实验一：

#include<iostream>

using namespace std;

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

;

class CPU{

public:

CPU():rank(P5),frequency(20),voltage(2.6){

cout<<"construct one CPU"<<endl;

cout<<"rank:"<<rank<<endl;

cout<<"frequency:"<<frequency<<"MHz"<<endl;

cout<<"voltage:"<<voltage<<"V"<<endl;

}

~CPU(){

cout<<"destruct CPU"<<endl;

}

void run()

{

cout<<"this cpu is running"<<endl;

}

void stop()

{

cout<<"this cpu stop running"<<endl;

}

private:

CPU\_Rank rank;

int frequency;

float voltage;

};

int main(){

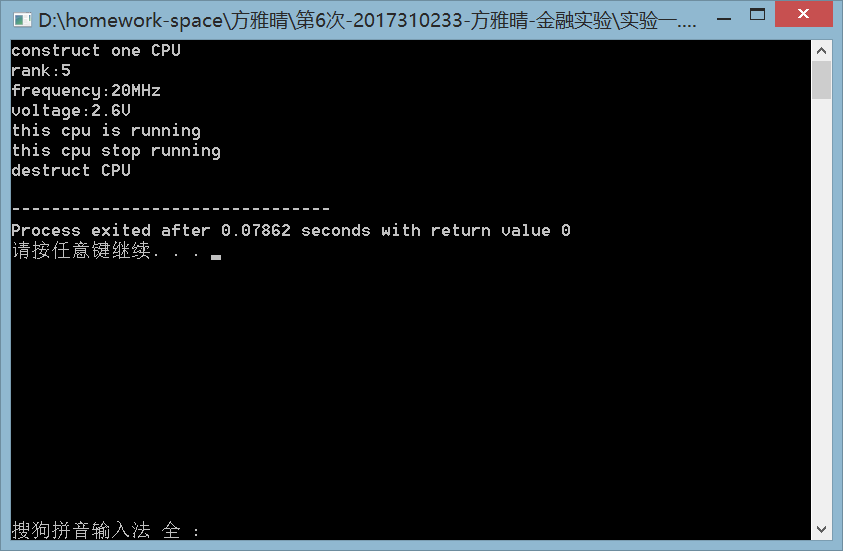
CPU c1;

c1.run();

c1.stop();

return 0;

}



实验二：

#include<iostream>

using namespace std;

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

;

class CPU{

public:

CPU():rank(P5),frequency(20),voltage(2.6){

cout<<"construct a CPU"<<endl;

}

CPU(CPU &c){

rank=c.rank;

frequency=c.frequency;

voltage=c.voltage;

cout<<"copy cpu"<<endl;

}

~CPU(){

cout<<"destruct CPU"<<endl;

}

private:

CPU\_Rank rank;

int frequency;

float voltage;

};

class RAM{

public:

RAM():frequency(2400),deposit(32){

cout<<"construct a ram"<<endl;

}

RAM(RAM &r){

frequency=r.frequency;

deposit=r.deposit;

cout<<"copy ram"<<endl;

}

~RAM(){

cout<<"destruct ram"<<endl;

}

private:

int frequency;

int deposit;

};

class CDROM{

public:

CDROM():frequency(3600),deposit(16){

cout<<"construct a cdrom"<<endl;

}

CDROM(CDROM &cd){

frequency=cd.frequency;

deposit=cd.deposit;

cout<<"copy cdrom"<<endl;

}

~CDROM(){

cout<<"destruct cdrom"<<endl;

}

private:

int frequency;

int deposit;

};

class computer{

private:

CPU cpu;

RAM ram;

CDROM cdrom;

public:

computer(CPU cpu1,RAM ram1 ,CDROM cdrom1):cpu(cpu1),ram(ram1),cdrom(cdrom1){

cout<<"construct computer"<<endl;

}

void run()

{

cout<<"this computer is running"<<endl;

}

void stop()

{

cout<<"this computer stop running"<<endl;

}

~computer(){

cout<<"destruct computer"<<endl;

}

};

int main(){

CPU cpu;

RAM ram;

CDROM cdrom;

cout<<"+++++++++++++++++++++++++++"<<endl;

computer com(cpu,ram,cdrom);

cout<<"+++++++++++++++++++++++++++"<<endl;

return 0;

}

实验三：

实验描述：

1. 运行以下程序，观察程序输出。

#include<iostream>

using namespace std;

void fn1();

int x=1,y=2;

int main()

{

cout<<"Begin..."<<endl;

cout<<"x="<<x<<endl;

cout<<"y="<<y<<endl;

cout<<"Evaluate x and y in main()..."<<endl;

int x=10,y=20;

cout<<"x="<<x<<endl;

cout<<"y="<<y<<endl;

cout<<"Step int fn1()..."<<endl;

fn1();

cout<<"Back in main"<<endl;

cout<<"x="<<x<<endl;

cout<<"y="<<y<<endl;

return 0;

}

void fn1()

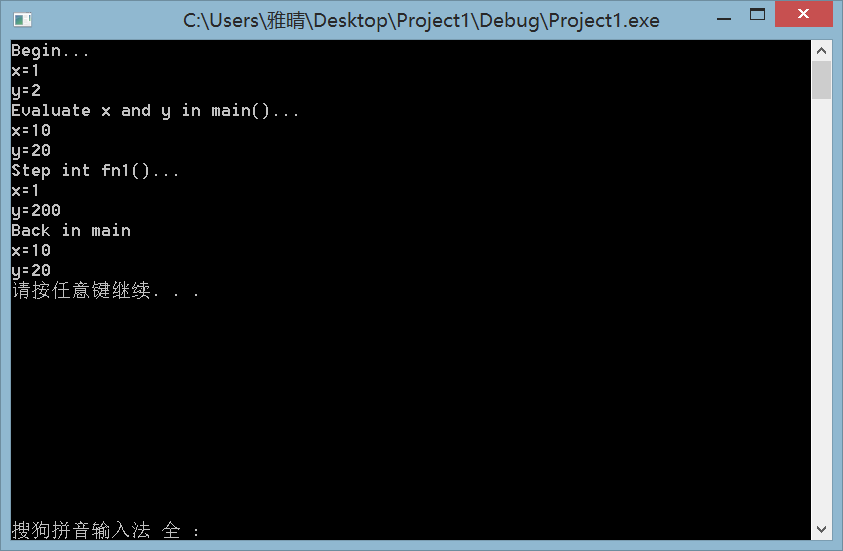
{

int y=200;

cout<<"x="<<x<<endl;

cout<<"y="<<y<<endl;

}



1. client.h

#ifndef client\_h

#define client\_h

#include<iostream>

#include<string>

using namespace std;

class CLIENT{

public:

CLIENT();

CLIENT(string n,string s,int a,bool v);

void get\_information();

string get\_vip();

void get\_total();

~CLIENT();

private:

string name;

string sex;

int age;

bool vip;

static int total;

};

#endif

Client.cpp

#include"client.h"

#include<iostream>

#include<string>

using namespace std;

int CLIENT::total=0;

CLIENT::CLIENT(string n,string s,int a,bool v):name(n),sex(s),age(a),vip(v)

{

total+=1;

cout<<"creat a client with information"<<endl;

}

string CLIENT::get\_vip()

{

if (vip == 0)

return "no";

else

return "yes";

}

void CLIENT::get\_information()

{

cout << "姓名：" << name << "\t性别：" << sex << "\t年龄：" << age << "\tVIP:" << get\_vip()<< endl;

}

void CLIENT::get\_total()

{

cout<<"目前有"<< total<<"位客户"<<endl;

}

CLIENT::CLIENT():name("null"), sex("null"), age(0), vip(0) {

total+=1;

cout<<"creat a client without information"<<endl;

}

CLIENT::~CLIENT()

{ total-=1;

cout<<"delete client "<<name<<endl;

}

lab5\_2.cpp

#include<iostream>

#include"client.h"

#include<string>

using namespace std;

int main(){

CLIENT A;

CLIENT \*P = &A;

P->get\_total();

P->get\_information();

CLIENT B("peter","male",18,1);

P = &B;

P->get\_total();

P->get\_information();

system("pause");

return 0;

}

