**C++第三次实验报告**

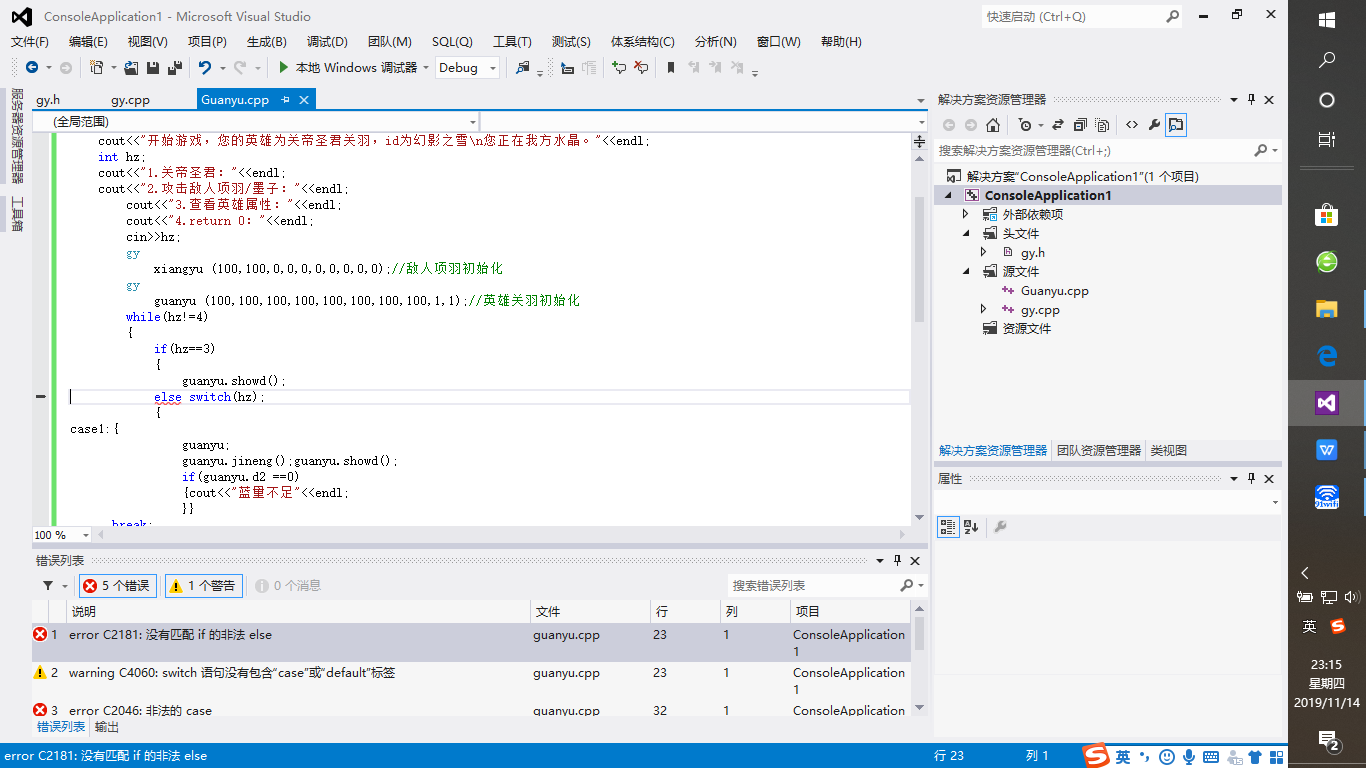
**1.设计程序的思路**

本次实验用到了C++中“类”这方面的知识，“类”是 C++ 的核心特性，通常被称为用户定义的类型。“类”用于指定对象的形式，它包含了数据表示法和用于处理数据的方法。类中的数据和方法称为类的成员。函数在一个类中被称为类的成员。本次程序中涉及到了“类”中的默认构造函数，重载构造函数，复制构造函数，析构函数，成员函数。

1. **程序执行时遇到的错误**

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、



else语句出现了错误，导致程序无法运行

应在if(hz==3)

{

guanyu.showd();后加 }

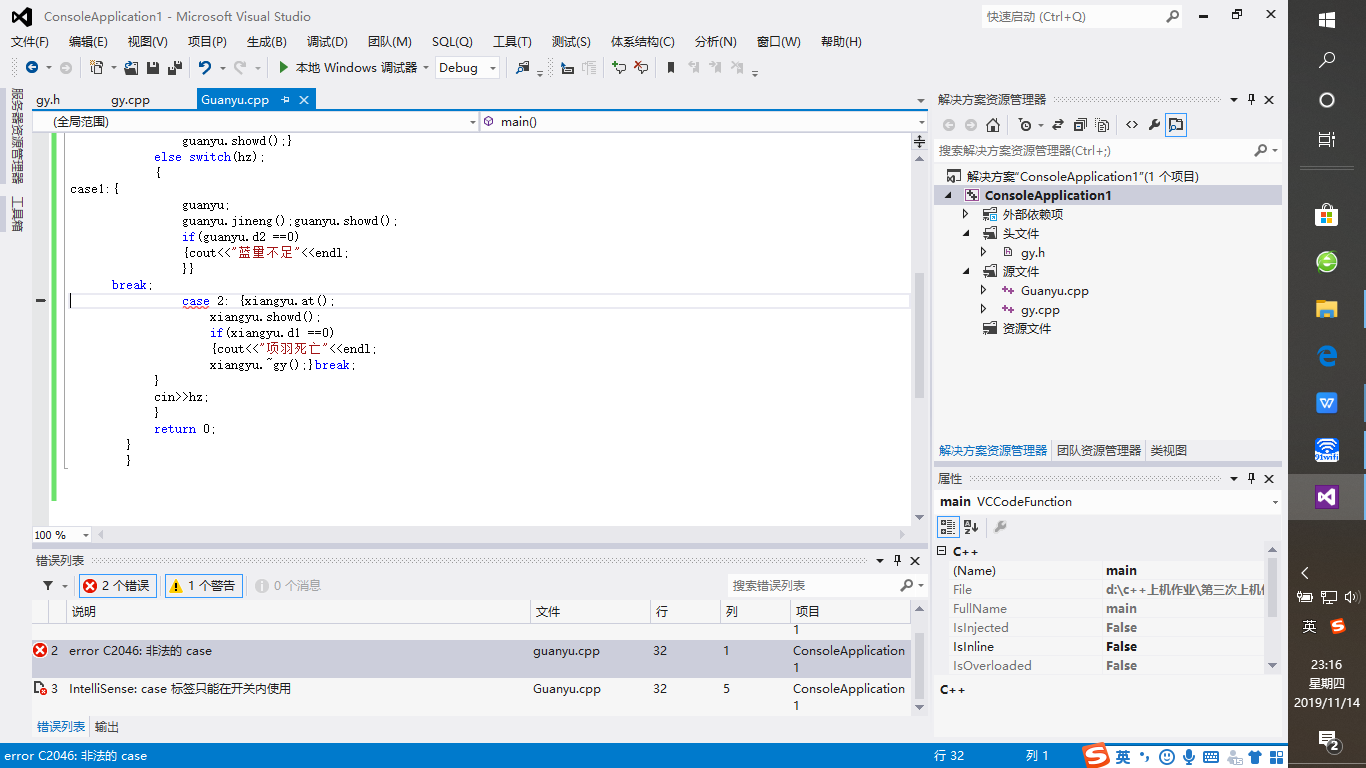
如果不加 } 则表示该语句没有结束，else语句也包含在了这段语句中

正确程序应为if(hz==3)

{

guanyu.showd();}

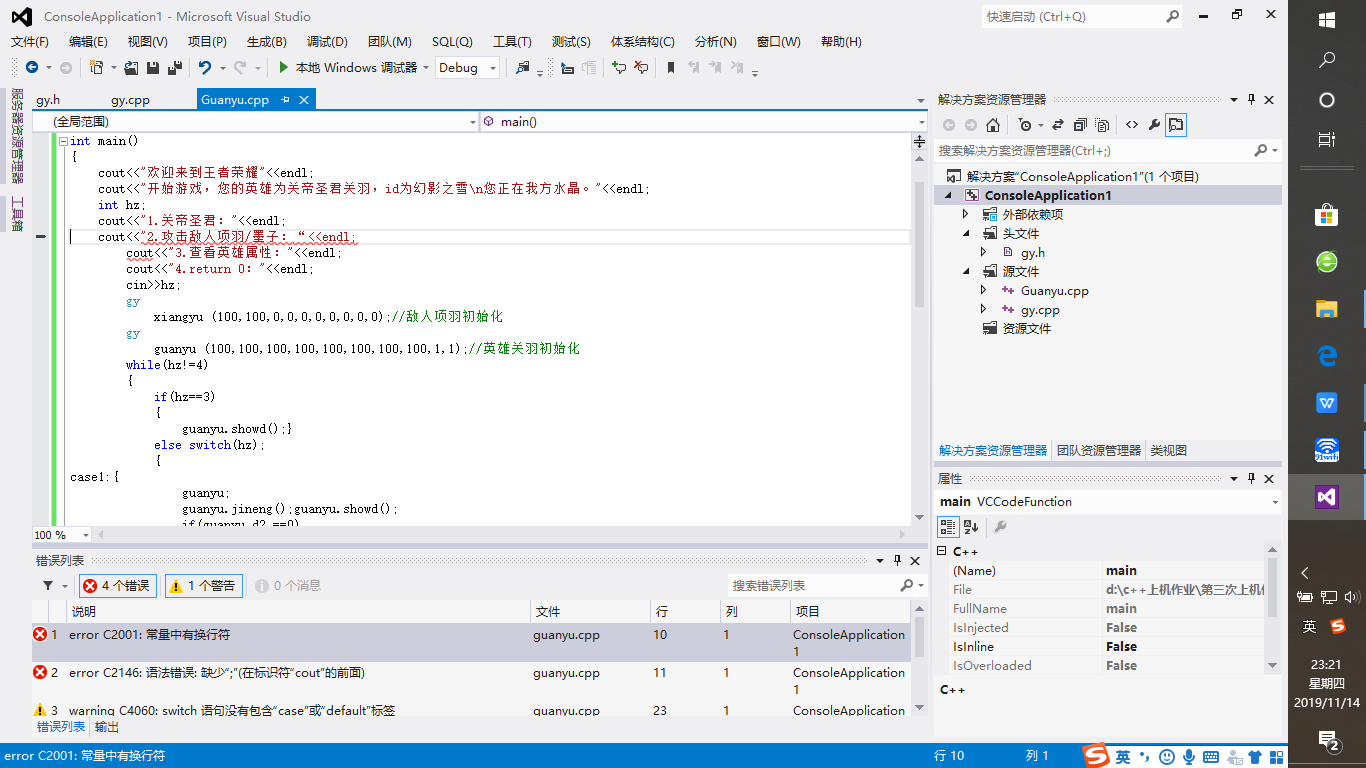
2、



这个错误没什么好说的，纯粹的程序格式错误，下次注意

应将case 2之间空格去掉，正确程序应为case2

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。

应将 cout<<"2.攻击敌人项羽/墨子：“<<endl; 之中的“改为" , 正确程序应为cout<<"2.攻击敌人项羽/墨子："<<endl;

1. **程序的主体代码**

**gy.h**

#include<iostream>

using namespace std;

class HERO

{

public:

HERO(); //默认构造函数

HERO(int x1,int x2,int x3); //重载构造函数

HERO(const HERO & name); //复制构造函数

~HERO(); //析构函数

HERO &operator=(const HERO &rhs);

void showskill();

private:

int skill1;

int skill2;

int skill3;

};

**gy.cpp**

#include"gy.h"

HERO::HERO()

{

cout<<"This is a default constructor!"<<endl;

skill1 = 0;

skill2 = 0;

skill3 = 0;

}

HERO::HERO(int x1,int x2,int x3)

{

cout<<"This is a overloaded constructor!"<<endl;

skill1 = x1;

skill2 = x2;

skill3 = x3;

}

HERO::HERO(const HERO & name)

{

cout<<"This is a copy constructor!"<<endl;

skill1 = name.skill1;

skill2 = name.skill2;

skill3 = name.skill3;

}

HERO::~HERO()

{

cout<<"This is a destructor!"<<endl;

}

HERO & HERO::operator=(const HERO &rhs)

{

if( this == &rhs)

{

return \*this;

}

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;

}

**guanyu.cpp**

#include<iostream>

#include"gy.h"

using namespace std;

int main()

HERO guanyu;

cout<<"guanyu skills:"<<endl;

libai.showskill();

HERO mozi = HERO(10, 10, 10);

cout<<"guanyu skills:"<<endl;

sunwukong.showskill();

HERO xiangyu(mozi);

cout<<"xiangyu skills:"<<endl;

xiangyu.showskill();

guanyu = mozi;

cout<<"guanyu skills:"<<endl;

guanyu.showskill();

return 0;

}