**实验作业**

**实验1：**

#include<iostream>

using namespace std;

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

class CPU

{

private:

CPU\_Rank rank;

int frequency;

float voltage;

public:

CPU(int newrank,int newfrequency,float newvoltage);

void run();

void stop();

~CPU(){cout<<"成功调用析构函数"<<endl;}

};

void CPU::run()

{

cout<<"程序开始执行"<<endl;

}

void CPU::stop()

{

cout<<"程序结束"<<endl;

}

CPU::CPU(int newrank,int newfrequency,float newvoltage)

{

rank=(CPU\_Rank)newrank;

frequency=newfrequency;

voltage=newvoltage;

cout<<"成功调用构造函数"<<endl;

cout<<"等级："<<CPU\_Rank(P1)<<endl;

cout<<"频率："<<frequency<<endl;

cout<<"电压："<<voltage<<endl;

}

int main()

{

CPU cpu(2,60,220);

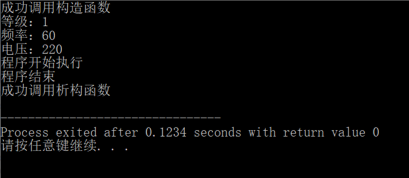
cpu.run();

cpu.stop();

return 0;

}

运行结果：



**实验2：**

#include <iostream>

using namespace std;

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

class CPU

{

private:

CPU\_Rank rank;

int frequency;

float voltage;

public:

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

{

rank=r;

frequency=f;

voltage=v;

cout<<"构造了一个CPU！"<<endl;

}

~CPU(){cout<<"析构了一个CPU！"<<endl;

}

CPU\_Rank GetRank()const{return rank;}

int GetFrequency()const{return frequency;}

float GetVoltage()const{return voltage;}

void SetRank(CPU\_Rank r){rank=r;

}

void SetFrequency(int f){frequency=f;

}

void SetVoltage(float v){voltage=v;

}

void Run(){cout<<"CPU开始运行！"<<endl;

}

void Stop(){cout<<"CPU停止运行！"<<endl;

}

};

int main() {

CPU cpu(P6,60,220);

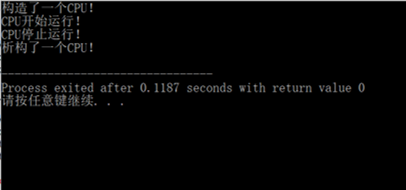
cpu.Run();

cpu.Stop();

return 0;

}

运行结果：



**实验3：**

#include"client.h"

#ifndef WYL\_CLIENT\_H

#define WYL\_CLIENT\_H

class CLIENT

{

public:

CLIENT () {ClientNumber++;};

CLIENT(CLIENT&xc);

static void ChangeServerName(char ch);

static void ShowServerName();

void ShowClientNumber();

~CLIENT();

private:

static char ServerName;

static int ClientNumber;

};

#endif;

#include<iostream>

using namespace std;

char c='T';

char CLIENT::ServerName=c;

int CLIENT::ClientNumber=0;

CLIENT::CLIENT(CLIENT&xc)

{

ClientNumber++;

ServerName=xc.ServerName;

}

CLIENT::~CLIENT(){ClientNumber--;cout<<"析构函数被调用"<<endl;}

void CLIENT::ChangeServerName(char ch)

{

ServerName=ch;

}

void CLIENT::ShowServerName(){cout<<"服务器的名称为："<<ServerName<<endl;}

void CLIENT::ShowClientNumber(){cout<<"现在为止客户机的数目为："<<ClientNumber<<endl;}

#include"client.h"

#include<iostream>

using namespace std;

int main ()

{

char ch1 = 'A',ch2 = 'B',char3 = 'C';

CLIENT::ShowServerName();

CLIENT myclient;

CLIENT::ChangeServerName(ch1);

myclient.ShowClientNumber();

myclient.ShowServerName();

CLIENT client2(myclient);

CLIENT::ShowServerName();

client2.ChangeServerName(ch2);

client2.ShowServerName();

client2.ShowClientNumber();

return 0;

}

运行结果：

