**C++实验报告三： 类和对象的学习**

**刘振宇 自18G2 201830312021**

**思路：**

这节课黄老师给我们讲了：类和对象：**类**（模板）的声明和定义对象、**构造函数**和**析构函数**实现方法。这次上机实验黄老师通过领着我们构建一个简单的王者荣耀英雄的类，经过这次上机课我对于类的构建，以及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，成员函数有了初步的了解

默认构造函数Hero();没有形参；给成员变量赋初值Hero(int x1,int x2,int x3);成员变量设置成私有Private，避免类外程序对成员变量进行更改 复制构造函数Hero &operator=(const Hero&rhs); Hero::Hero(int x1,int x2,int x3)其中::是**作用域运算符**

其中Const关键字是一个C语言（ANSI C）的关键字，具有着举足轻重的地位。它限定一个变量不允许被改变，产生静态作用。使用const在一定程度上可以提高程序的安全性和可靠性。另外，在观看别人代码的时候，清晰理解const所起的作用，对理解对方的程序也有一定帮助。

**析构函数**：

Hero::~Hero() //析构函数

{

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

}

**等号运算符：**

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;

}

**程序：**

**hero.h的内容**

#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;

};

**hero.cpp的内容**

#include"HERO.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;

}

**main.cpp的内容**

#include<iostream>

#include"HERO.h"

using namespace std;

int main()

{

HERO libai;

cout<<"libai skills:"<<endl;

libai.showskill();

HERO sunwukong = HERO(10, 10, 10);

cout<<"libaiskills:"<<endl;

sunwukong.showskill();

HERO hanxin(sunwukong);

cout<<"hanxin skills:"<<endl;

hanxin.showskill();

libai = sunwukong;

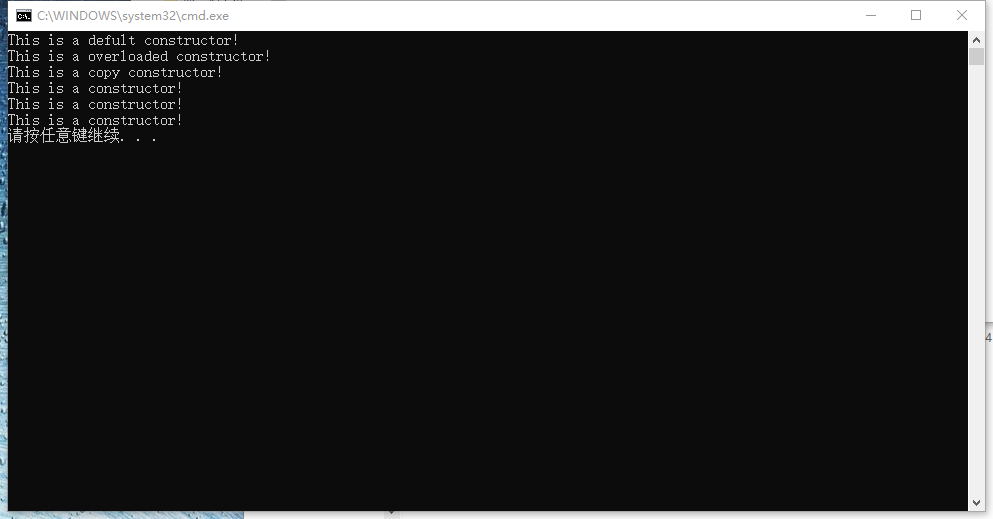
cout<<"libai skills:"<<endl;

libai.showskill();

return 0;

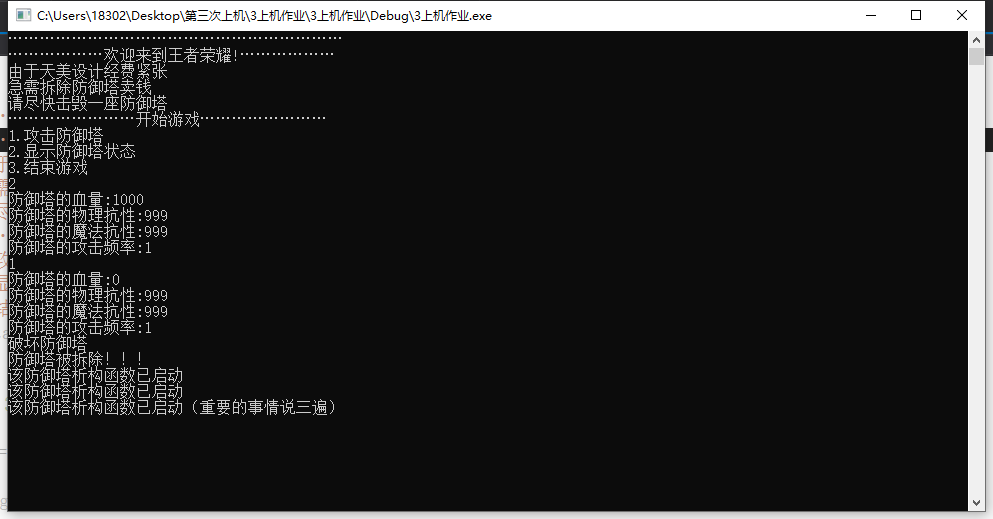
}

程序运行结果



**第三次上机作业：**（见附件）

拆除防御塔，并运行防御塔析构函数。



问题：

1、单引号&双引号的区别：

单引号‘’字符

双引号“”字符串

2、生成和调试的区别：

生成：顺序执行程序，不检查程序语句错误

调试：可以开始执行（不调试）加断点逐步运行查看程序运行结果，可以利用C++内部程序检查程序语句

**感悟：**

类（class）是面向对象程序设计的最基本的概念，是比结构体类型更安全更有效的数据类型，是C++最强有力的特征，看到了class就自然知道这是个C++程序，是进行封装和数据隐藏的工具，它将数据和操作紧密地结合起来。对象是类的实例，类是对象的抽象，面向对象程序设计中的对象来源于我们的现实世界，程序的可读性更强些。

定义一个类： class+类的名称

读程序看到class就能确定这是一个类，类是一种数据类型，是一种用户定义的抽象的数据类型。（它代表了一些对象的共性和特性）

定义类的对象：类：对象名称

**类类型声明的一般形式：**

**Class 类名{**

**[private]**

**私有数据成员和成员函数**

**public**

**公有数据成员和成员函数**

**}；**

花括号表示了一个类的声明范围，最后分号表示类声明结束。在类中封装有关数据和对这些数据进行操作的函数，即数据成员和成员函数。private public 还有protected关键字是成员访问限定符，声明各成员的访问属性：私有、公有、保护。这三种关键字的声明顺序可以任意出现任意次。private是私有属性，是默认值，外界不可以访问其中的数据成员和成员函数，以提高程序的安全性。

作用域运算符： “：：”称为作用域运算符。在类外定义成员函数时，必须在成员函数名之前缀上类名，在类名和函数名之间应加上作用域运算符“：：”，想要在局部变量的作用域内使用同名的全局变量，可以在该变量前加上“：：”

如Void Point：：setpoint(inta,intb)

**类名 对象名1，对象名2，...**

**对象名.数据成员名**

**对象名.成员函数名[(实参表)]**

我觉得类是把具有共同属性和行为的事物聚在一起构成的集合。一个类也就是用户声明的一个数据类型。C++中类与对象的关系可以用数据类型int和整型变量i之间的关系来类比。类类型和int类型均代表的是一般的概念，而对象和整型变量确是代表具体的东西。