**实验一**

//cpu.h

#ifndef cpu\_h

#define cpu\_h

#include<iostream>

using namespace std;

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

class CPU {

public:

CPU() { cout << "constructing cpu" << endl; }

CPU(enum CPU\_RANK rank1, float vol, int freq) :rank(rank1), frequency(freq), voltage(vol) {

cout << "constructing cpu" << endl;

};

~CPU() { cout << "deconstructing cpu" << endl; };

void run() {};

void stop() {};

void show() {

cout << "The rank is P" << rank << endl;

cout << "The frequency is " << frequency << "MHz" << endl;

}

private:

enum CPU\_RANK rank;

float voltage;

int frequency;

};

#endif

//cpu.cpp

#include"cpu.h"

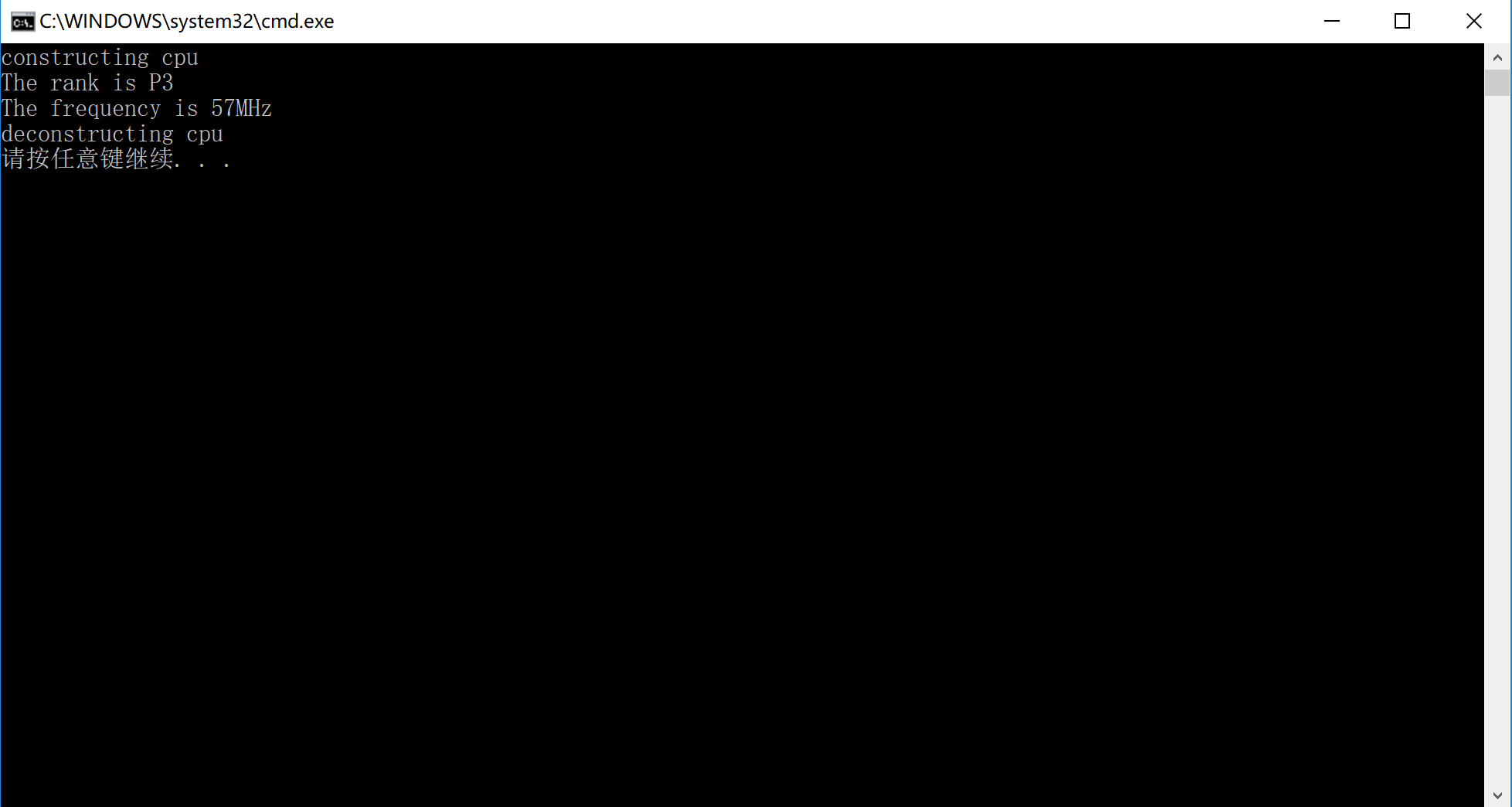
int main() {

CPU cpu1(P6, 3.2, 57);

cpu1.show();

return 0;

}



**实验二**

//ram.h

#ifndef ram\_h

#define ram\_h

#include<iostream>

using namespace std;

class ram {

public:

ram() {}

void show() {

cout << "It has a ram" << endl;

}

};

#endif

//cdrom.h

#ifndef cdrom\_h

#define cdrom\_h

#include<iostream>

using namespace std;

class cdrom {

public:

cdrom() {};

void show() {

cout << "It has a cdrom" << endl;

}

};

#endif

//computer.cpp

#include"cpu.h"

#include"cdrom.h"

#include"ram.h"

class compurter {

public:

compurter() {};

compurter(CPU &cpu1, ram &ram1, cdrom &cdrom1) :c\_cpu(cpu1), c\_ram(ram1), c\_cdrom(cdrom1) {};

void show() {

c\_cpu.show();

c\_cdrom.show();

c\_ram.show();

}

void run() {

cout << "The compurter is runing."<<endl;

}

void stop() {

cout << "The compurter has shut down." << endl;

}

private:

CPU c\_cpu;

ram c\_ram;

cdrom c\_cdrom;

};

int main() {

CPU cpu1(P6, 3.2, 57);

ram ram1;

cdrom cdrom1;

compurter compurter1(cpu1, ram1, cdrom1);

compurter1.run();

compurter1.show();

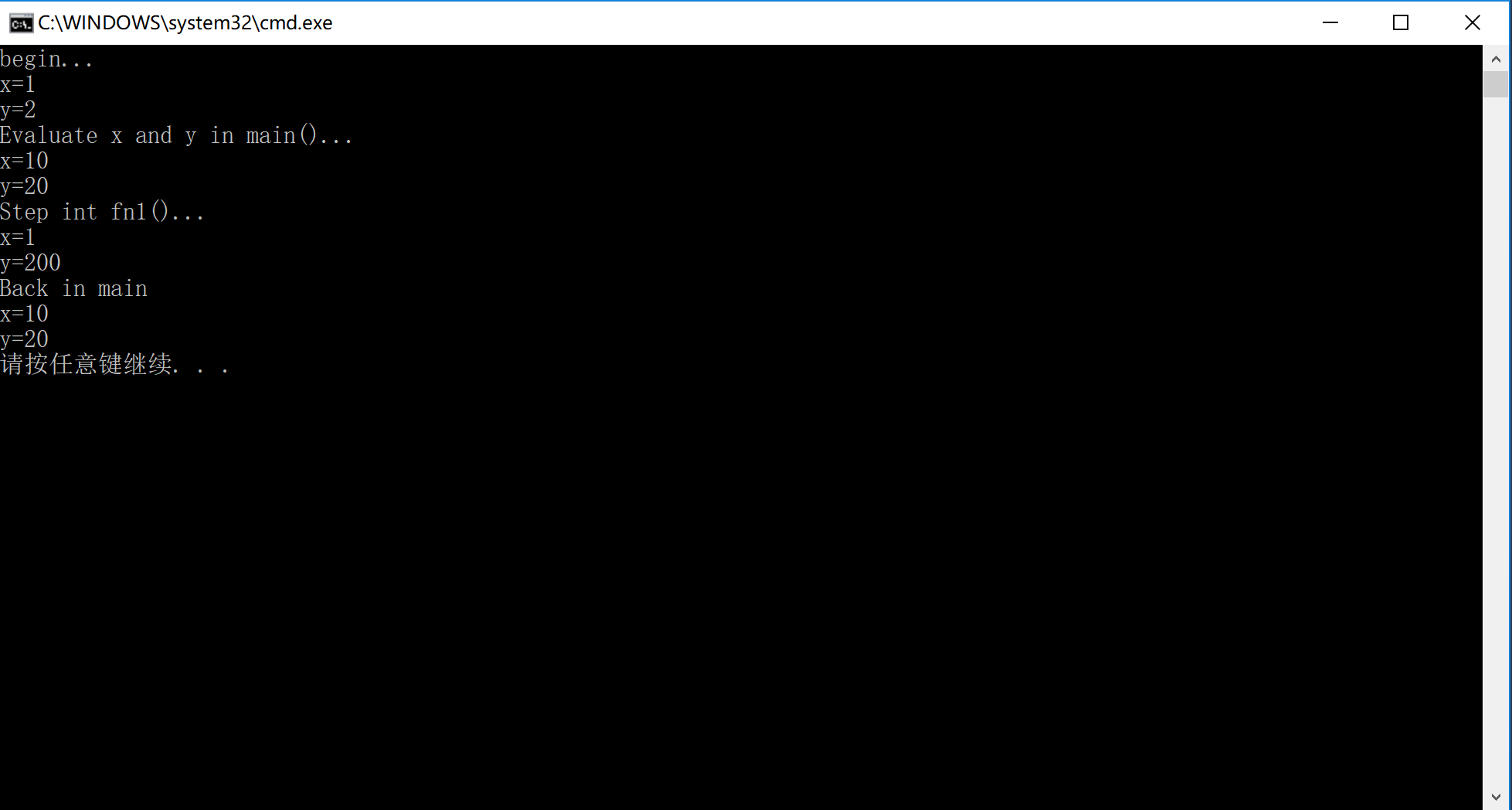
compurter1.stop();

return 0;

}

**实验三**

（1）



（2）

//client.h

#include<iostream>

#ifndef client\_h

#define client\_h

class client

{

public:

client() {};

client(int i);

client operator + (client &c);

client(client & c);

client operator++(int);

void show();

private:

static int count;

int i;

};

#endif

//client.cpp

#include "client.h"

using namespace std;

int client::count = 0;

void client::show() {

cout << i << endl;

}

client client::operator++(int) {

i++;

return client(i);

}

client client::operator +(client &c) {

count += c.i;

return client(i + c.i);

};

client::client(int i) :i(i) {

count += i;

};

client::client(client &c) :i(i) {

count += i;

cout << "CopyConsturctor" << endl;

};

int main() {

client a1(5), a2(7), a3,a4;

cout << "a1=";

a1.show();

cout << "a2=";

a2.show();

a3 = a1 + a2;

a4 = a2++;

cout << "a1+a2=";

a3.show();

cout << "a2+1=";

a4.show();

return 0;

}

