实验一

代码：

#include "pch.h"

#include <iostream>

using namespace std;

enum CPU\_Rank { P1 = 1, P2, P3, P4, P5, P6, P7 };

class CPU {

public:

CPU(CPU\_Rank r,int f,float v) {

rank = r;

frequency = f;

voltage = v;

cout << "调用构造函数" << endl;

};

~CPU() {

cout << "调用析构函数" << endl;

}

void run() {

cout << "CPU运行" << endl;

};

void stop() {

cout << "CPU停止" << endl;

};

private:

CPU\_Rank rank;

int frequency;

float voltage;

};

int main()

{

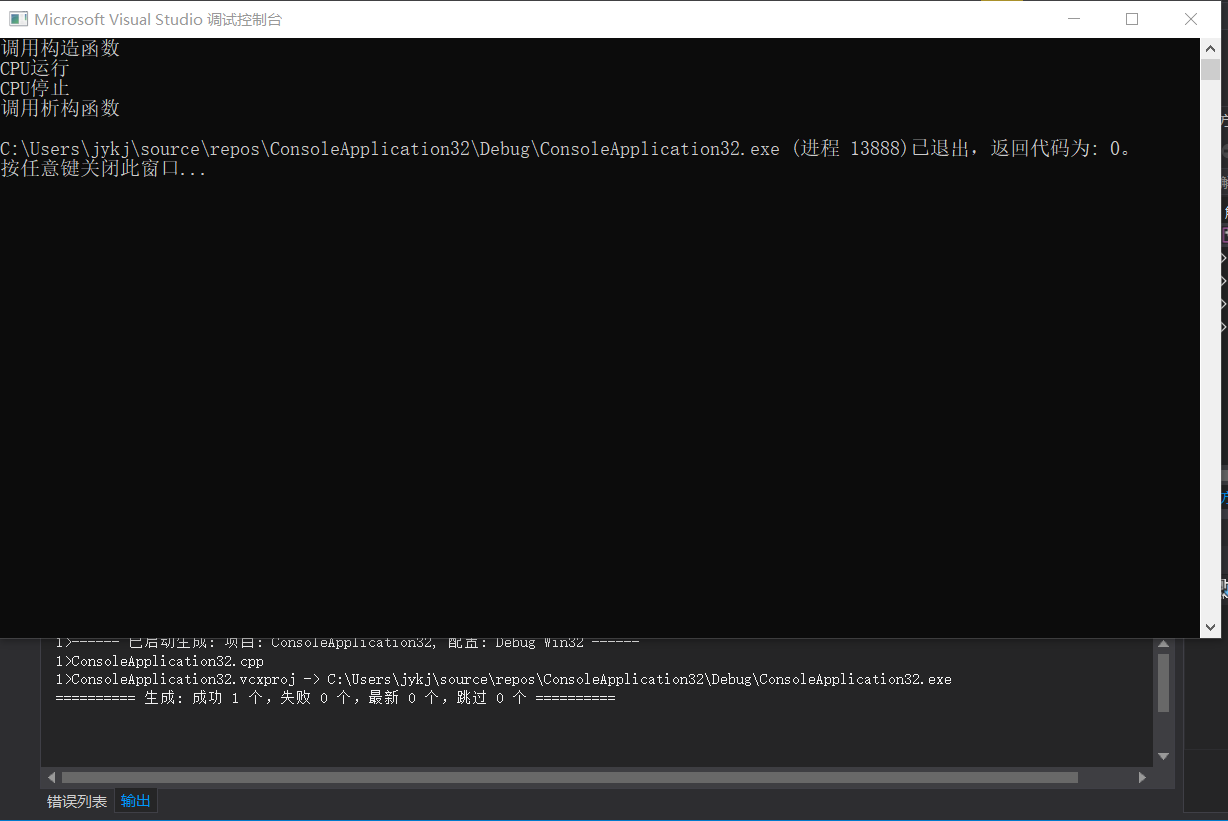
CPU cpu1(P3,200,110);

cpu1.run();

cpu1.stop();

}

运行结果：



实验二

代码：

#include "pch.h"

#include <iostream>

using namespace std;

enum CPU\_Rank { P1 = 1, P2, P3, P4, P5, P6, P7 };

class CPU {

public:

CPU(CPU\_Rank r, int f, float v) {

rank = r;

frequency = f;

voltage = v;

cout << "调用CPU的构造函数" << endl;

};

CPU() {

cout << "调用CPU的默认构造函数" << endl;

}

~CPU() {

cout << "调用CPU的析构函数" << endl;

}

void run() {

cout << "CPU运行" << endl;

};

void stop() {

cout << "CPU停止" << endl;

};

private:

CPU\_Rank rank;

int frequency;

float voltage;

};

class RAM {

public:

RAM() {

cout << "调用RAM的构造函数" << endl;

}

~RAM() {

cout << "调用RAM的析构函数" << endl;

}

};

class CDROM {

public:

CDROM() {

cout << "调用CDROM的构造函数" << endl;

}

~CDROM() {

cout << "调用CDROM的析构函数" << endl;

}

};

class Computer {

public:

Computer() {

cout << "调用Computer构造函数" << endl;

}

~Computer() {

cout << "调用Computer析构函数" << endl;

}

void run() {

cout << "computer运行" << endl;

}

void stop() {

cout << "computer停止" << endl;

}

private:

CPU cpu;

RAM ram;

CDROM cdrom;

};

int main()

{

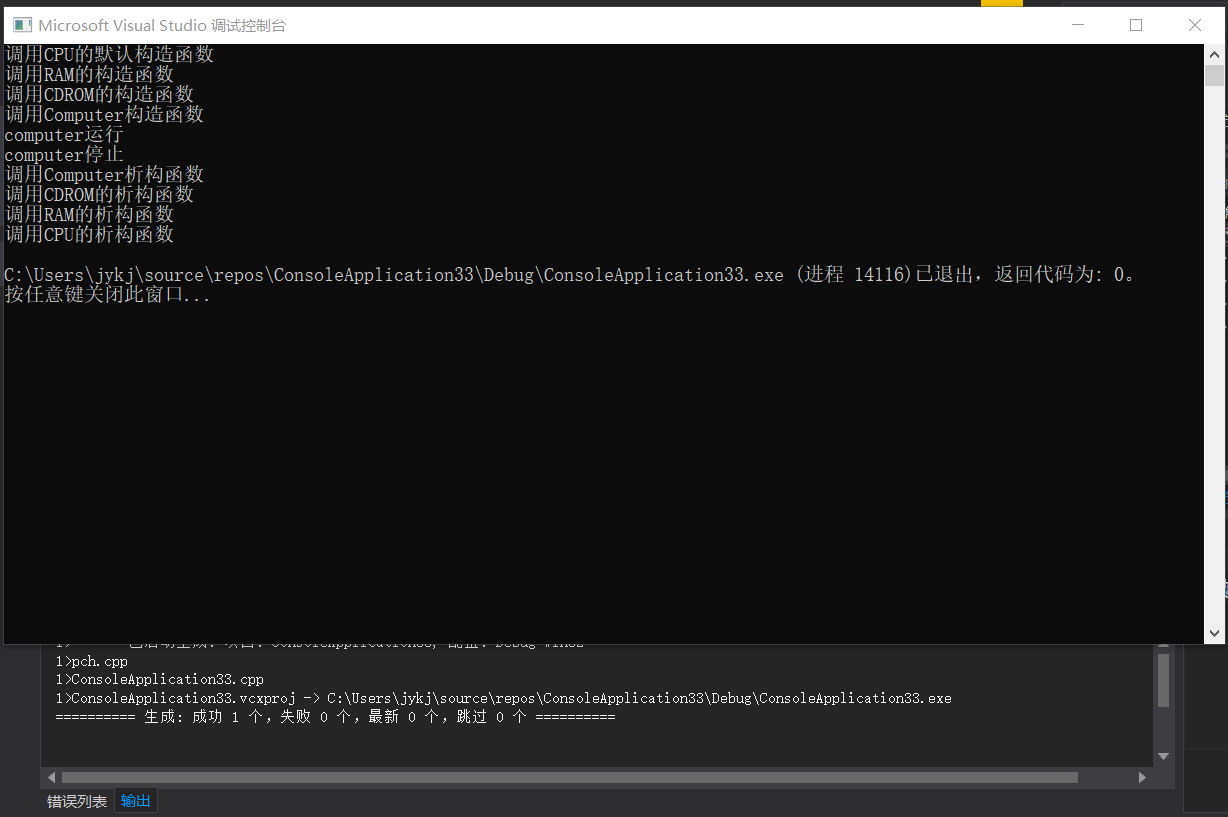
Computer com;

com.run();

com.stop();

}

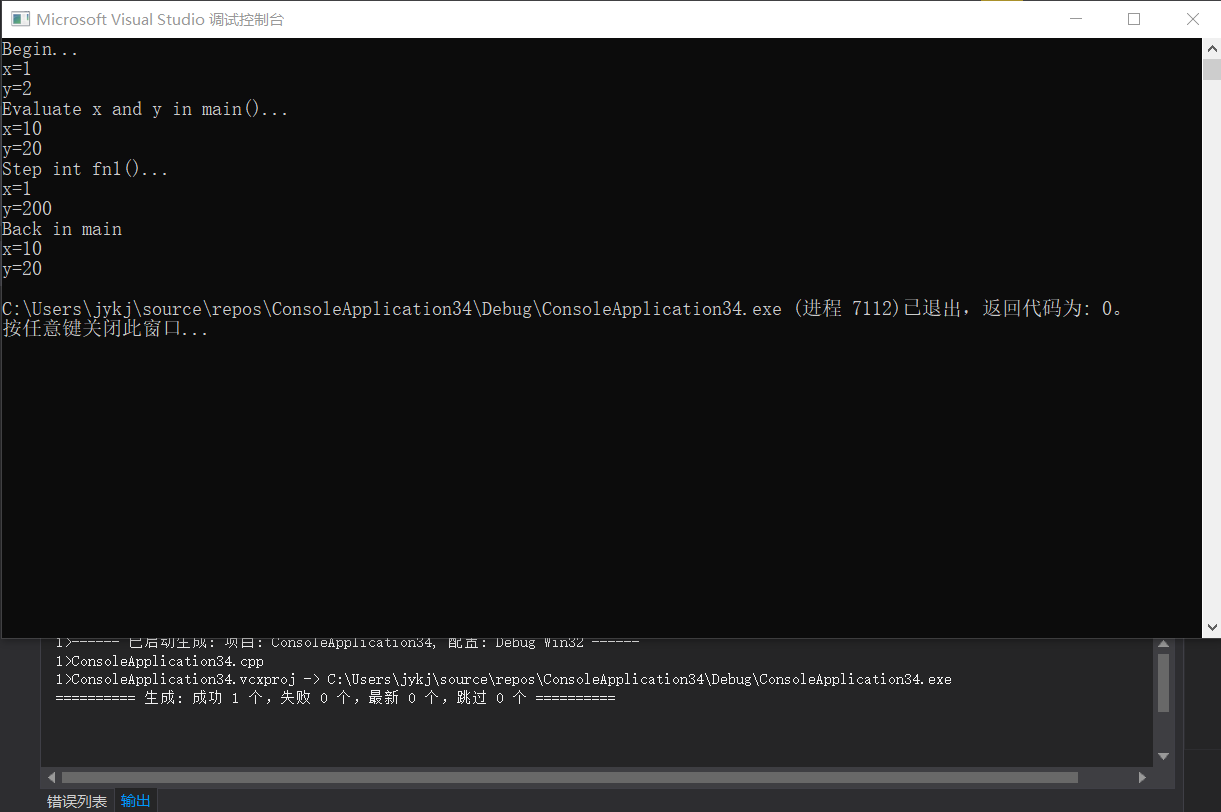
运行结果：



实验三：

（1）

运行结果：



（2）代码：

#include<iostream>

#ifndef client\_h

#define client\_h

class client {

public:

client();

~client();

static void Changeservername(char s);

static void showServername();

static void showClientnum();

private:

static char Servername;

static int Clientnum;

};

#endif

#include<iostream>

#include"client.h"

using namespace std;

client::client() { Clientnum++; }

client::~client() { Clientnum--; }

void client::Changeservername(char s) { Servername = s; }

void client::showServername() { cout << "服务器名：" << Servername << endl; }

void client::showClientnum() { cout << "客户总数：" << Clientnum << endl; }

char client::Servername = 'A';

int client::Clientnum = 0;

#include<iostream>

#include"client.h"

using namespace std;

void main() {

client::showServername();

client::showClientnum();

client::Changeservername('D');

client c;

c.showServername();

c.showClientnum();

{

client b;

b.showServername();

b.showClientnum();

}

client::showServername();

client::showClientnum();

}

运行结果：

