**实验一**

// 实验1.cpp

#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 rank, int frequency, float voltage);

~CPU() { cout << "析构函数ＣＰＵ" << endl; }

void run() { cout << "运行程序CPU" << endl; }

void stop() { cout << "结束程序CPU" << endl; }

private:

int frequency;

float voltage;

CPU\_Rank rank;

};

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

rank = r;

frequency = f;

voltage = v;

cout << "构造函数ＣＰＵ" << endl;

cout << "rank=" << rank << endl;

cout << "frequency=" << frequency << endl;

cout << "voltage=" << voltage << endl;

}

int main(){

CPU cpu(P4, 5, 4.23);

cpu.run();

cpu.stop();

return 0;

}

输出结果：

构造函数ＣＰＵ

rank=4

frequency=5

voltage=4.23

运行程序CPU

结束程序CPU

析构函数CPU

**实验二**

// 实验2.cpp

#include "pch.h"

#include<iostream>

using namespace std;

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

class CPU {

public:

CPU() { rank = P1, frequency = 3, voltage = 8.88; }

CPU(CPU\_Rank rank, int frequency, float voltage);

~CPU() { cout << "析构函数CPU" << endl; }

void run() { cout << "运行程序CPU" << endl; }

void stop() { cout << "结束程序CPU" << endl; }

private:

int frequency;

float voltage;

CPU\_Rank rank;

};

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

rank = r;

frequency = f;

voltage = v;

cout << "构造函数CPU" << endl;

cout << "rank=" << rank << endl;

cout << "frequency=" << frequency << endl;

cout << "voltage=" << voltage << endl;

}

class RAM {

public:

RAM() { ram = 0; }

RAM(const int r) {

ram = r;

cout << "构造函数RAM" << endl;

}

~RAM() { cout << "析构函数RAM" << endl; }

int getRAM()const { return ram; }

private:

int ram;

};

class CDROM {

public:

CDROM() { cdrom = 0; }

CDROM(const int c) {

cdrom = c;

cout << "构造函数CDROM" << endl;

}

~CDROM() { cout << "析构函数CDROM" << endl; }

int getCDROM()const { return cdrom; }

private:

int cdrom;

};

class Computer {

public:

Computer(CPU cpu, const RAM r, const CDROM c);

~Computer() { cout << "析构函数Computer" << endl; }

void run() { cout << "运行程序Computer" << endl; }

void stop() { cout << "结束程序Computer" << endl; }

private:

CPU cpu;

RAM ram;

CDROM cdrom;

};

Computer::Computer(CPU cpu, const RAM r, const CDROM c) {

cpu = cpu;

ram = r;

cdrom = c;

cout << "构造函数Computer" << endl;

cout << "ram=" << ram.getRAM() << endl;

cout << "cdrom=" << cdrom.getCDROM() << endl;

}

int main()

{

CPU cpu(P3, 2, 8.77);

Computer computer(cpu, 5, 8);

computer.run();

computer.stop();

return 0;

}

输出结果：

构造函数CPU

rank=3

frequency=2

voltage=8.77

构造函数CDROM

构造函数RAM

构造函数Computer

ram=5

cdrom=8

析构函数CPU

析构函数RAM

析构函数CDROM

运行程序Computer

结束程序Computer

析构函数Computer

析构函数CDROM

析构函数RAM

析构函数CPU

析构函数CPU

**实验三**

（1）

输出结果：

Begin...

x=1 //命名空间作用域x，y

y=2

Evaluate x and y in main()...

x=10 //主函数的x，y覆盖了命名空间的x，y

y=20

Step int fn1()...

x=1 //fn1()函数的y覆盖了命名空间的y；命名空间的x仍有效

y=200

Back in main

x=10 //主函数的x，y覆盖了命名空间的x，y

y=20

（2）

//client.h

#ifndef CLIENT\_H

#define CLIENT\_H

class CLIENT

{

public:

CLIENT();

~CLIENT();

static void ChangeServerName(char aa);

static void showServerName();

static void showClientNum();

private:

static char ServerName;

static int ClientNum;

};

#endif

//client.cpp

#include "client.h"

#include<iostream>

using namespace std;

CLIENT::CLIENT(){ClientNum++;}

CLIENT::~CLIENT(){ClientNum--;}

void CLIENT::ChangeServerName(char aa) { ServerName = aa; }

void CLIENT::showServerName() { cout << "ServerName:" << ServerName << endl; }

void CLIENT::showClientNum() { cout << "ClientNum:" << ClientNum <<endl; }

char CLIENT::ServerName = 'A';

int CLIENT::ClientNum = 4;

// lab5\_2.cpp

#include"client.h"

#include <iostream>

int main()

{

CLIENT::showServerName();

CLIENT::showClientNum();

CLIENT::ChangeServerName('B');

CLIENT a;

a.showServerName();

a.showClientNum();

{

CLIENT b;

b.showServerName();

b.showClientNum();

}

CLIENT::showServerName();

CLIENT::showClientNum();

return 0;

}

输出结果：

ServerName:A

ClientNum:4

ServerName:B

ClientNum:5

ServerName:B

ClientNum:6

ServerName:B

ClientNum:5