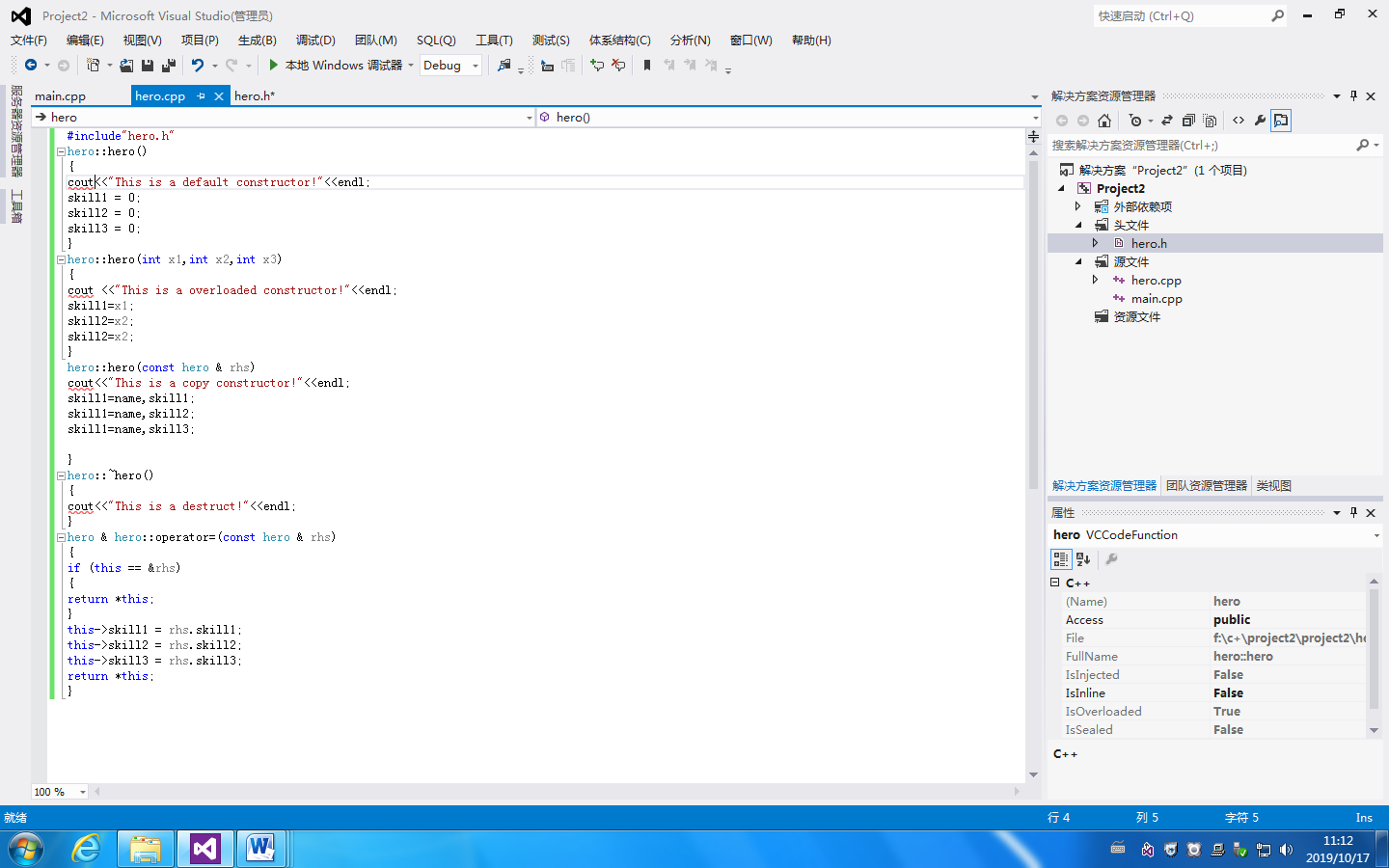
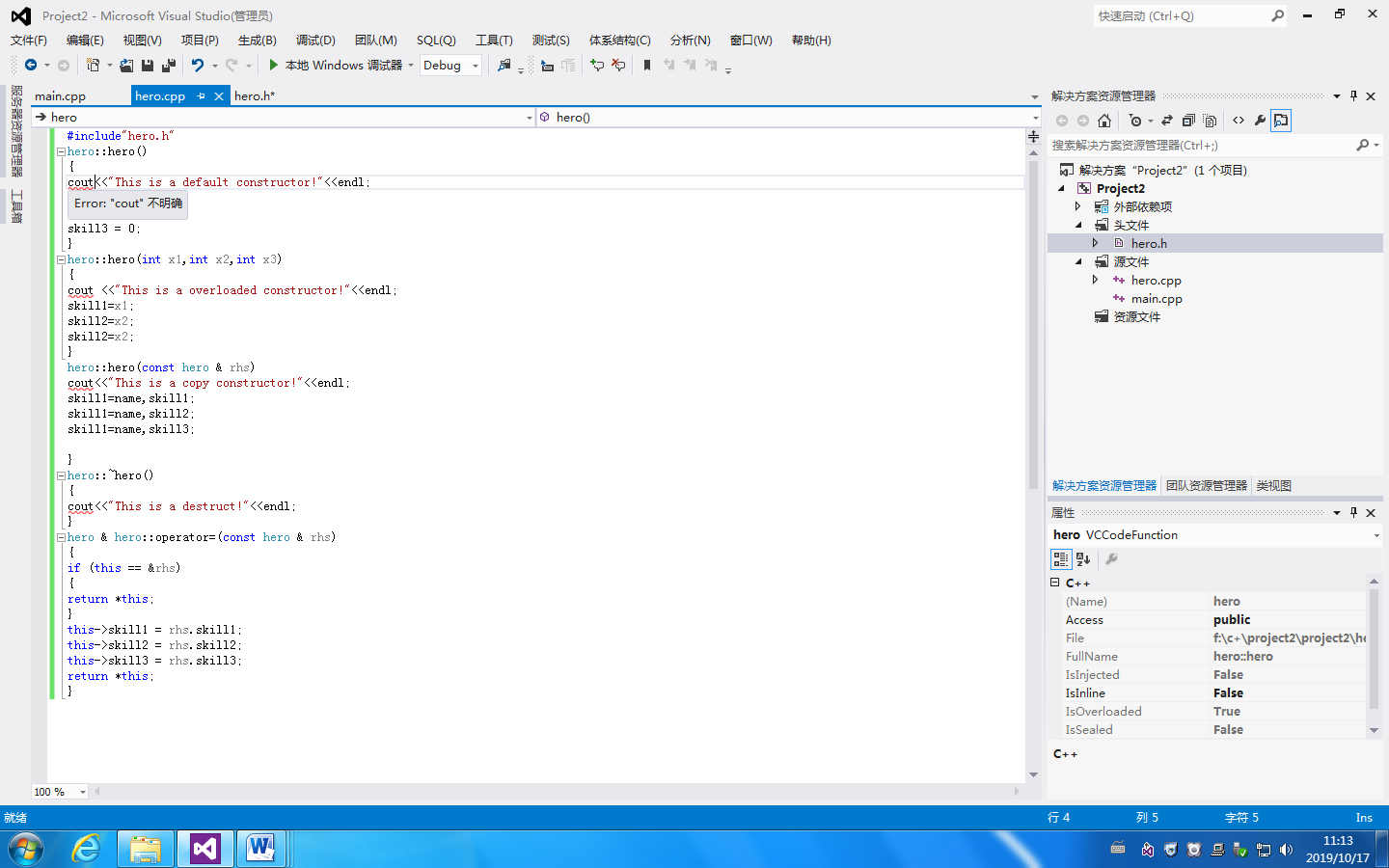
上机实验3练习出现的错误及解决方法

一、

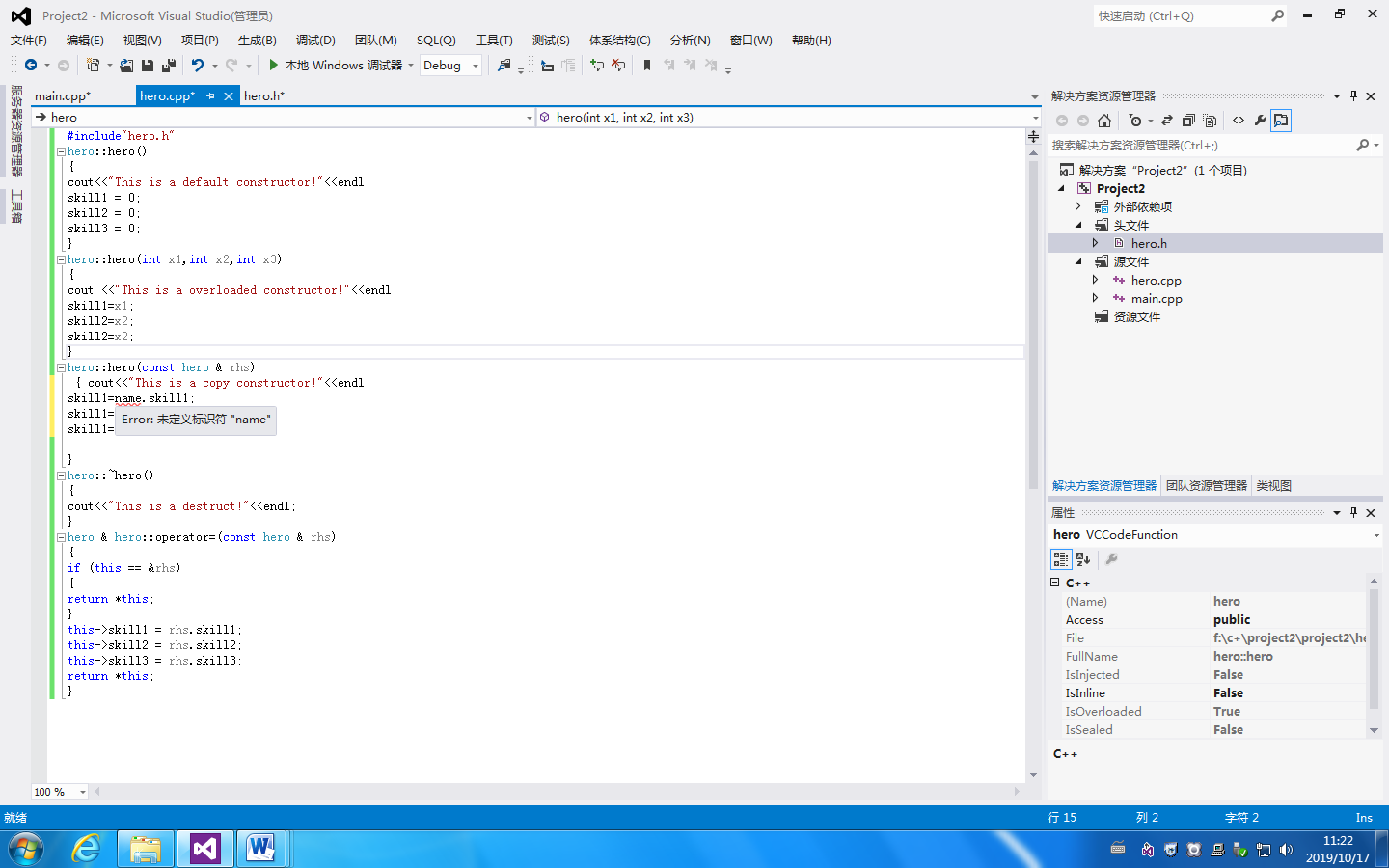


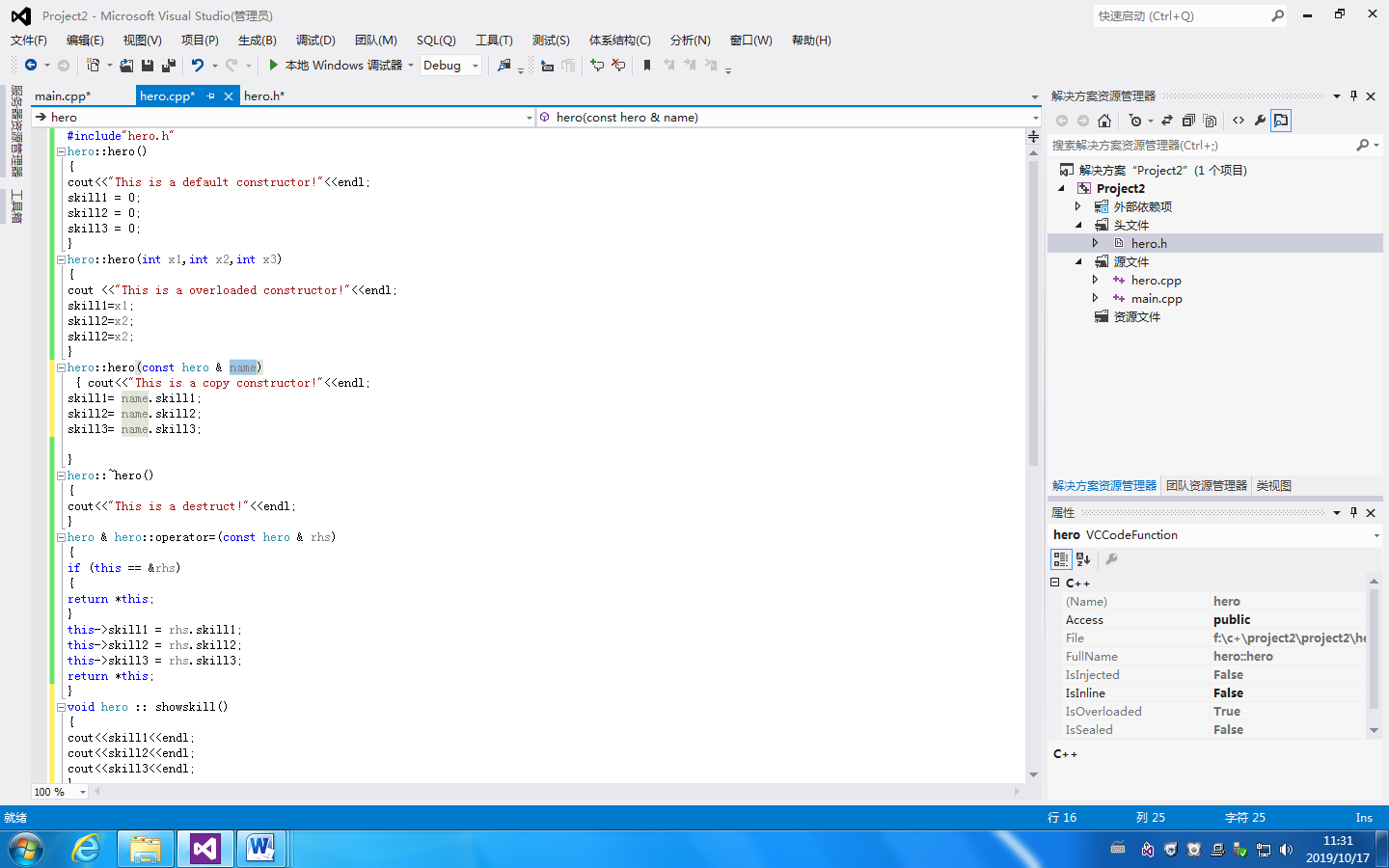


错误：cout 不明确

解决：原因是在头文件里重复定义了，将重复定义删除后，错误消失。

二、

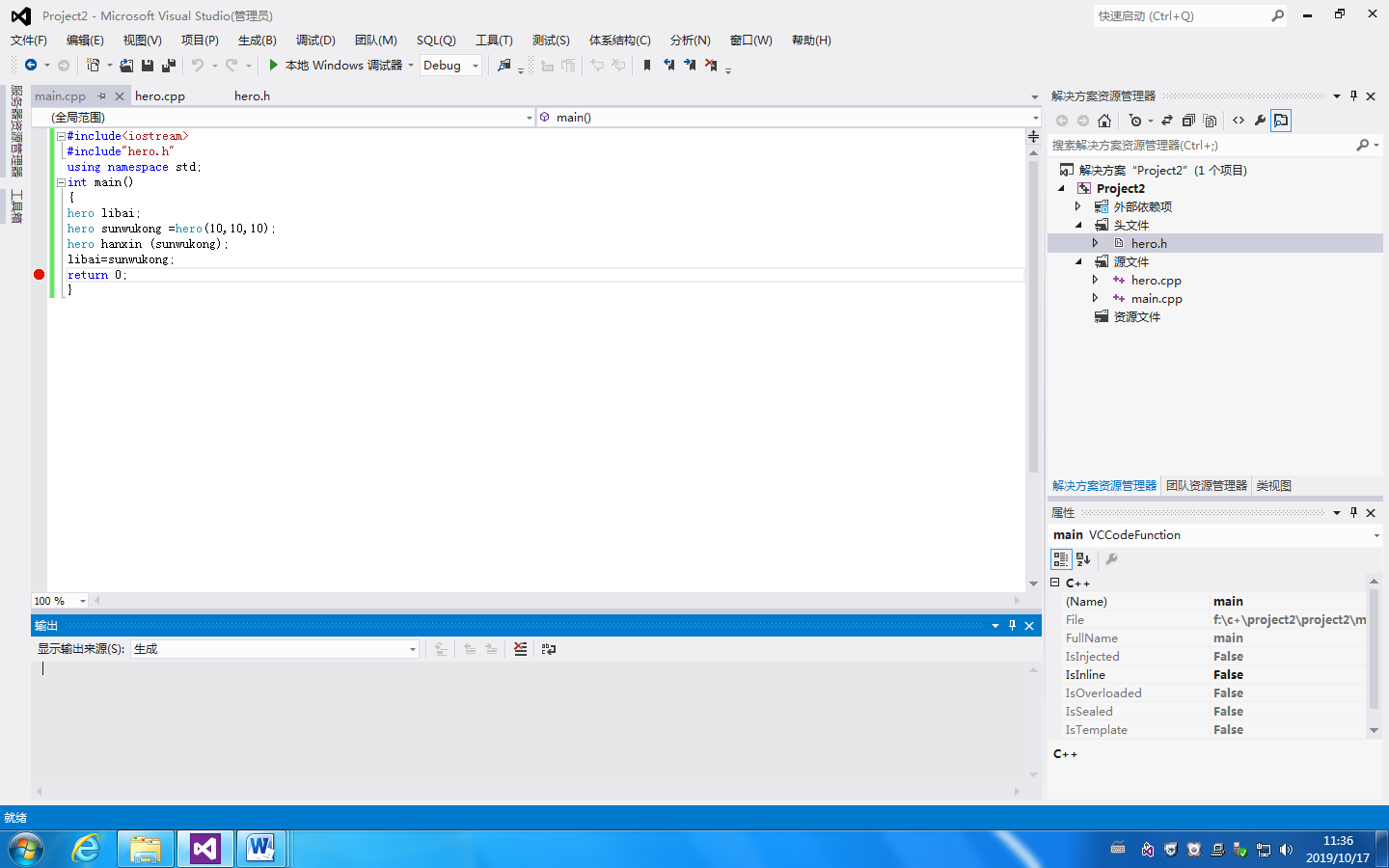


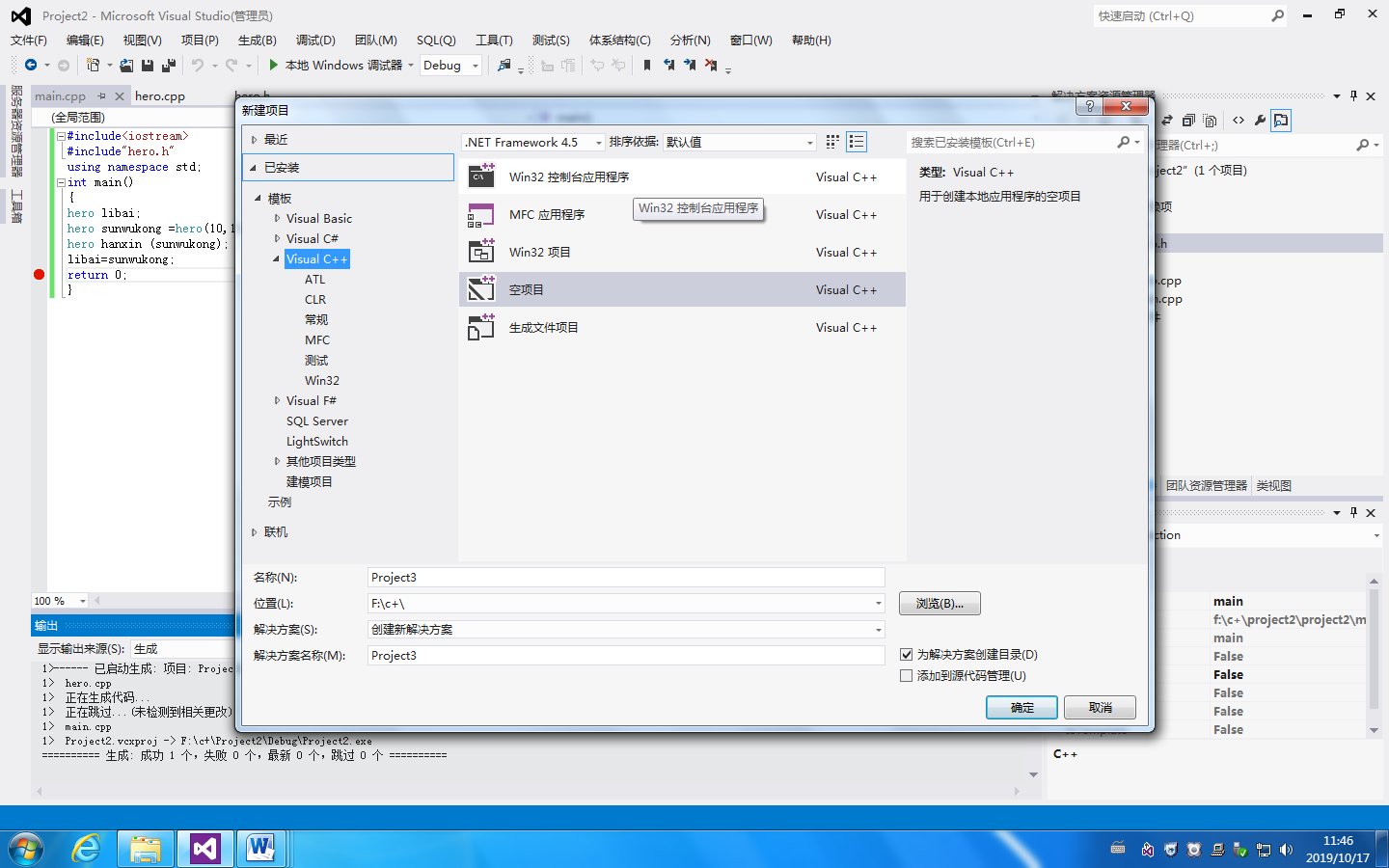
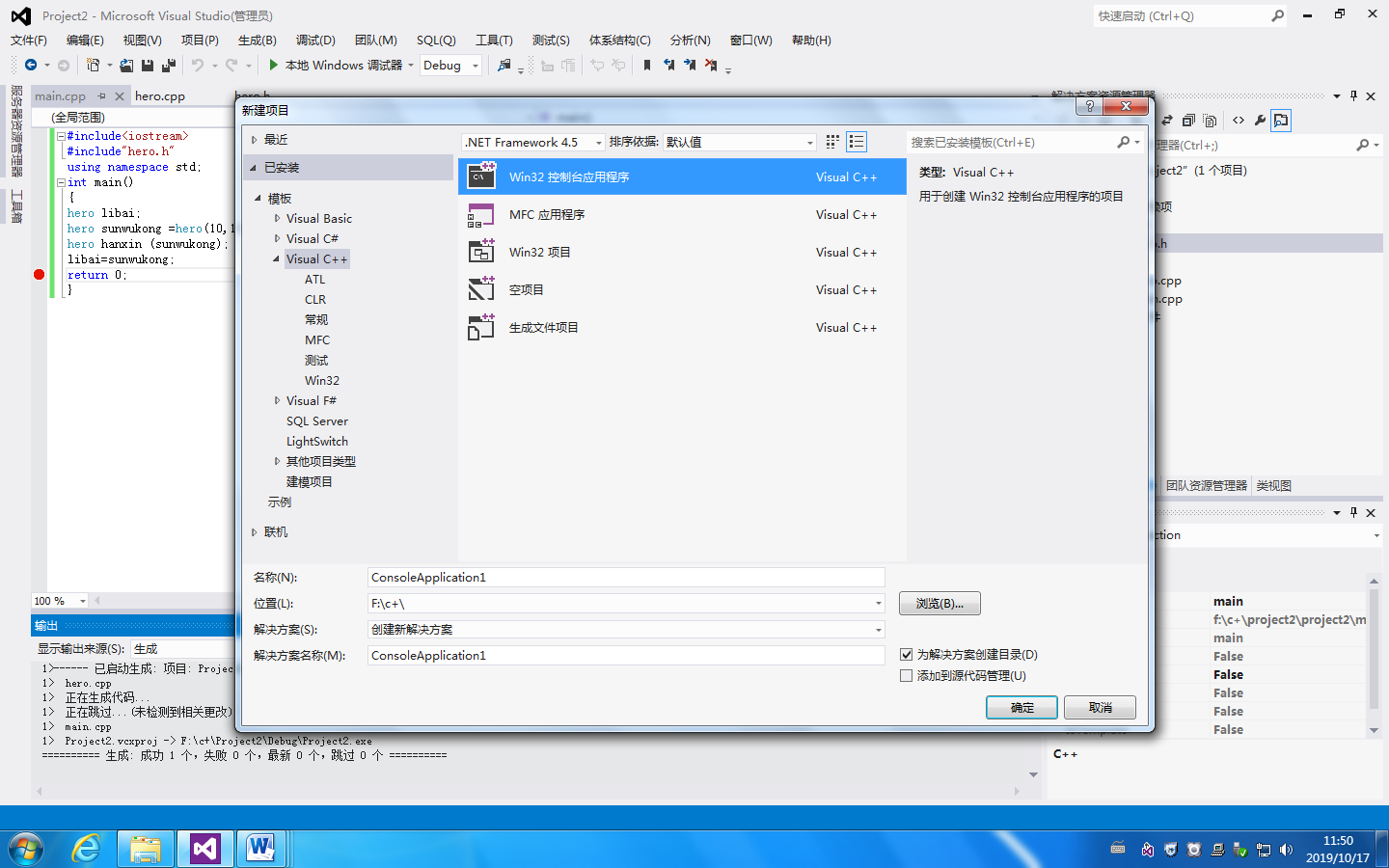


错误：未定义标识符“name”。

解决：原因是函数定义错误，应该为：“hero::hero(const hero & name)”

三、



错误：执行时结果出现闪退现象，增加断点或者“开始执行（不调试）”未能解决问题。

解决方法：错误原因是因为创建项目时选择了“Win32”，应该选择“Win32控制台应用程序”，改正后程序就能正常运行。

***上机作业：***在析构函数中显示这是哪一个对象的析构函数，实现一个技能的基本功能

1. 逻辑思路：编写程序时定义一个内联函数，功能是输出“I kill the hero.”用于表明英雄1血量为0时英雄1被杀。创建一个类HERO，在类的公有成员里定义一个构造函数用于输出“欢迎来到王者荣耀”欢迎语，然后定义1个普通函数在函数中加入一个FOR循环（释放一个技能），功能是定义英雄血量（100），并且实现血量递减，在血量为0是退出循环。以相同的方式定义五个功能相似的类，并在每一个类里分别自定义一个析构函数。

定义主函数。在主函数中分别创建5个类下面的一个对象A,B,C,D,E，并且分别调用五个类中的普通函数，释放技能完成五连绝世，同时输出析构函数中的内容。

2、源程序：

#include<iostream>

using namespace std;

inline int KILL()

{

cout<<"I kill the hero."<<endl;

return 0;}

class hero

{

public :

hero()

{

cout<<"欢迎来到王者荣耀"<<endl;

}

int hero1()

{

int blood;

for(blood=100;blood=0;)

{

blood = blood - 50;

return 0;

}

}

~hero()

{

KILL;};

};

class hero2

{

public :

int hero21()

{

int blood;

for(blood=100;blood=0;)

{

blood = blood - 50;

return 0;

}

}

~hero2()

{

KILL;};

};

class hero3

{

public :

int hero31()

{

int blood;

for(blood=100;blood=0;)

{

blood = blood - 50;

return 0;

}

}

~hero3()

{

KILL;};

};

class hero4

{

public :

int hero41()

{

int blood;

for(blood=100;blood=0;)

{

blood = blood - 50;

return 0;

}

}

~hero4()

{

KILL;};

};

class hero5

{

public :

int hero51()

{

int blood;

for(blood=100;blood=0;)

{

blood = blood - 50;

return 0;

}

}

~hero5()

{

KILL;};

};

int main()

{

hero A;

hero2 B;

hero3 C;

hero4 D;

hero5 E ;

A.hero1();

cout<<"hero1析构函数1"<<KILL()<<endl;

B.hero21();

cout<<" hero2析构函数2"<<KILL()<<endl;

C.hero31();

cout<<"hero3 析构函数3"<<KILL()<<endl;

D.hero41();

cout<<"hero4析构函数4"<<KILL()<<endl;

E.hero51();

cout<<"hero5析构函数5"<<KILL()<<endl;

return 0;

}