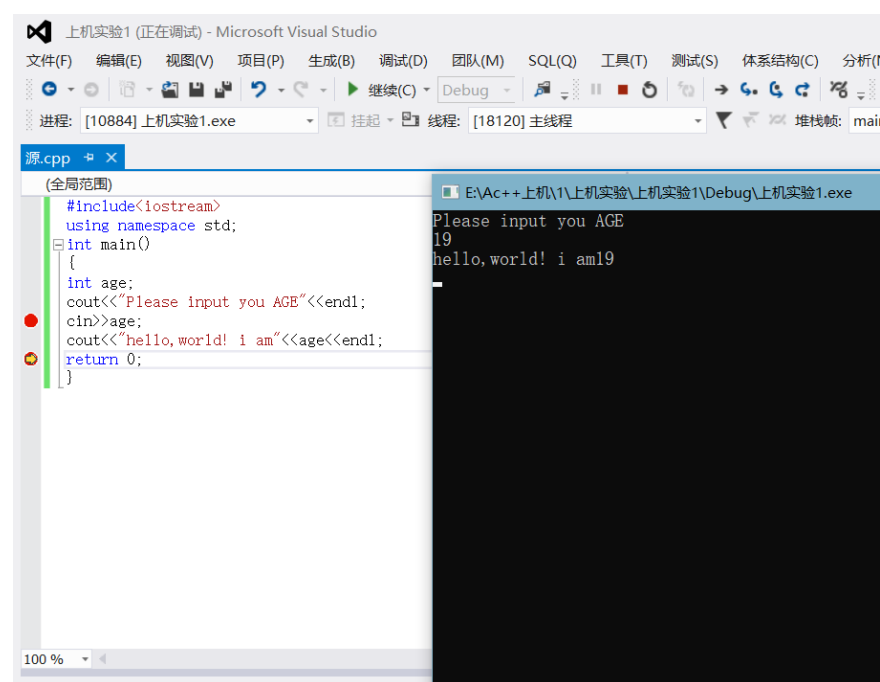
5 运行结果以及产生的错误



如图，分别在程序的第七行和第九行添加了断点，当进行本地调试的时候，程序会自动停止在当前位置。

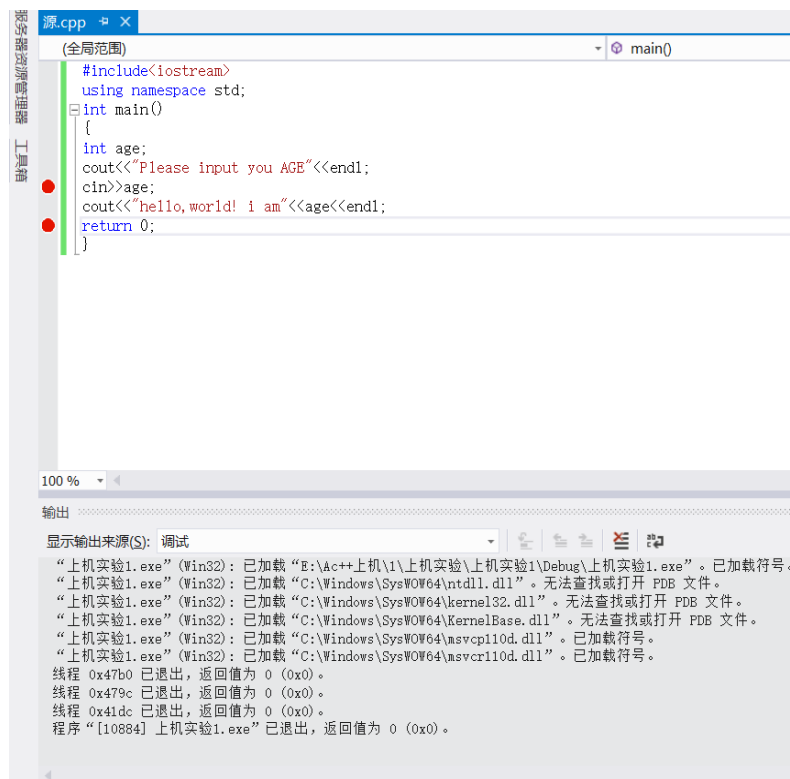
在你键入年龄，并回车后，win32 控制台界面并不会发生任何变化，我认为这**并不是一个 bug**，仅仅是因为程序在运行到了断点处而被中断。

点击继续执行



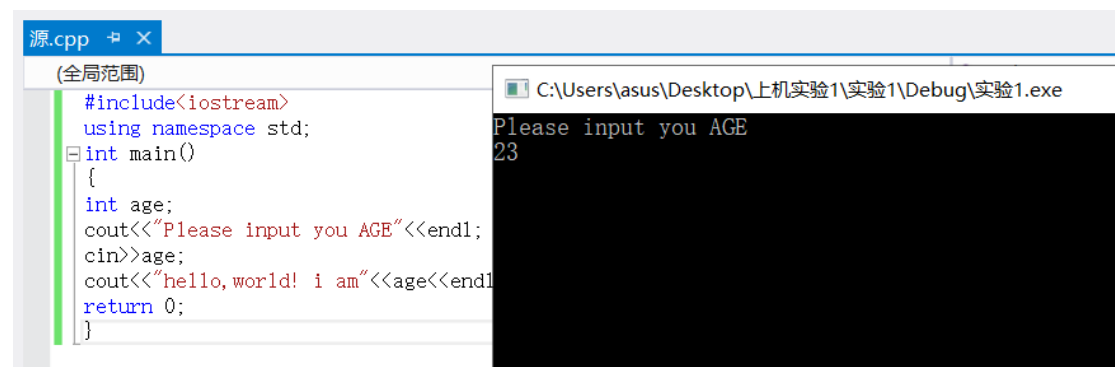
由上图可以看出，在继续执行后，程序停留在了return 0; 也就是第二个断点处，因此，我们清楚的感受到程序的运行过程。

继续执行，程序终止。返回值为 0

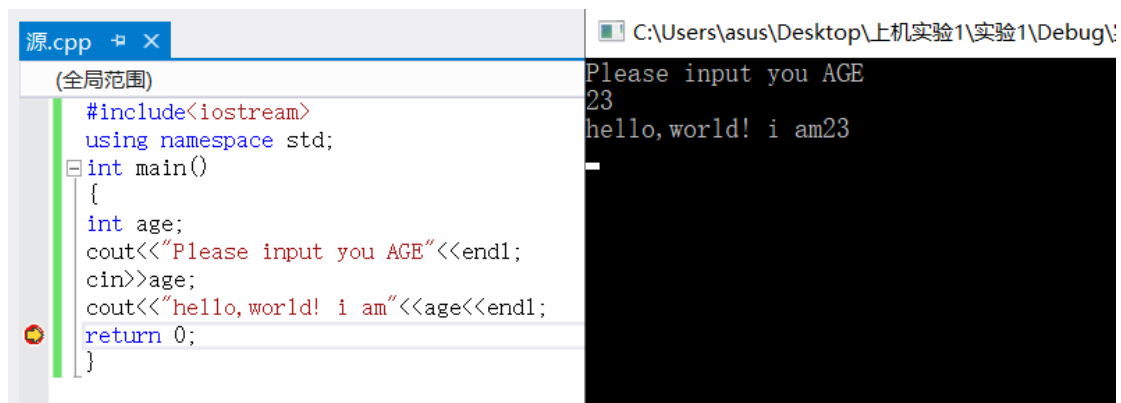


6

运行结果以及产生的错误



在这里程序会出现结果，并且闪退，所以我们在这里引入了断点。



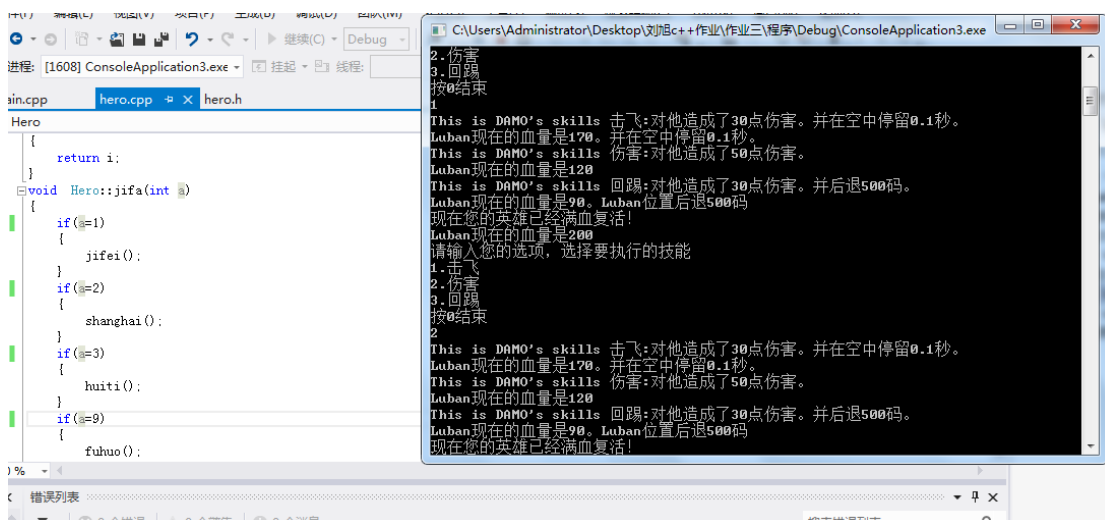
方法一：在程序断点，当进行本地调试的时候，程序会自动停止在当前位置，从而不执行其 return 指令，则不会闪退。点击继续执行，程序退出。

方法二：运用开始执行（不调试），也可以防止程序出现闪退的现象。

方法三：运用#include<conio.h>来调用 getch() 这个函数

7

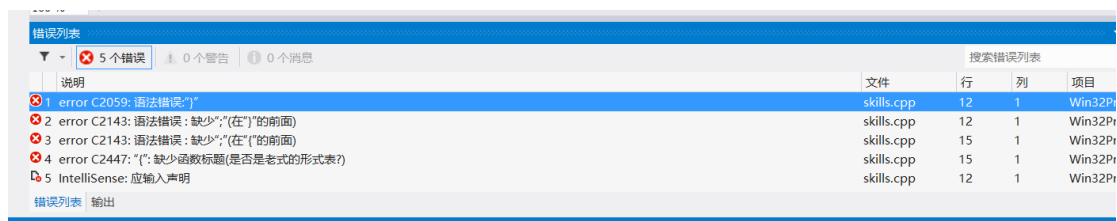
错误与改正



在实现技能的选择过程中，少添加了一个“=”，应该使用逻辑等“==”，使得技能的选择出现了差错，技能的效果同时实现了，并没有达到理想的结果。

8

一、遇到问题



二、分析问题并解决

1、 当我使用 if 语句时，程序中当孙尚香攻击使用二技能时，如果敌人在其有效、范围

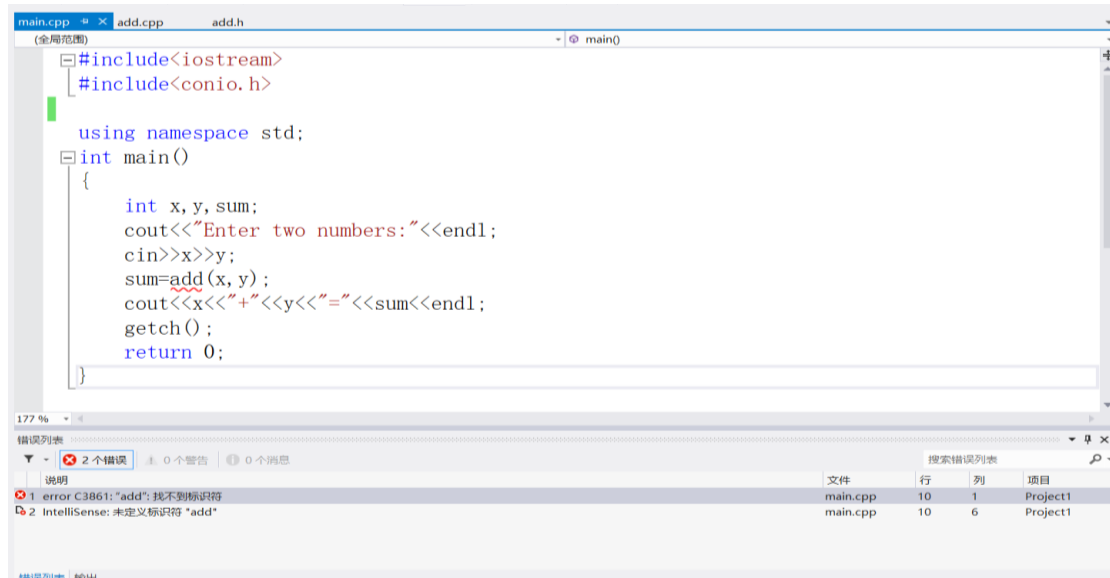
内将会触发技能效果，否则将会失败，在其中没有写出 else 的这种情况，所以会导致程序出现问题。

- 2、在定义活力劲发这个技能函数中少写了一个大括号，这也导致程序在调试时出现了错误。

当将这两个错误改正后，程序运行成功，最终实现五连绝世的输出。

9

产生的错误及解决方法

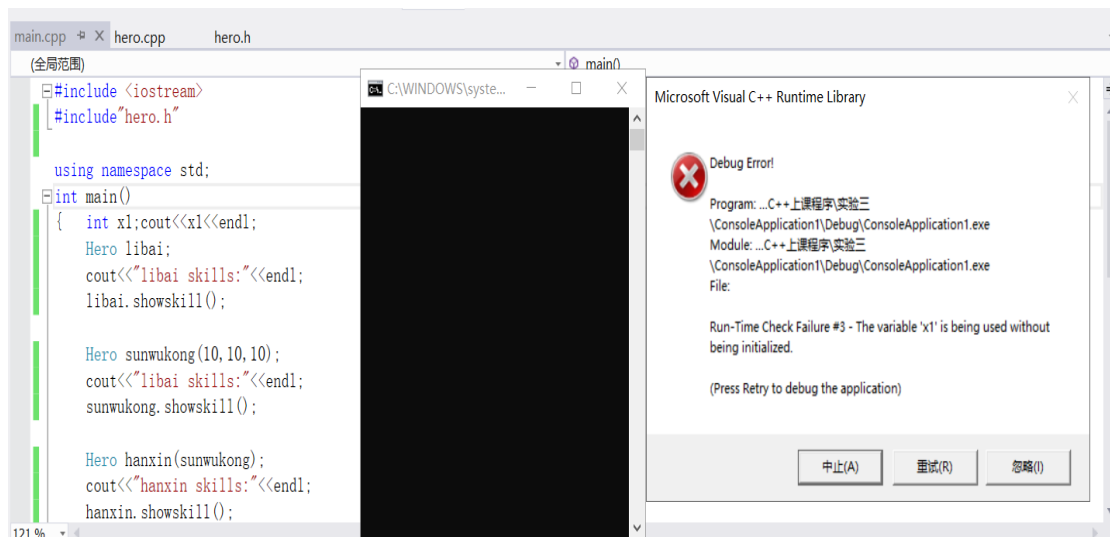


产生的错误：“add”找不到标识符；未定义标识符“add”。

解决方法：因为在 main.cpp 中并没有对 add 这个函数进行声明或者定义。而在 add.h 中进行了声明，所以调用时需要在 main.cpp 头文件中加上#include<add.h>来进行调用。

10

产生的错误及解决方法

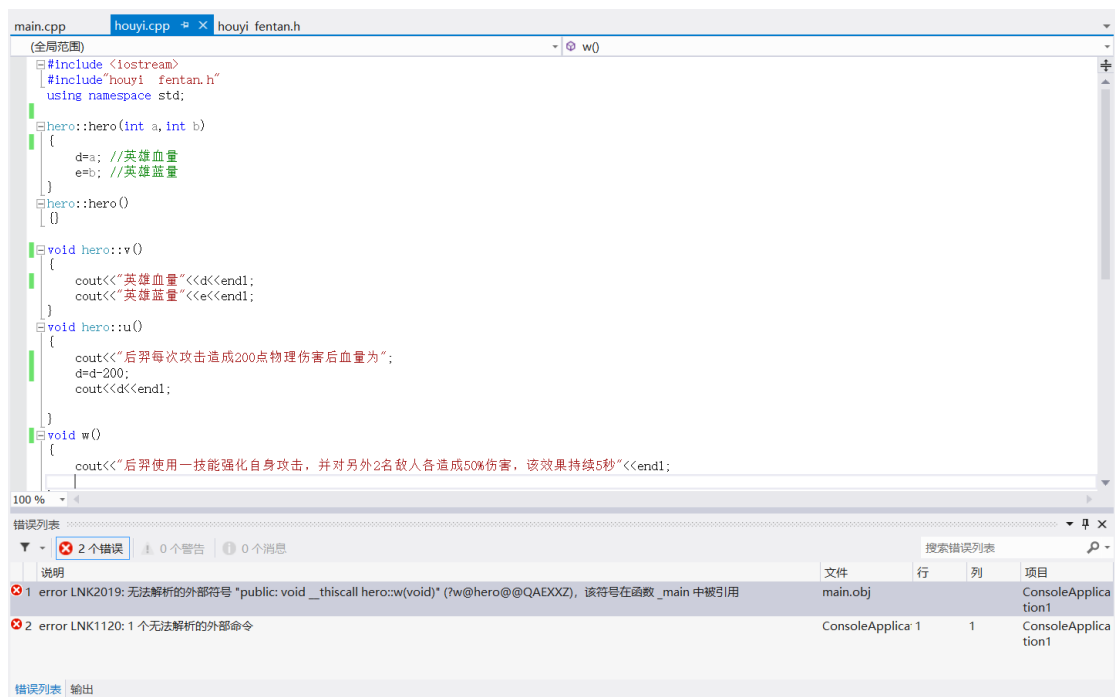


实验产生的错误：变量 x1 在未初始化的情况下被使用。

解决方法：将该程序的第五行“int x1; cout<<x1<<endl;”删去即可。

11

实验错误

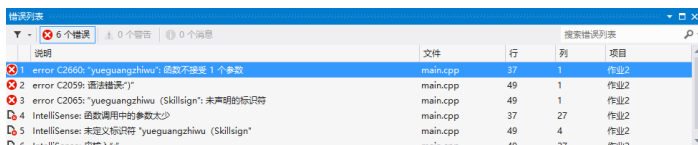


产生的错误：(1) 无法解析的外部符号 "public: void __thiscall hero::w(void)" (?w@hero@@@QAEXXZ)，该符号在函数 _main 中被引用。(2) 1 个无法解析的外部命令。

解决方法：说实话这个问题一直困扰我很长时间。**第一次出现**，是刚安装上这个程序，当时出现直接蒙了，应为这个错误 Debug 只给报错，但是不说是哪里有错，我请教了多方神圣他们说你的 Debug 中没有生成 exe 文件，当时一致认为我的程序软件出了问题。然后我就把软件卸载了重新安装了，然而并没有什么软用，安装后照样还是这个问题；**第二次出现**，是在做这个程序的时候，当时有请教了百度，百度说我的文件项目建错了，然后重新建了工程后，确实老长时间没有在出现过。但是直到今天。**第三次出现**，我在仔细阅读了我的程序确实程序中看似没错误，但是就是因为往往被容易忽略的问题。**改的地方：**把程序中的“void w()”函数改为“void hero::w()”。这样就可以解决函数出现的问题。

13

出现的错误：



解决：

在调动被动技能时，加入一个形参变量 attack_number

利用断点单步调试程序，
在每次释放大招时设置断点
总共释放四次大招

```
int origin_x, origin_y;
cout<<"luna location:"<<endl;
cin>>origin_x>>origin_y;
int attack_number = 0;
bool Skillsign = false;
int damage_count=0;
bool damage_sign = false;
attack(attack_number);
attack(attack_number);

xiangyueshan(Skillsign);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);

damage_count++;
attack(attack_number);
yueguangzhivu(attack_number, Skillsign);

damage_sign= xinyuetuji(origin_x, origin_y, Skillsign);
damage_count++;

attack(attack_number);
shirujiamang(Skillsign);
attack(attack_number);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);

damage_count++;
attack(attack_number);
yueguangzhivu(attack_number, Skillsign);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);
```

Enter Five targets:
后羿
嫦娥
鲁班
小乔
鲁班 location:
0 0
Attack number
0
Attack number
1
Do magic damage 30 and the skillsign duration is 5s
Enter lenth X and length
d
Wait for the cooling timeAttack number
2
Do magic damage 30 and the skillsign duration is 5s

```
int origin_x, origin_y;
cout<<"luna location:"<<endl;
cin>>origin_x>>origin_y;
int attack_number = 0;
bool Skillsign = false;
int damage_count=0;
bool damage_sign = false;
attack(attack_number);
attack(attack_number);

xiangyueshan(Skillsign);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);

damage_count++;
attack(attack_number);
yueguangzhivu(attack_number, Skillsign);

damage_sign= xinyuetuji(origin_x, origin_y, Skillsign);
damage_count++;

attack(attack_number);
shirujiamang(Skillsign);
attack(attack_number);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);

damage_count++;
attack(attack_number);
yueguangzhivu(attack_number, Skillsign);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);
```

Enter Five targets:
后羿
嫦娥
鲁班
小乔
鲁班 location:
0 0
Attack number
0
Attack number
1
Do magic damage 30 and the skillsign duration is 5s
Enter lenth X and length
d
Wait for the cooling timeAttack number
2
Do magic damage 30 and the skillsign duration is 5s

```
int origin_x, origin_y;
cout<<"luna location:"<<endl;
cin>>origin_x>>origin_y;
int attack_number = 0;
bool Skillsign = false;
int damage_count=0;
bool damage_sign = false;
attack(attack_number);
attack(attack_number);

xiangyueshan(Skillsign);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);

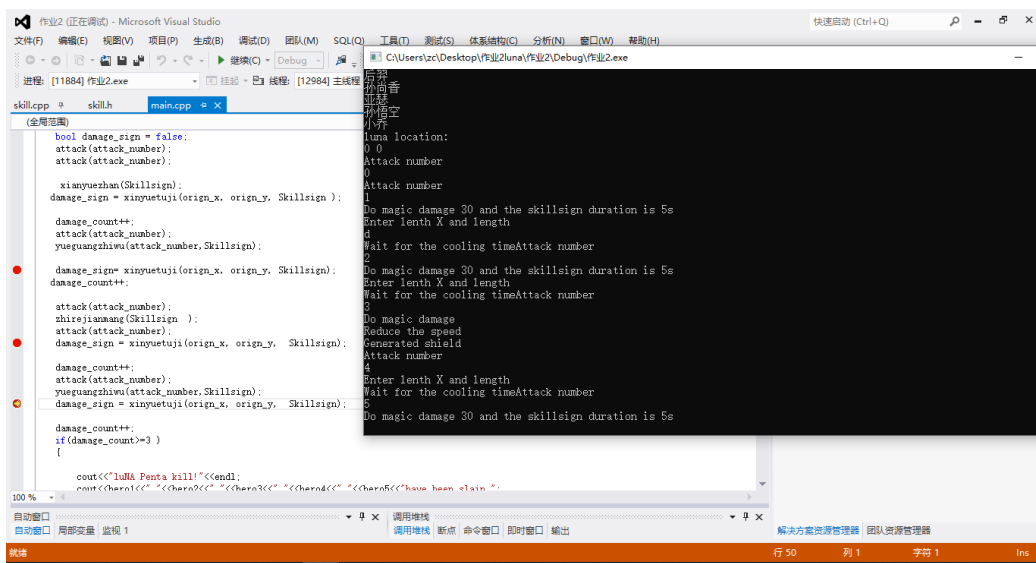
damage_count++;
attack(attack_number);
yueguangzhivu(attack_number, Skillsign);

damage_sign= xinyuetuji(origin_x, origin_y, Skillsign);
damage_count++;

attack(attack_number);
shirujiamang(Skillsign);
attack(attack_number);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);

damage_count++;
attack(attack_number);
yueguangzhivu(attack_number, Skillsign);
damage_sign = xinyuetuji(origin_x, origin_y, Skillsign);
```

Enter Five targets:
后羿
嫦娥
鲁班
小乔
鲁班 location:
0 0
Attack number
0
Attack number
1
Do magic damage 30 and the skillsign duration is 5s
Enter lenth X and length
d
Wait for the cooling timeAttack number
2
Do magic damage 30 and the skillsign duration is 5s
Enter lenth Y and length
d
Wait for the cooling timeAttack number
3
Do magic damage
Reduce the speed
Generated shield
Attack number
4



14

如图所示，执行程序后出现 6 个错误：

| 错误列表 | | | | | 搜索错误列表 | | |
|---------------------------------|----------|----|---|-------------------------|--------|--|--|
| 6 个错误 | | | | | | | |
| 说明 | 文件 | 行 | 列 | 项目 | | | |
| 1 error C2065: "jisha": 未声明的标识符 | main.cpp | 19 | 1 | ConsoleApplica tion1 | | | |
| 2 error C2065: "jisha": 未声明的标识符 | main.cpp | 20 | 1 | ConsoleApplica tion1 | | | |
| 3 error C2065: "jisha": 未声明的标识符 | main.cpp | 21 | 1 | ConsoleApplica tion1 | | | |
| 4 error C2065: "jisha": 未声明的标识符 | main.cpp | 22 | 1 | ConsoleApplica tion1 | | | |
| 5 error C2065: "jisha": 未声明的标识符 | main.cpp | 23 | 1 | ConsoleApplica tion1 | | | |
| 6 IntelliSense: 未定义标识符 "jisha" | main.cpp | 19 | 6 | ConsoleApplica tion1 | | | |

错误原因：没有理解好局部变量和全局变量的区别，局部变量是指在函数内定义的变量，即在一个函数内部定义的变量，只在本函数范围内有效。而全局变量是指在函数外定义的变量，即从定义变量的位置到本源文件结束都有效。在程序中，当有外部变量时，main 函数中的同名变量就是全局变量。如果没有外部变量，main 函数里的变量依然是局部变量。其目的是增加函数间数据联系的渠道。由于同一文件中的所有函数都能引用全局变量的值，因此如果在一个函数中改变了全局变量的值，就能影响到其他函数中全局变量的值，相当于各个函数之间有直接的传递渠道。

修改程序为：

```
#include <iostream>
#include "app.h"
using namespace std;
int jisha;
inline void gongji ()
{
    jisha=jisha+1;
}
int main()
{
    cout<<"welcome to 召唤师峡谷"<<endl;
    cout<<"游戏开始，英雄盲僧\n1 杀人 2 退出按其他你试试"<<endl;
```

```

int LOL;
cin>>LOL;
while(LOL!=2)
{
    if (LOL==1){gongji();}
    if (jisha==1){cout<<"*****First Blood*****"<<endl;}
    if (jisha==2){cout<<"*****Double Kill*****"<<endl;}
    if (jisha==3){cout<<"*****Triple Kill*****"<<endl;}
    if (jisha==4){cout<<"*****Quadra Kill*****"<<endl;}
    if (jisha==5){cout<<"*****Penta Kill*****"<<endl;}
    cin>>LOL ;
}
return 0;

```

15

错误

执行程序后输入数字之后持续向后滚动输出，无法停止。

```

攻击:100
血量:-270750
攻击:100
血量:-270800
攻击:100
血量:-270850
攻击:100
血量:-270900
攻击:100
血量:-270950
攻击:100
血量:-271000
攻击:100
血量:-271050
攻击:100
血量:-271100
攻击:100
血量:-271150
攻击:100
血量:-271200
攻击:100
血量:-271250
攻击:100
血量:-271300
攻击:100
血量:-271350
攻击:100
血量:-271400
攻击:100
血量:-

```

修改 main.cpp 程序为:

```

#include <iostream>
#include "zy3.h"
using namespace std;
int CJ;
int main()
{cout<<"欢迎来到英雄联盟"<<endl;
    cout<<"英雄为盲僧"<<endl;
    int LM;
    LOL Liqing(2100,100);
    LOL bluebuff(3200,100);
    bluebuff.CJ=2;
    cout<<"1.天音波/回音击"<<endl;

```

```

cout<<"2.天雷破/催筋断骨"<<endl;
cout<<"3.退出游戏"<<endl;
cin>>LM;
while(LM!=3)
{
    switch (LM)
    {
        case 1:{
            bluebuff.FLZT2();bluebuff.showd ();
            if (Liqing.QQQ ==0)
            {cout <<"能量不足"<<endl;
            }}
            break ;
        case 2: if (CJ=0){cout<<"惩戒次数不足"<<endl;cin>>LM;}
            bluebuff.FLZT1(); bluebuff.showd ();
            if (bluebuff .FFF==0)
            {cout<<"击杀蓝 buff"<<endl ;
            bluebuff.~LOL ();}break ;
        default :cout<<"biu~~"<<endl;break;
    }
    cin>>LM;
}
return 0;
}

```

16

一. 实验中的错误

1. 在用对象调用成员函数时后面没加 ();

| | | | | | |
|-----|--------------------------------------------------------------------------------------------|----------|----|---|-----|
| ✖ 1 | error C3867: "hero::chengshoushanghai": 函数调用缺少参数列表; 请使用"&hero::chengshoushanghai"创建指向成员的指针 | main.cpp | 11 | 1 | 作业4 |
| ✖ 2 | error C3867: "hero::chengshoushanghai": 函数调用缺少参数列表; 请使用"&hero::chengshoushanghai"创建指向成员的指针 | main.cpp | 12 | 1 | 作业4 |

修改后:

```

int shanghai=700;
hero yingxing(1000);
yingxing.jineng(shanghai);
yingxing.chengshoushanghai(shanghai);

```

2. shanghai 变量没有在承受伤害的函数中引用

```

void hero::chengshoushanghai()
{
    shengmingzhi1=shanghai+2;
    shengmingzhi2=xueliang3 -(shanghai/3);
    shengmingzhi3=xueliang1 +(shanghai/3);
    cout<<"李白当前生命值为"<<shengmingzhi1<<endl;
    cout<<"韩信当前生命值为"<<shengmingzhi2<<endl;
    cout<<"孙悟空当前生命值为"<<shengmingzhi3<<endl;
};

void hero::jineng(int shanghai)
{
    cout<<"孙悟空技能造成的伤害为"<<shanghai
    <<endl;
};

void hero::chengshoushanghai()

```


结果为：

修改：在 chengshoushanghai 函数后面引用变量 sahnghai

```
void hero::chengshoushanghai(int shanghai)
{
    shengmingzhi1=xueliang2-(shanghai/2);
    shengmingzhi2=xueliang3 -(shanghai/3);
    shengmingzhi3=xueliang1 +(shanghai/3+shanghai/2)/3;
    cout<<"李白当前生命值为"<<shengmingzhi1<<endl;
    cout<<"韩信当前生命值为"<<shengmingzhi2<<endl;
    cout<<"孙悟空当前生命值为"<<shengmingzhi3<<endl;
};
```

则输出结果为：

17

作业程序出现错误及改正

1、出现错误

在 hero.h 程序中未声明标识符，程序虽然可以运行显示结果，但是错误列表仍会显示错误。

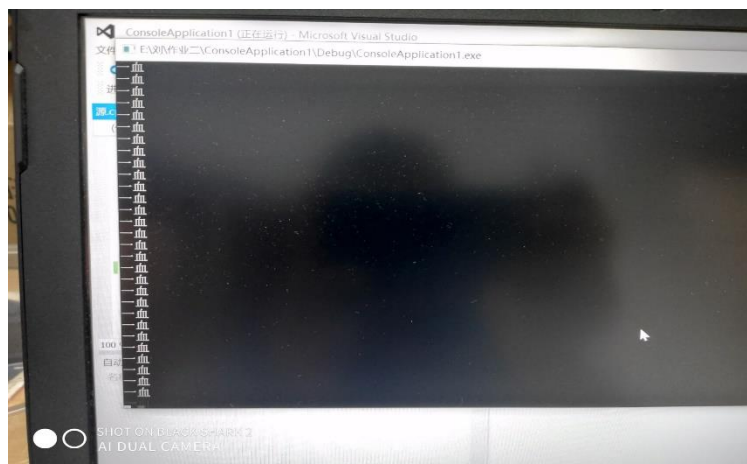


2、改正

在 hero.h 程序最后声明标识符 a, b。

18

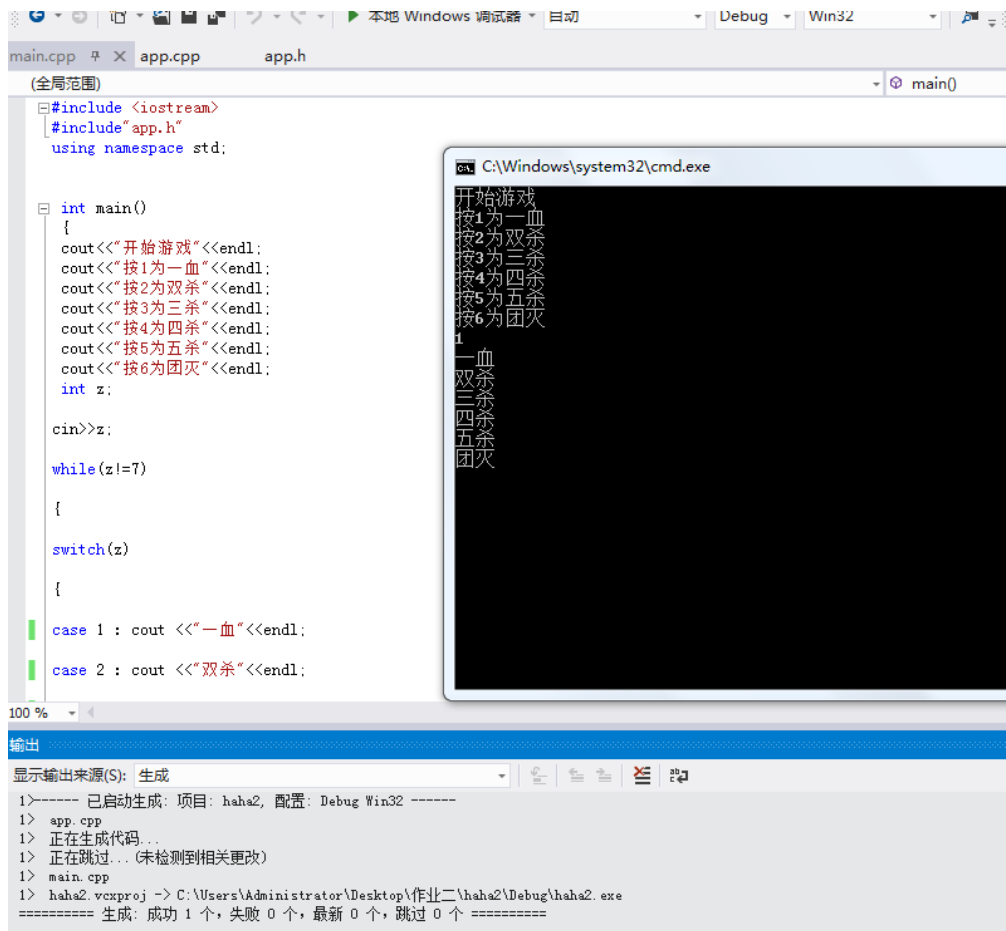
出现的问题



进行普攻后，一血循环输出，while 语句使用错误，加入 break 跳脱后，变正常

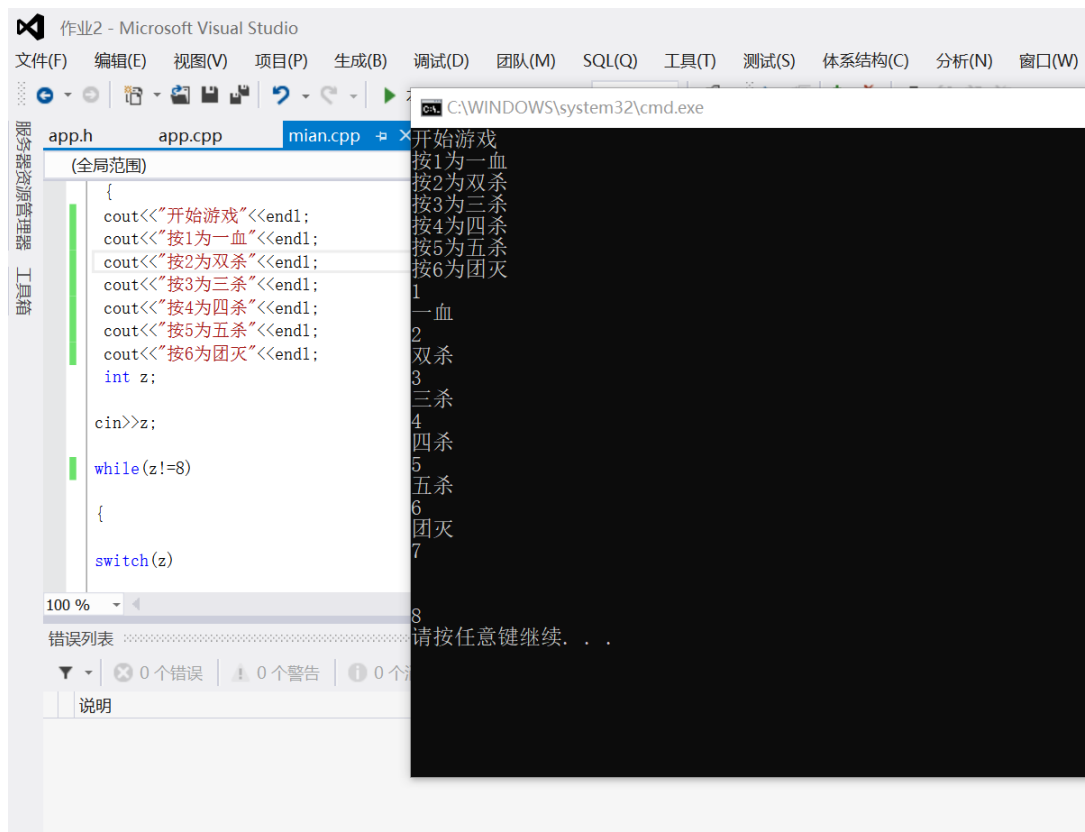
19

产生的错误：调用头文件的指令顺手把"<"错输入为<>，这不符合头文件调用语句的规则。所以出现了错误警报，如下图：



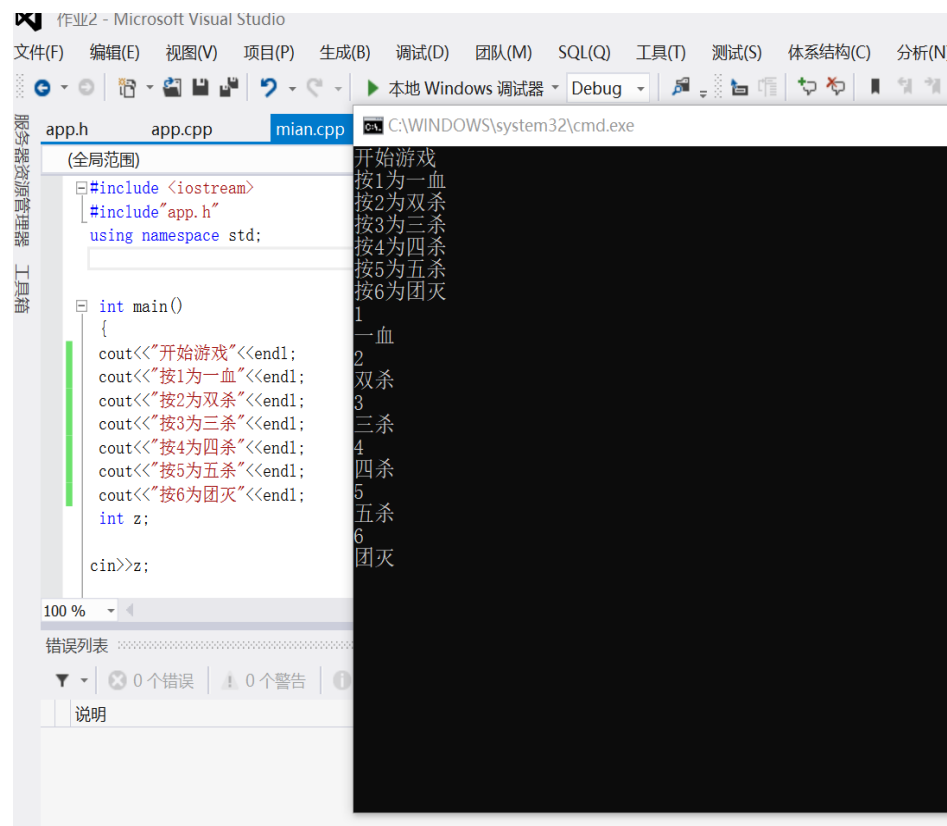
(2)

while 循环语句后的 z 的值取得太大，导致按 6 之后的数字程序不会结束：



四、错误改正：

加上 break 语句和更改 z 的取值范围以使程序可以正常运行，改完后程序运行结果为：



22

作业程序出现错误及改正

1、出现错误

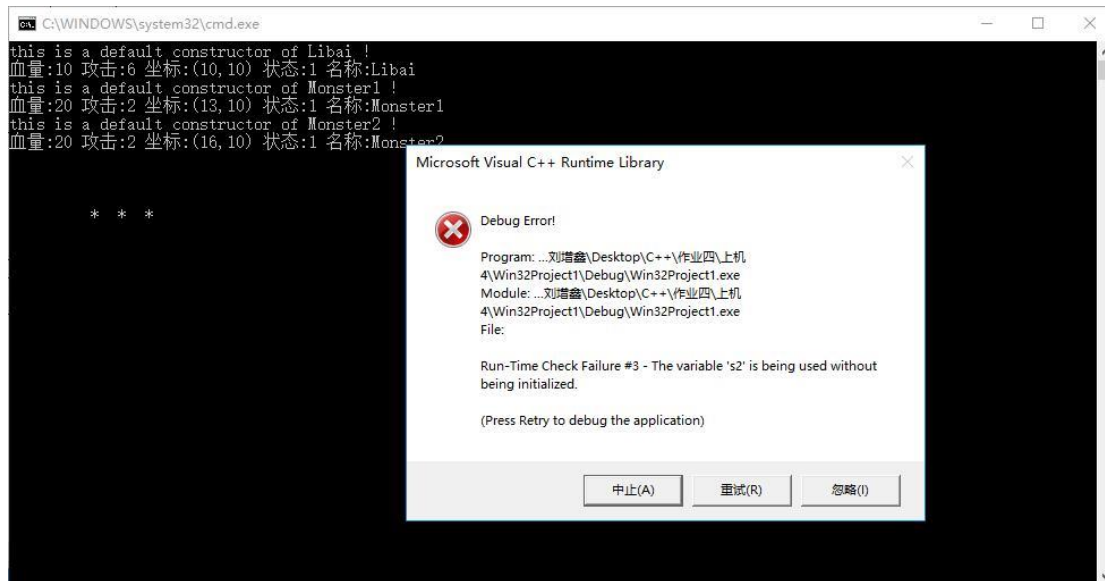


改正

这是因为类 HERO2 的参数放在外部定义出现的错误。将在 HERO2()上面一行定义的参数 int n 移到 HERO2 () 括号内既可。

23

在调试过程中，出现错误如下

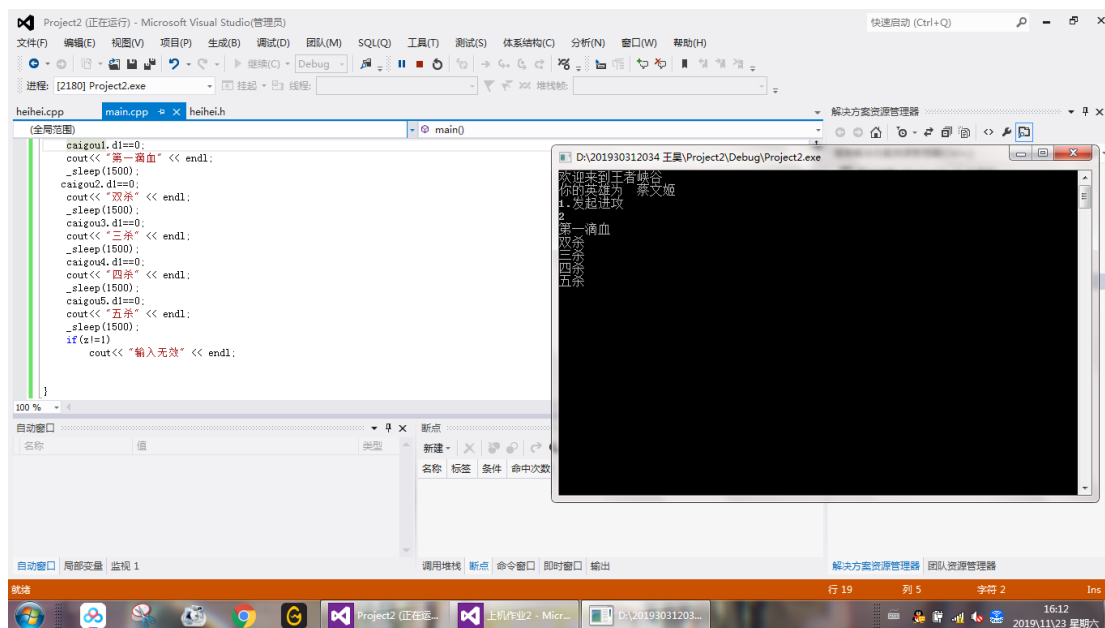


即定义了 s2,也使用了 s2, 却没有给 s2 赋初值（初始化）。

24

● 出现问题

当输入 2 或其他数字时, 执行与 1 同样的操作



● 解决方案

起初尝试用 if, else 选择语句进行区分。期望运行时输入 1 时, 正确执行五杀, 输入非 1 时, 无反应或者显示输入错误, 结果发现不大会用, 于是选择使用 switch 语句,

```
while(z!=2)
{
    switch(z)
    {
        case 1:caigou1.at();break ;
        default:cout<< "输入错误" << endl;break;
    }
}
```

并找出了几个错误

```
_sleep(1500);
caigou3.d1==0;
cout<< "三杀" << endl;
std::ostream std::cout
cout<< "四杀" << endl;
_sleep(1500);
caigou5.d1==0;
cout<< "五杀" << endl;
_sleep(1500);}
cin>>z;
}

return 0;
```

错误 1 . , 程序最后没有添加 return0;

错误 2. while(z!=2) 没有对 z 进行限制

错误 3. case 1:caigou1.at();break ; 没写 break, 跳出循环

25

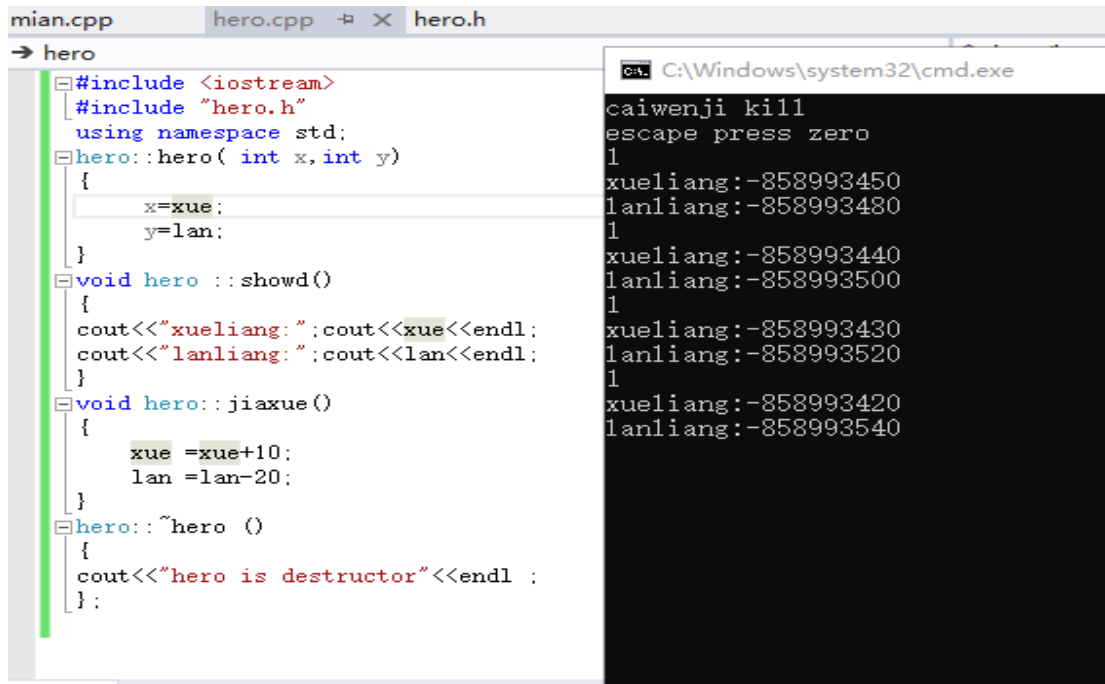
错误及总结

The screenshot shows a C++ IDE with a code editor and an error list. The code editor displays the following code:

```
#include <iostream>
#include "hero.h"
using namespace std;
hero::hero( int x,int y)
{
    int x=xue;
    int y=lan;
}
void hero::showd()
{
    cout<<"xueliang:";cout<<xue<<endl;
    cout<<"lanliang:";cout<<lan<<endl;
}
void hero::jiaxue()
{
    xue =xue+10;
    lan =lan-20;
}
hero::~hero ()
{
    cout<<"hero is destructor"<<endl ;
};
```

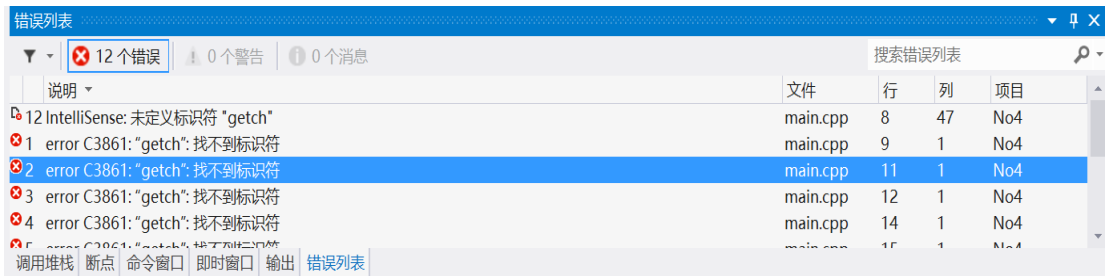
The error list at the bottom shows two errors:

| 说明 | 文件 | 行 | 列 | 项目 |
|--------------------------|----------|---|---|--------|
| 1 error C2082: 形参"x"的重定义 | hero.cpp | 6 | 1 | 蔡文姬一技能 |
| 2 error C2082: 形参"y"的重定义 | hero.cpp | 7 | 1 | 蔡文姬一技能 |

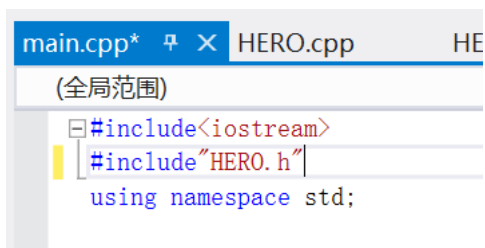


重定义即去掉 int,
第二个错误造成初值赋不上, 是因为方向反了。

26 出现错误

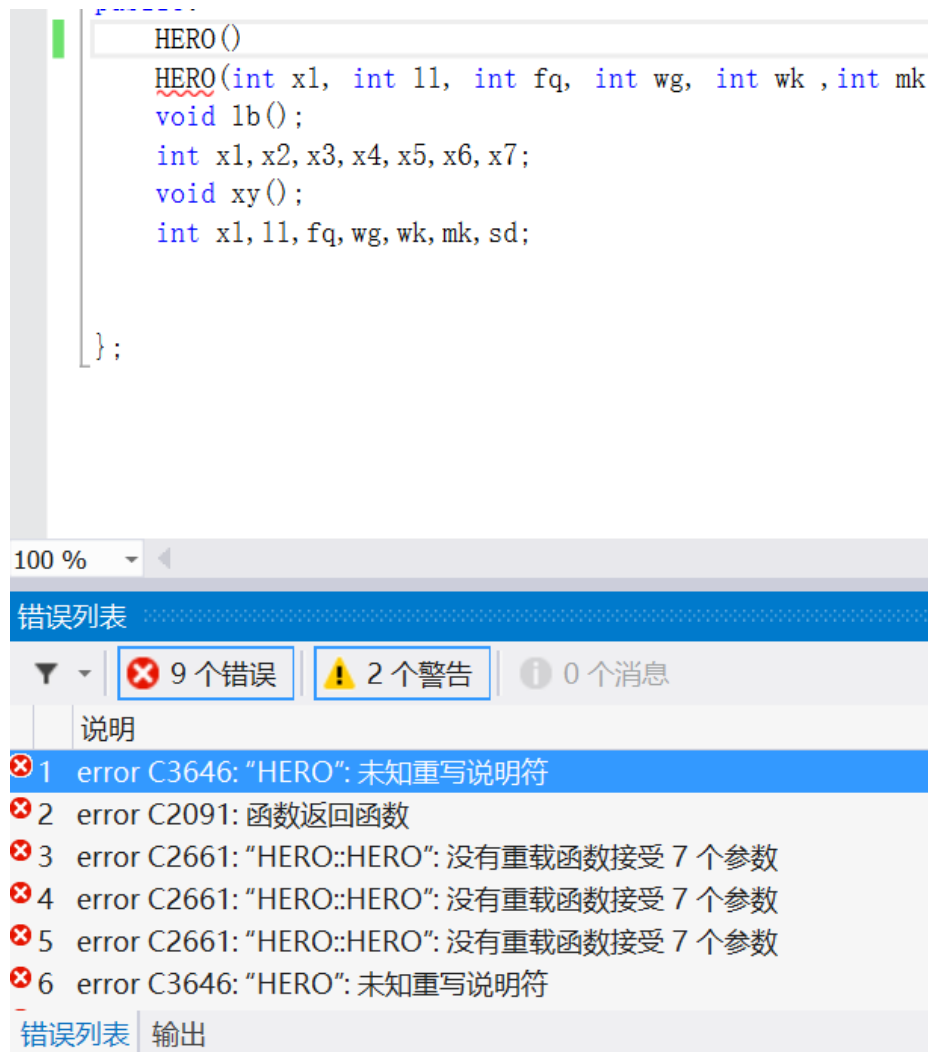


2、改 正



在 main.cpp 内的程序
#include"HERO.h"与 using namespace std; 之间加上
程序#include<conio.h>既可。#include<conio.h>表
示预处理命令, 表示程序包含 conio.h 库文件
conio.h 库文件定义了通过控制台进行数据输入和数
据输出的函数, 主要是一些用户通过按键盘产生的对应操作。比如常用的 getch()函数
就是在这个头文件里定义的

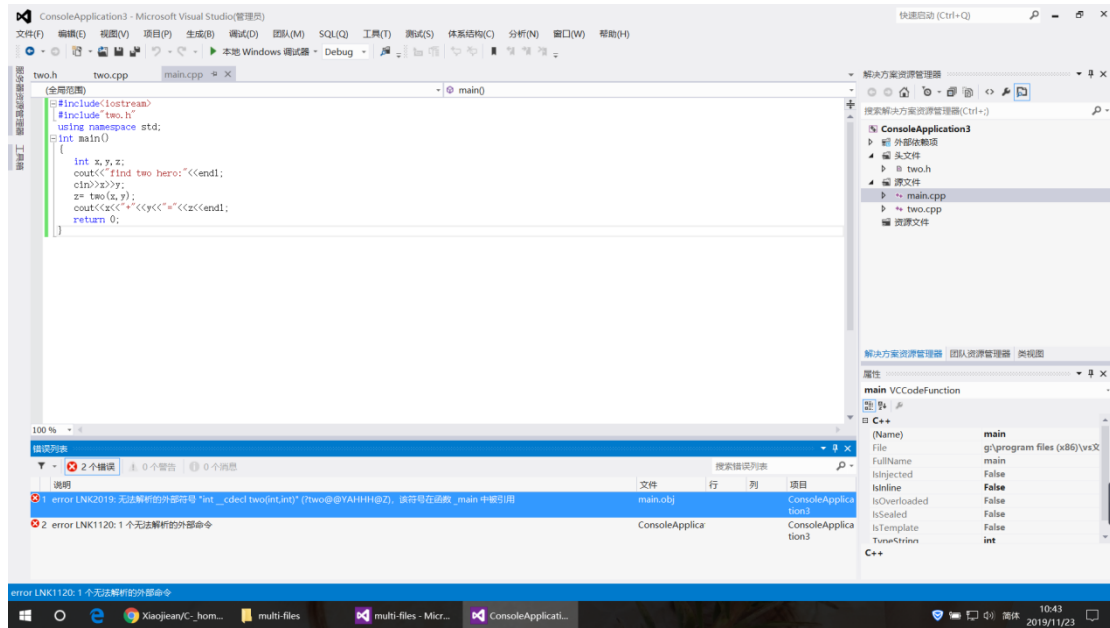
27 错误及分析



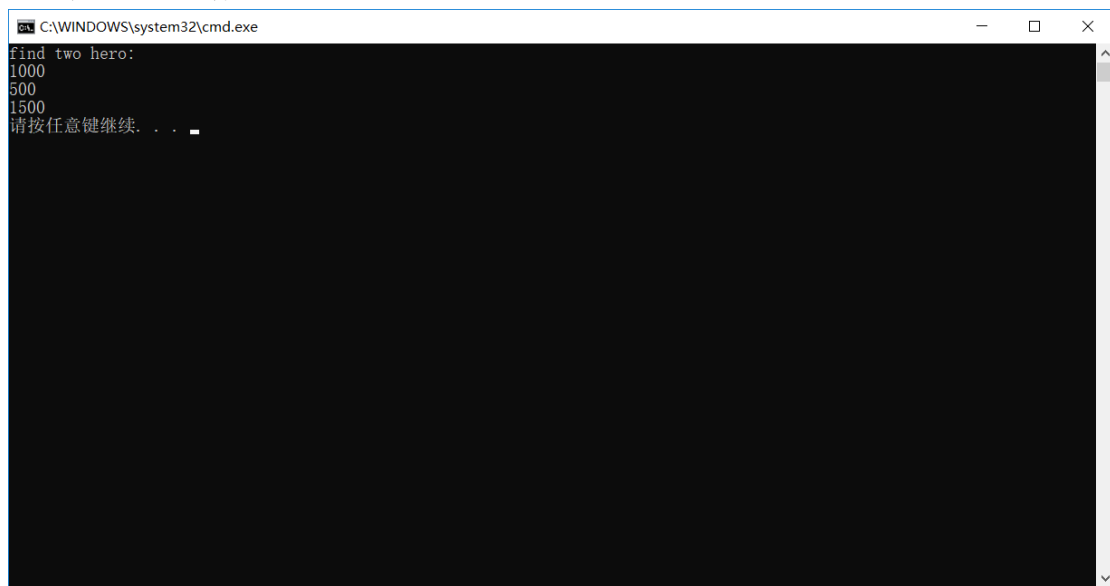
一开始她没有错误，后发现 HERO（）后面忘了加分号，单只许多错误。另把 x1 到 x7 全删了，把 x1 全替换为血量 xl,发现最后结果不对，不过我觉得应该可以改，只是我没改对。

28

产生的错误：本次实验遇到了很棘手的错误，多次尝试都没有解决。因为程序本身没有错误提示，显示的为外部函数的错误，这是以前从未出现的情况。错误提示如下图：



五：解决方法：由于多次尝试解决未果，所以只好重建项目，重新输入代码。所幸问题得到了解决。程序正确运行后显示如下图：



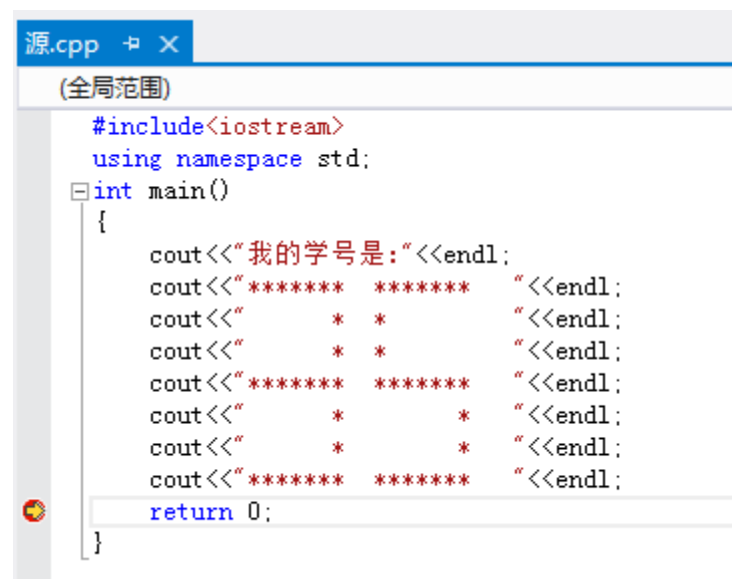
29

出现错误



最后一行的 cout 语句后面忘记了“;”;
程序运行完毕之后运行结果直接闪退，无法看见运行结果。

2、改正



```
源.cpp  X
(全局范围)

#include<iostream>
using namespace std;
int main()
{
    cout<<"我的学号是:"<<endl;
    cout<<"*****  *****" <<endl;
    cout<<"      *  *      " <<endl;
    cout<<"      *  *      " <<endl;
    cout<<"*****  *****" <<endl;
    cout<<"      *      *  " <<endl;
    cout<<"      *      *  " <<endl;
    cout<<"*****  *****" <<endl;
    return 0;
}
```

在 cout 后加“;”，程序即可运行；

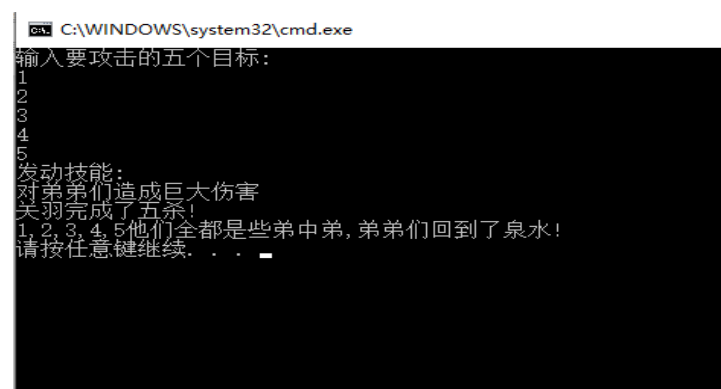
在 return0 前加上断点，是程序停止在 return0 之前，程序即可显示输出结果，点击继续，程序运行结果闪退消失。

问题：

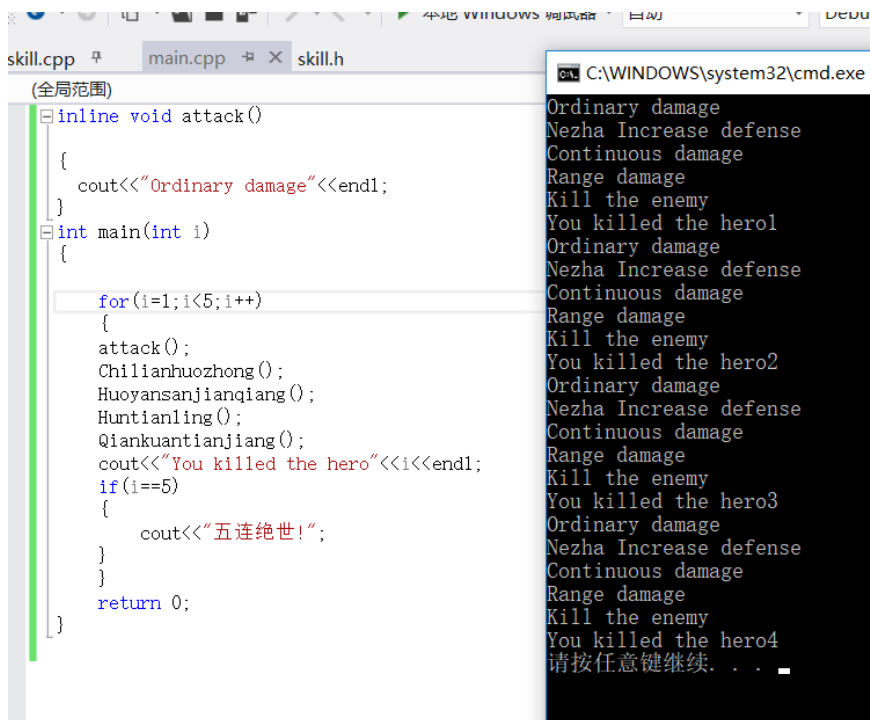
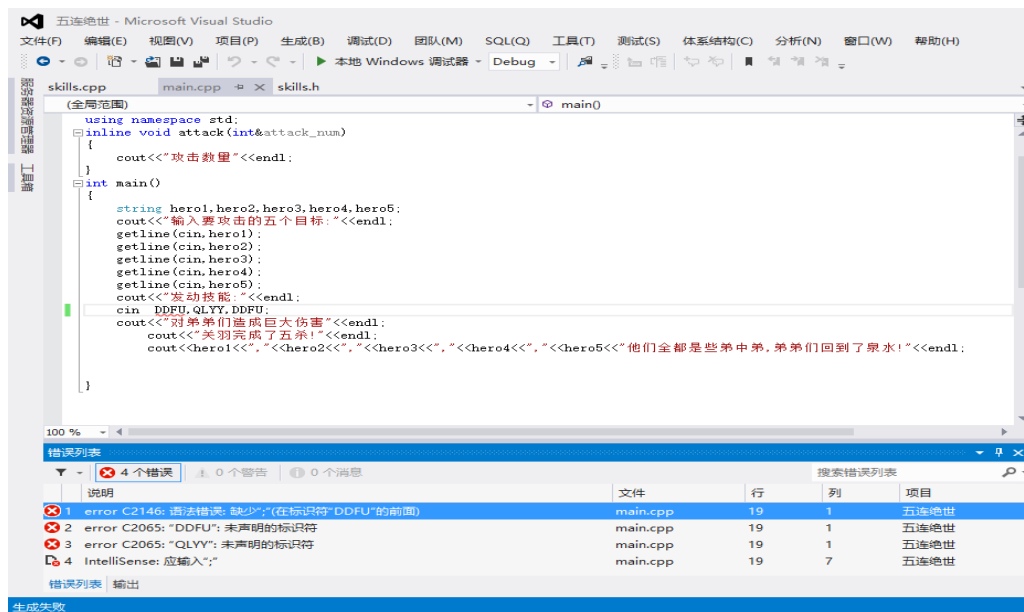
当时在差不多编写完这个程序时，试运行期间发现有个别定义变量的名称打错了，导致系统报错，无法正常运行；

30 程序执行时遇到的错误

本次实验所用到的程序因要用到内联函数所以相比于第一次实验来说较为复杂，该错误是在程序没错误的情况下输出结果无符合设想结果，输出结果如下图所示：



```
C:\WINDOWS\system32\cmd.exe
输入要攻击的五个目标：
1
2
3
4
5
发动技能：
对弟弟们造成巨大伤害
关羽完成了五杀！
1, 2, 3, 4, 5他们全都是些弟中弟，弟弟们回到了泉水！
请按任意键继续. . .
```



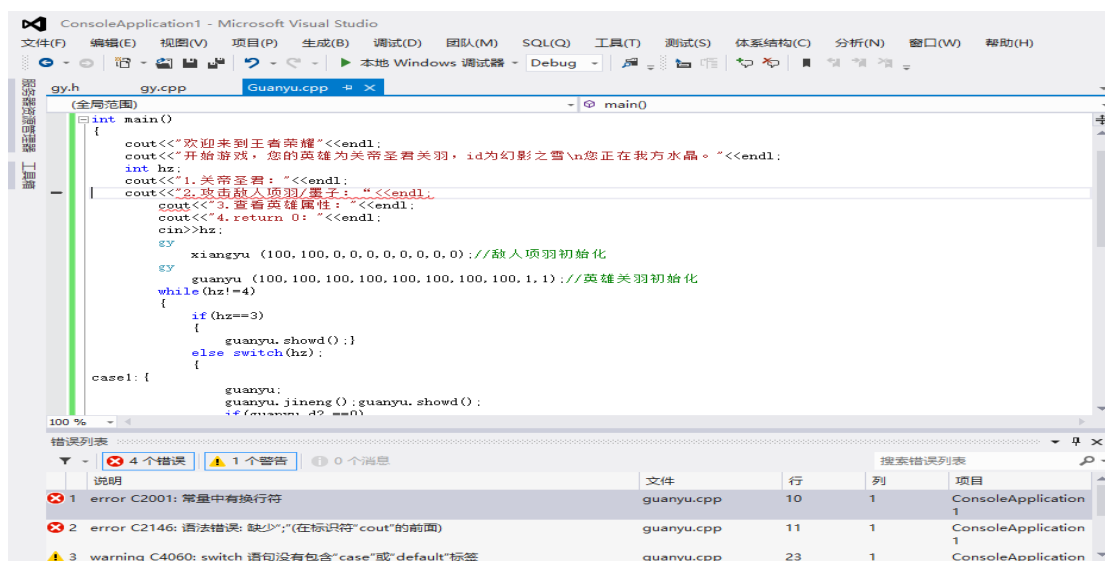
这个错误是无法输出五连绝世，错误就在 for 语句，应该为 (i<6) ,(i<5)只能输出四杀。

31

本次实验因涉及到“类”的运用，所以相对前两次实验来说较为复杂、程序较为繁琐，错误不好找。程序错误如下图所示：

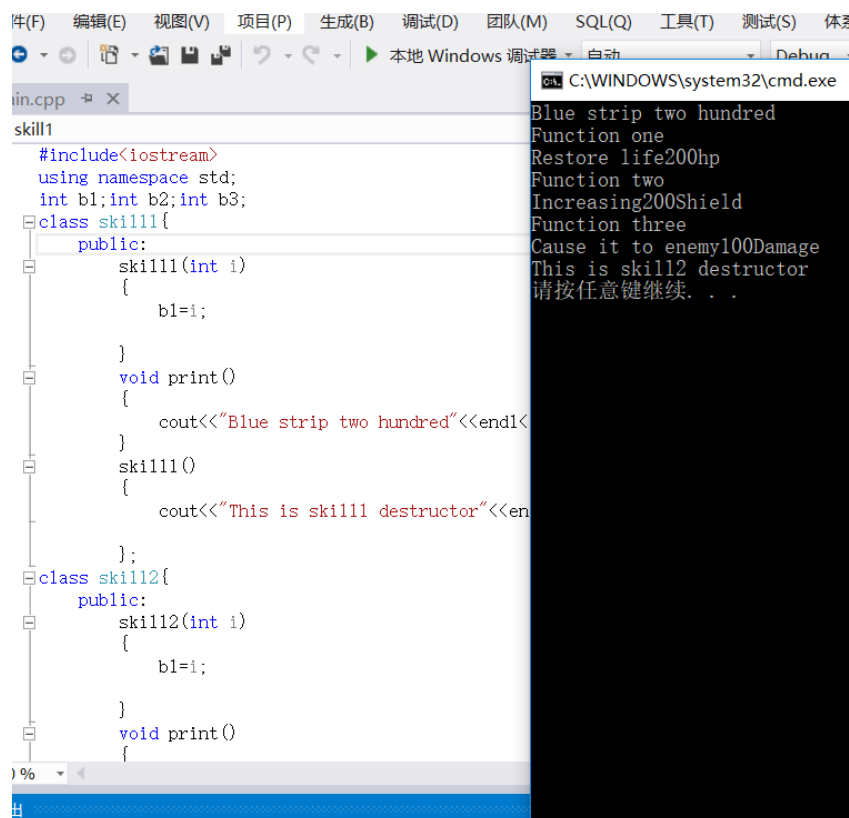
1、

3、



实验用到的软件 VS2012 是一款只识别英文标点符号的软件，该软件中的程序所用到的标点符号只能是英文的，中文标点无法识别，如果不小心输入中文标点则程序无法运行。

32 出现的错误



错误就是只输出了一个 skill2 的析构函数，skill1 的析构函数没有输出，原因就在 skill1() 的前面没有加“~”，还是忘记了之前说的波浪线。

在书写的过程中遇到啦一些小的错误和之前相比。例如；

```
b1=a*(i/(a+b));
cout<<"Hero 1 take damages as"<<a1<<endl<<"Hero 2 take damages as"<<b1<<endl
}
void main()
{
    cout<<"Enter shield values for hero 1 and hero 2 (0到100范围内) "
    cin>>a>>b;
    Error: 应输入";"
}
```

这是在书写的过程中忘记“ ; ”，一定要注重细节。

33

生成和调试的区别

(1)生成通常只是对工程进行编译，也就是说将工程编译成一个可以运行的形式。比如：exe文件。

(2)调试通常是假设工程存在错误，所以才以调试模式运行。在该模式下，可以对程序设置断点，可以查看变量在各个阶段的值变化，可以让程序进行逐步运行。以便更好地发现和修改程序的错误。

2.单引号和双引号区别

单引号里的内容表示是字符，双引号里的内容表示是字符串。

3.断点调试解决程序闪退问题、添加语句解决程序闪退问题、逐语句逐过程，跳出

常见问题：

程序运行秒退

法一：添加头文件

conio.h 头文件

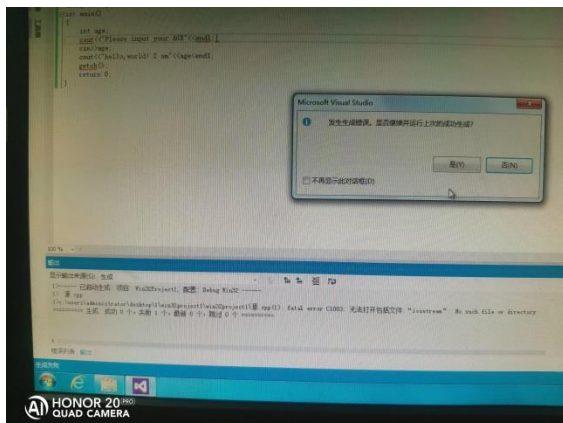
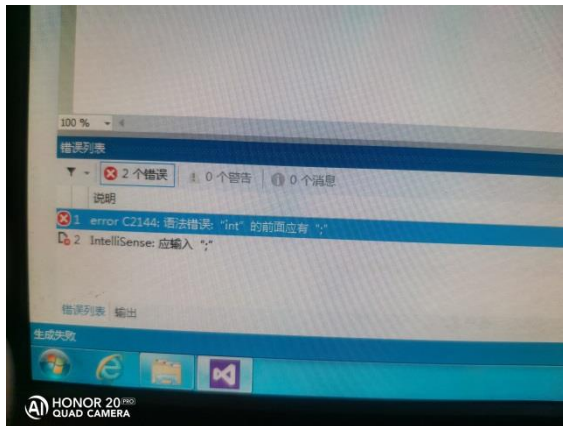
```
#include<iostream>
//#include<conio.h>
using namespace std;
int main()
{
    int age;
    cout<<"Please input you AGE"<<endl;
    cin>>age;
    cout<<"hello,world! I am "<<age<<endl;
    //getch();
    return 0;
}
```

一开始输入的程序里是没有这两句被注释掉的语句的，结果程序运行会发生闪退的情况，无法正常运行显示输出内容。

之后跟随老师的讲解我把这两句程序添加进去，然后闪退的问题就解决了。

getch()这个函数是一个不回显函数，当用户按下某个字符时，函数自动读取，无需按回车

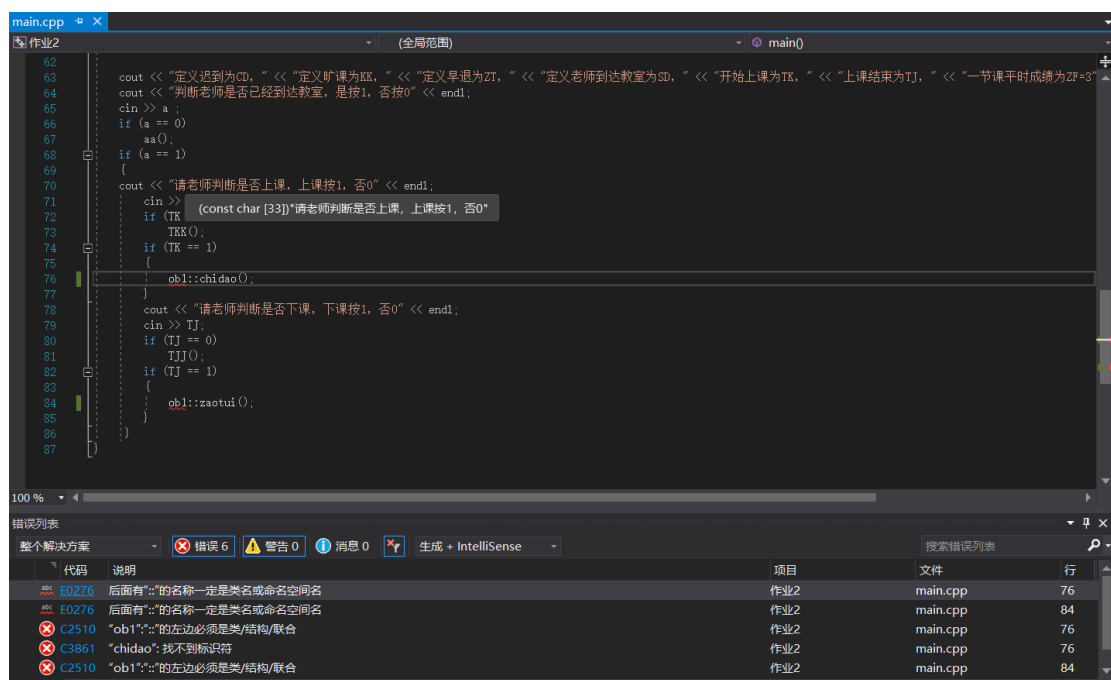
34 产生的错误



四 解决方法

通过查找 C++ 课本, 询问老师, 和同学讨论, 以及下面英文错误说明, 逐步修改, 最后完善程序

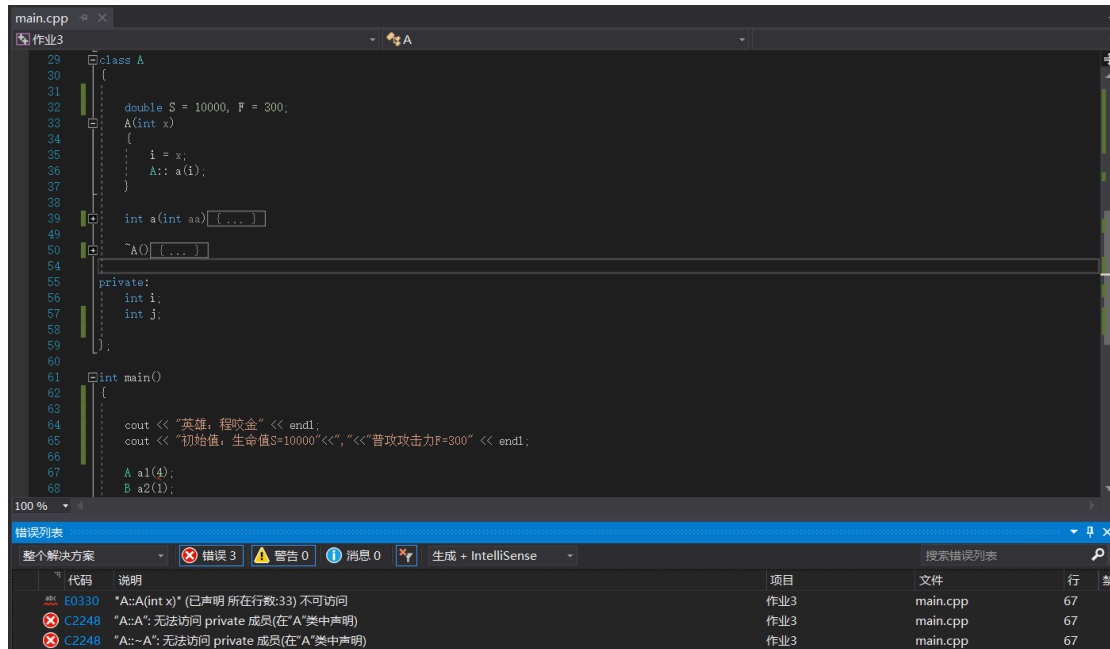
35 产生的代码错误和解决方法



错误: 1.用对象调用类中的格式写错了。

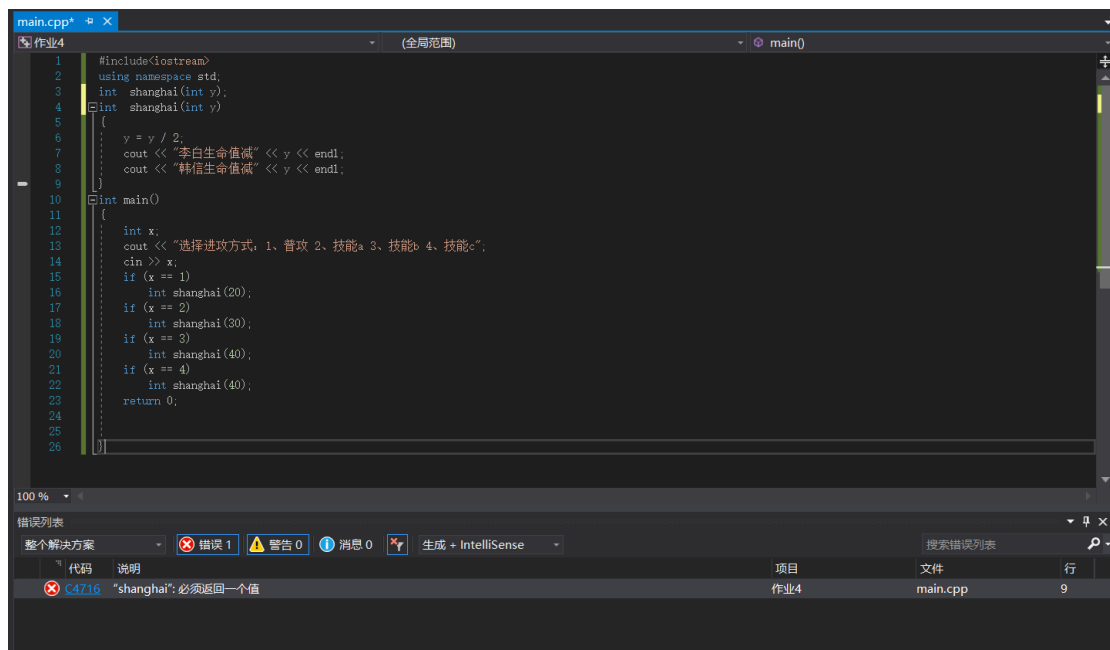
2.原来的程序, if 语句对是否上课, 是否下课, 老师是否来了, 只能处理等于 1 的情况, 一旦等于 0 就跳出了, 但实际应为 0 时应该返回等待, 之后在重新判断。
解决方法: 1.将“ob1:: chidao () ”和“ob1:: zaotui () ”改为“ob1.chidao () ”
“ob1.zaotui () ”。
2.加了一个这个函数之后解决了

36 产生的代码错误和解决方法



错误: 开始时没把 A (int x) 构造函数定义为 public 类型, 所以自动默认为私有的, 所以在主函数中就不能访问 A 类的构造函数。
解决方法: 在 31 行加了一个 public:

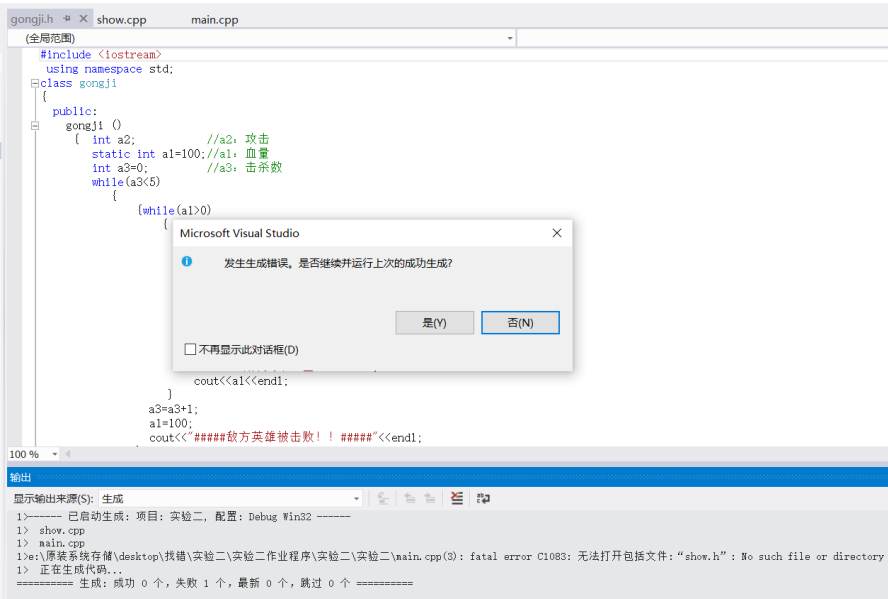
37 产生的代码错误和解决方法



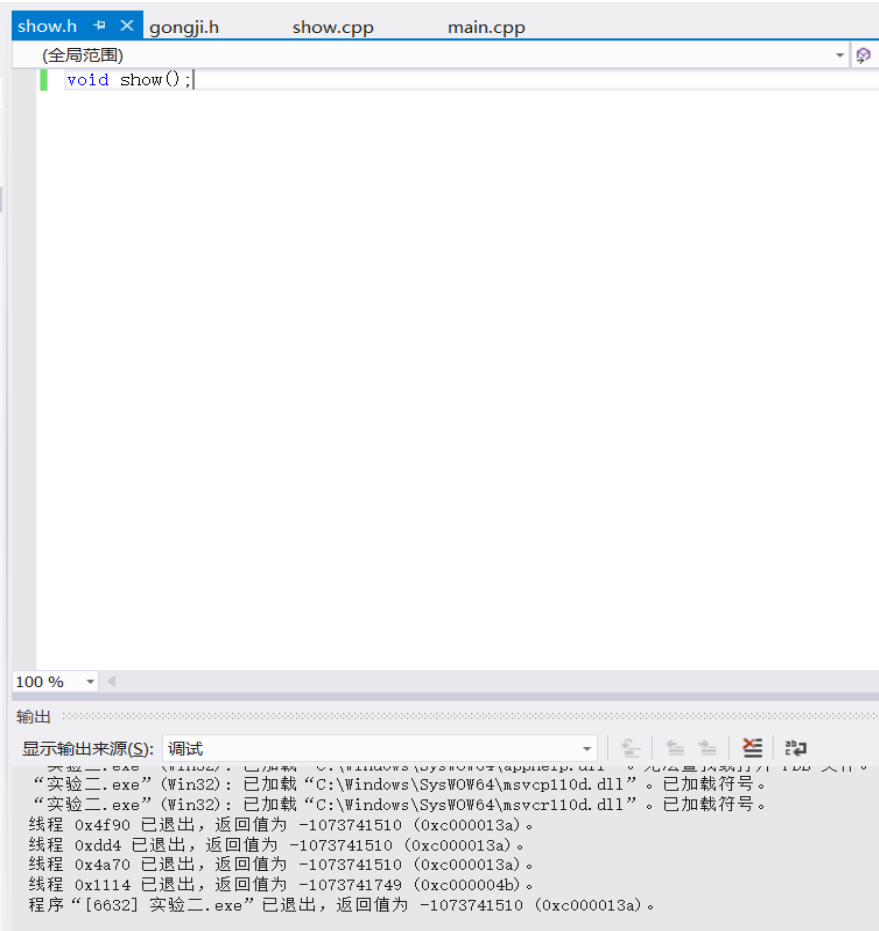
错误: 由于定义了“shanghai”函数的返回类型为 int 型, 是有返回值的, 所以最后应返回一个整数型的数, 但刚开始程序上未写返回值 return, 所以程序出现报错。

改进方法：在“shanghai”函数最后加了一个“return 0;”。

38 在本次调试过程中出现了程序无法正常编译的现象，如下图：



错误更改：根据提示，程序无法调用“show”函数，经过检查发现未建立“show”函数的头文件，更改如下：



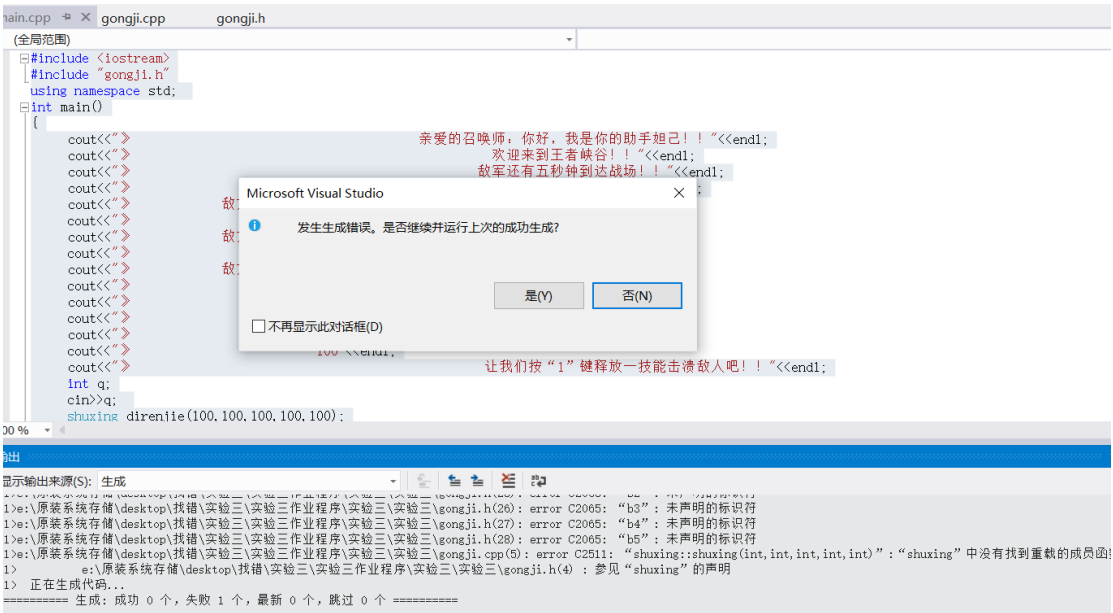
2、在程序运行后发现，击败一个英雄就会出现“五连绝世”的现象，现象如下：

程序调试中产生的错误

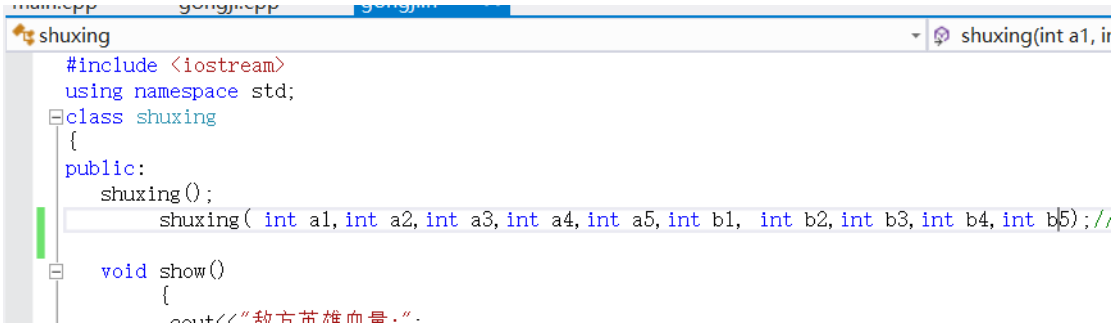
| | 说明 | 文件 | 行 |
|---|---------------------------------------------|----------|---|
| 1 | error C2146: 语法错误: 缺少 ";" (在标识符 "cout" 的前面) | main.cpp | 1 |
| 2 | IntelliSense: 应输入 ";" | main.cpp | 1 |

在此次的上机程序实验中，程序并不复杂，而且较上一次的实验，对软件的应用比较娴熟，因此并没有产生较大的问题，但是在输入时出现了分号遗漏的问题，好在问题并不难，而且比较好发现，容易修改。

40 程序错误：编译过程中发现程序无法编译，现象如下：



错误更改：通过系统提示发现 b 类参数未声明，通过检查发现将 b 定义在了构造函数中致使 b 无法被调用，我们将 b 放到构造函数外定义即可，错误如下：



改正程序如下：

```
main.cpp  gongji.cpp  gongji.h*  x
shuxing
#include <iostream>
using namespace std;
class shuxing
{
public:
    shuxing();
    shuxing( int a1,int a2,int a3,int a4,int a5); //血量: a1 魔抗: a2 物抗: a3
    int b1; int b2; int b3; int b4; int b5;
    void show()
    {
        cout<<"敌方英雄血量:";
        cout<<b1 <<endl;
        cout<<"敌方英雄魔抗:";
        cout<<b2 <<endl;
        cout<<"敌方英雄物抗:";
        cout<<b3<<endl;
        ... "敌方英雄..."
    }
}
```

41

程序调试中产生的错误

```
    }
    this->skill1 = rhs.skill1;
    this->skill2 = rhs.skill2;
    this->skill3 = rhs.skill3;
    return *this;
}

void HERO::showskill()
{
    cout <<skill1 << endl;
    cout <<skill2 << endl;
    cout <<skill3 << endl;
}
```

%

无列表

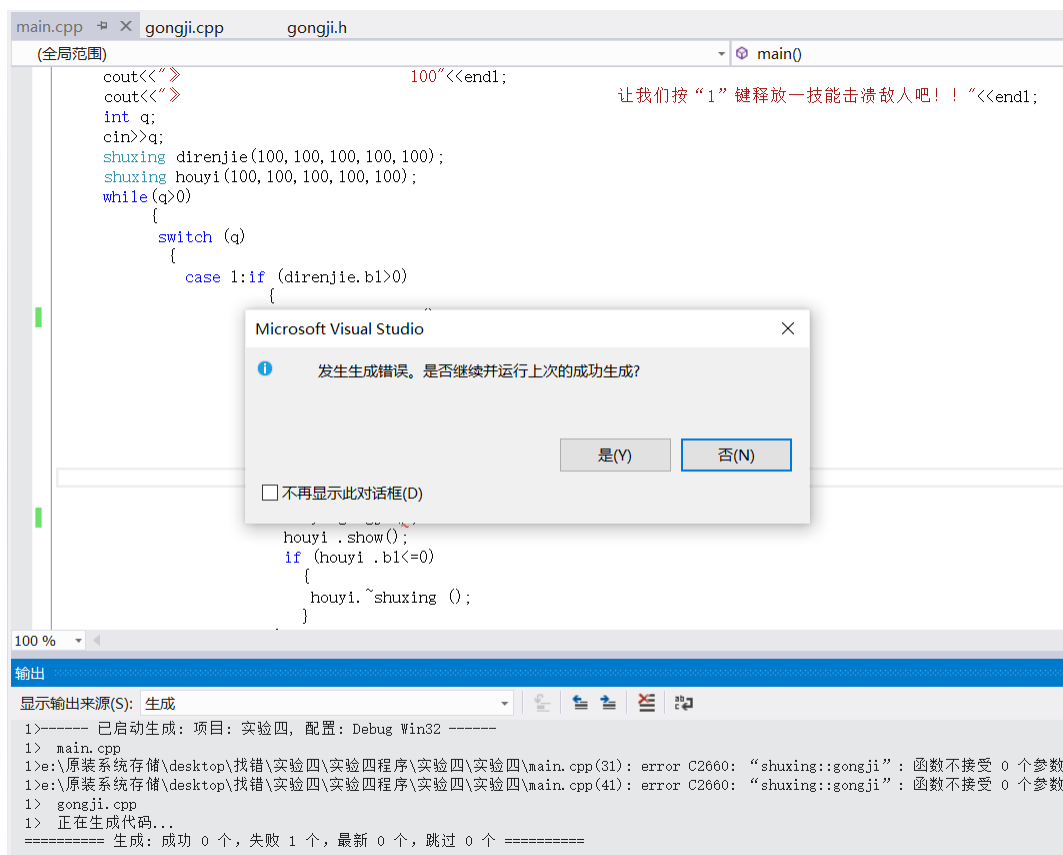
6 个错误 0 个警告 0 个消息

说明

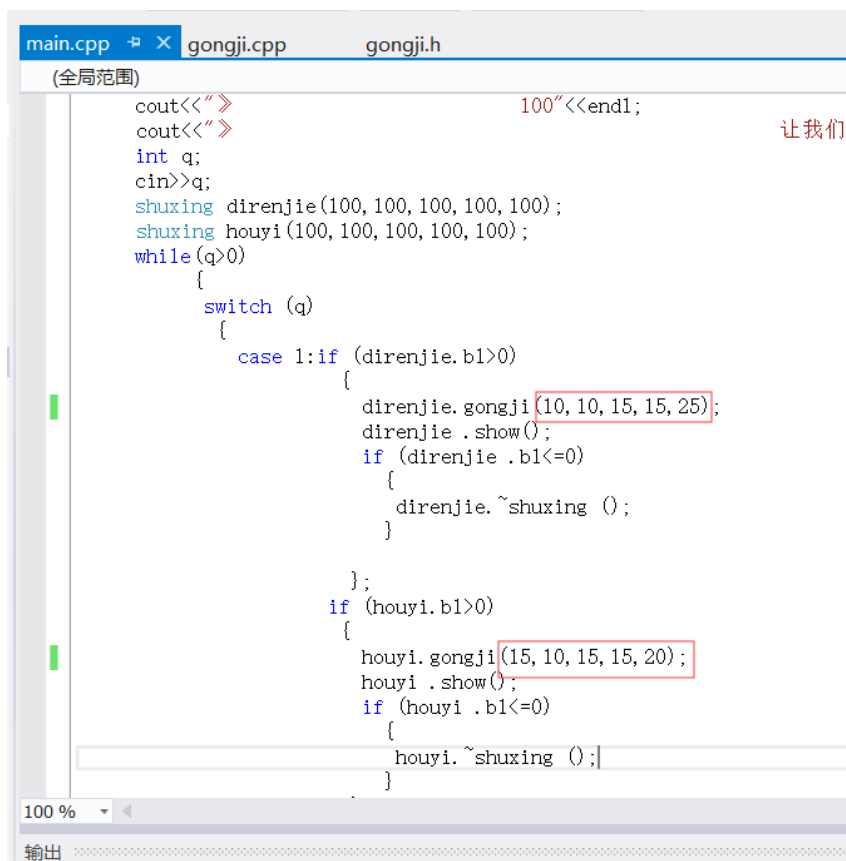
- 1 error C2440: "=" : 无法从 "const HERO" 转换为 "int"
- 2 error C2440: "=" : 无法从 "const HERO" 转换为 "int"
- 3 error C2440: "=" : 无法从 "const HERO" 转换为 "int"
- 4 IntelliSense: 不存在从 "const HERO" 到 "int" 的适当转换函数
- 5 IntelliSense: 不存在从 "const HERO" 到 "int" 的适当转换函数
- 6 IntelliSense: 不存在从 "const HERO" 到 "int" 的适当转换函数

在此次调试时，最初将如图红色波浪处后面的“.”误打为“，”导致了程序的调试失败，在输入时，如果出现此类错误那么 const HERO 中的值将无法成功地赋予 int，从而导致第一次调试出现错误。

42 程序错误：程序无法正常编译，现象如下：



错误更正: 经过观察系统提示发现,“gongji”函数不接受 0 个参数,是因为没有给“gongji”函数赋值, 只要对其进行赋值即可, 改正如下:



43 错误原因与更正

1. 由于 main 函数太长, {} 太多, 以至于后来程序无法正常运行
解决方法: 可用鼠标拖动选择“{” 可以在下面找到对应的 “}”, 以此类推可以找出哪个是不完整的。

2. 程序太过繁琐, case 定义的技能太多, 以至于经常编乱

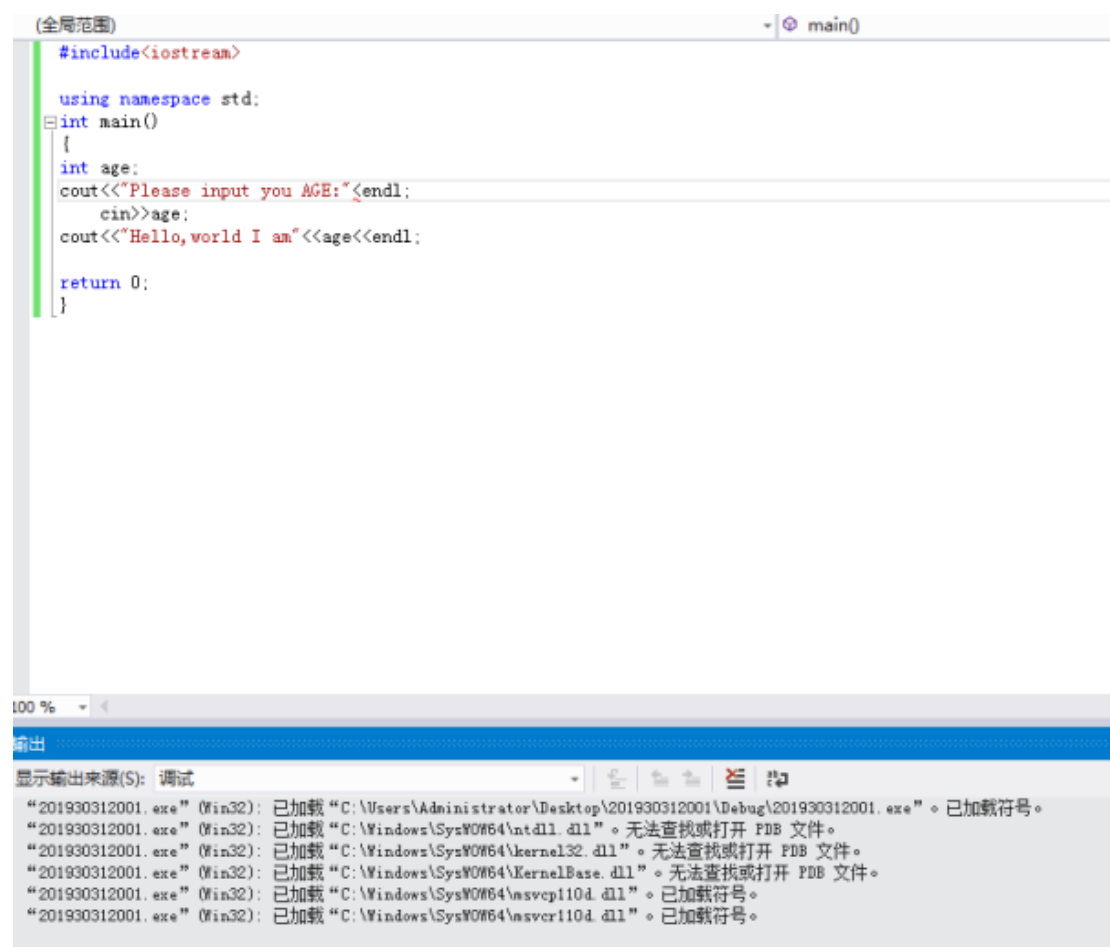
3.



分号没有标对位置

44 因为符号打错而导致程序调试失败

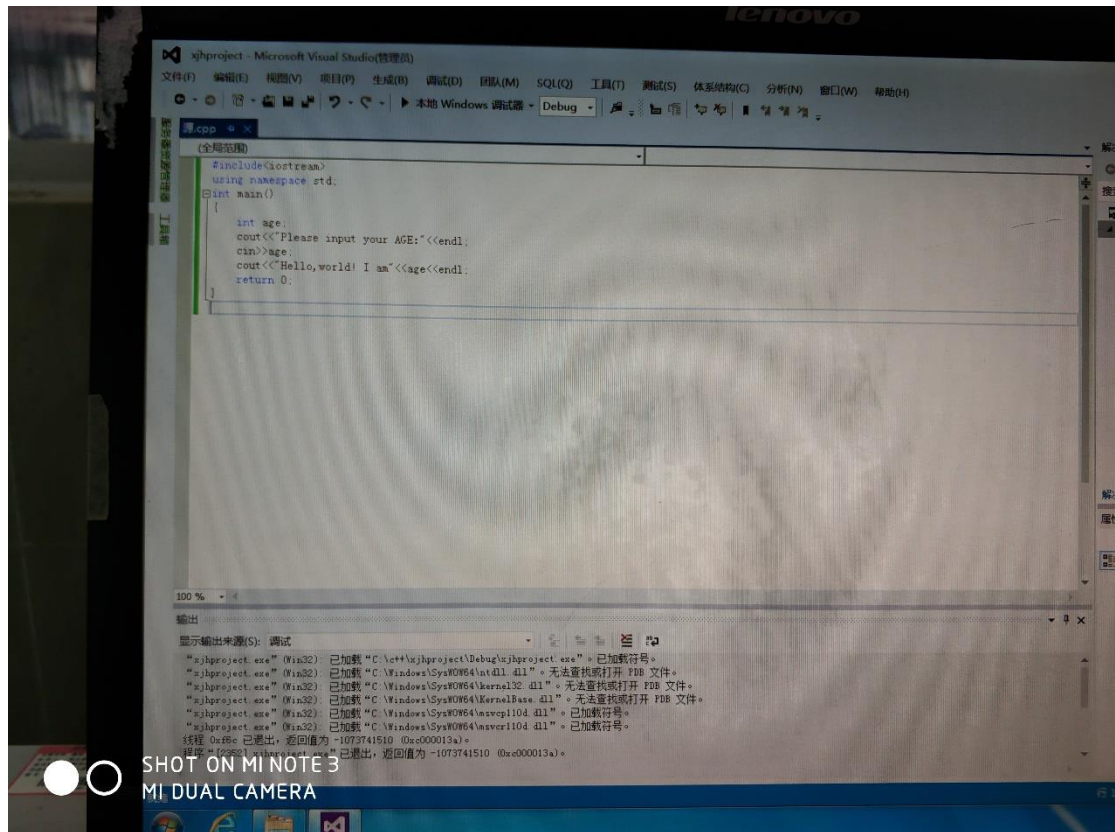
在调试时, 最开始出现了因为符号输入错误, 产生了程序调试失败的问题, 这很大程度上是粗心造成的, 这样的问题比较容易发现, 但不应该出现。



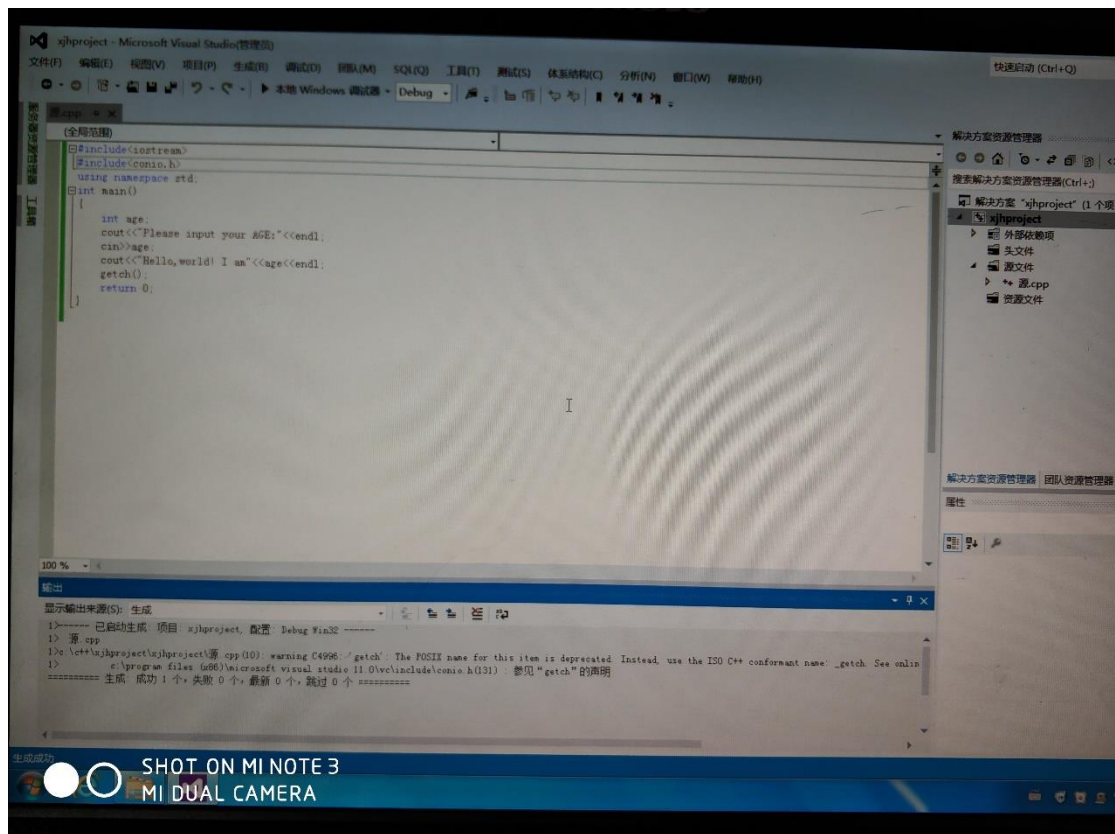
2、出现了程序显示闪退的问题

一开始调试程序，程序没有错误，但成功运行后出现了闪退的问题，直接点击启动调试，程序成功执行，但是在输入了年龄之后，直接闪退。可以使用“开始执行（不调试）”指令进行调试，这样就不会出现闪退的问题；或者直接加一个 `getchar ()` 的函数，使得系统在获得一个字符后才会退出；亦可以加一个断点，使得程序不进行到 `return 0;` 就不会使得系统退出。

产生代码错误及分析与改正

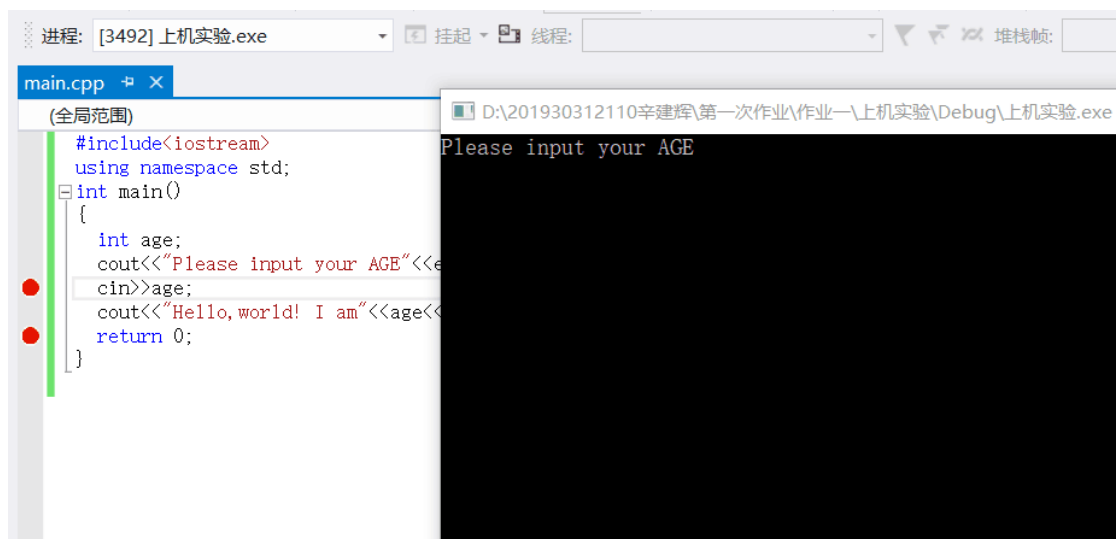


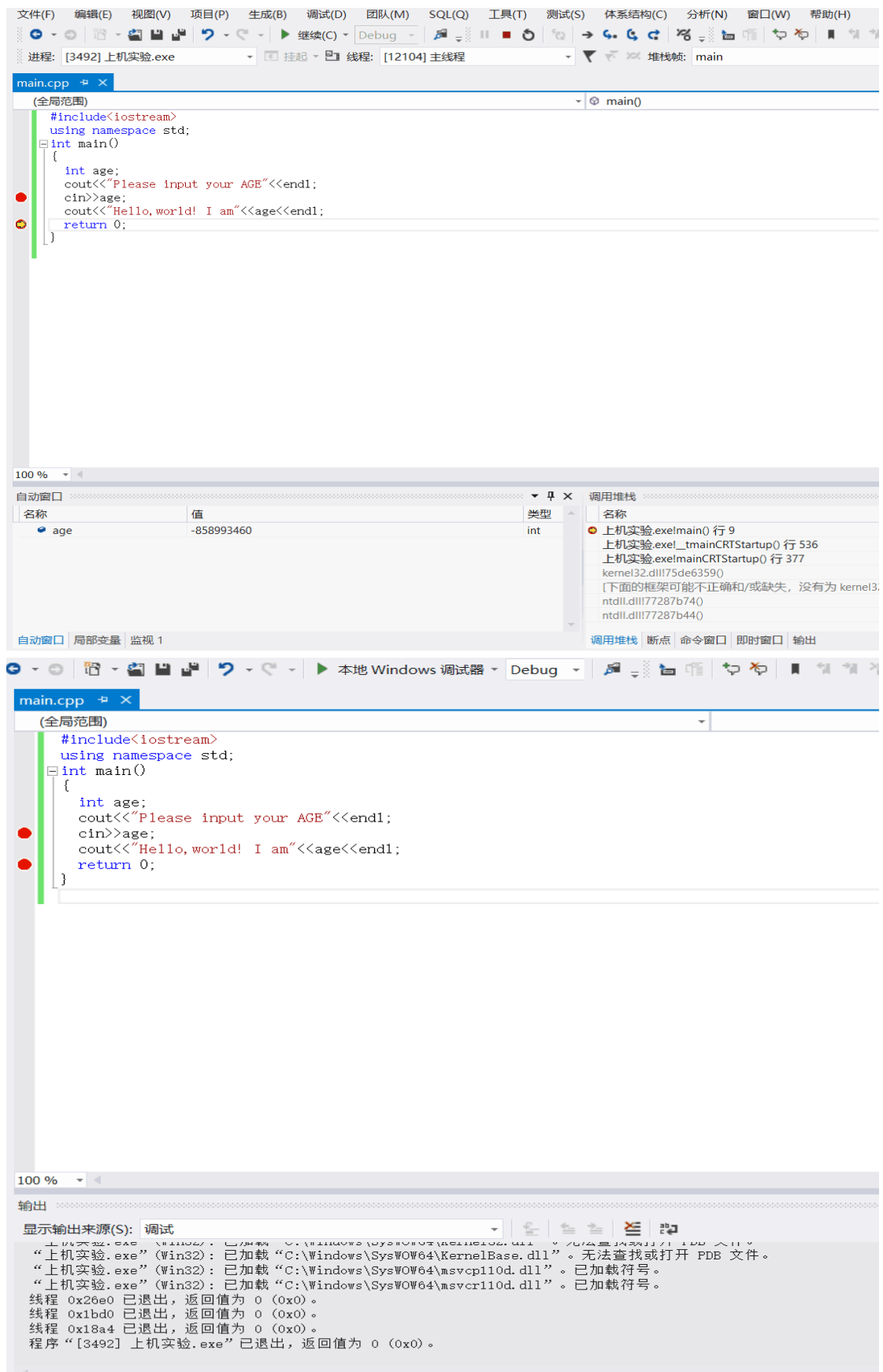
如图所示，输入这个程序会显示出错。经检查是未编译造成的，可以直接进行编译。这里我们对程序进行了修改，加入了程序“#include<conio.h>”和“getch();”。该程序就会自动编译，如图



现在就会成功。

为了更好的理解程序，我们学习了断点。





输入年龄 age, 还是会跳出来, 是因为程序在运行到断点处而被中断。由上图得出, 在继续执行后, 程序停留在 return 0, 我们可以更加了解程序运行过程。

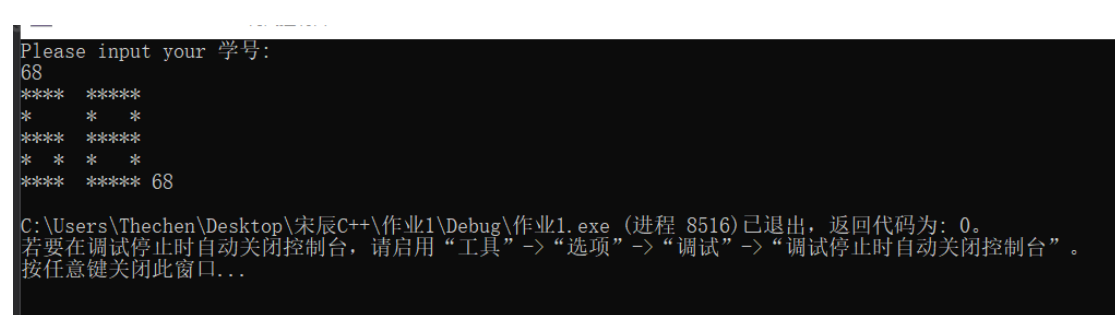
45 在输出的时候*号之间空格不好确定。

解决办法：

用多行 cout 可以解决对不好的问题

```
cout<<"* * *   * * *"<<endl;
cout<<"*           *   *"<<endl;
cout<<"* * *   * * *"<<endl;
cout<<"*   *   *   *"<<endl;
cout<<"* * *   * * *"<<endl;
```

截图：

A screenshot of a C++ program's output in a console window. The output shows a star pattern with five lines of stars, followed by the text "Please input your 学号:" and the number "68". The star pattern is as follows:
**** *
* * *
**** *
* * *
**** *
The text "Please input your 学号:" is followed by the number "68".
At the bottom of the console window, there is a message: "C:\Users\Thechen\Desktop\宋辰C++\作业1\Debug\作业1.exe (进程 8516) 已退出, 返回代码为: 0。若要在调试停止时自动关闭控制台, 请启用“工具”->“选项”->“调试”->“调试停止时自动关闭控制台”。按任意键关闭此窗口..."