#include<iostream>

using namespace std;

//class Test {

// int \_data;

//public:

// Test(int data) :\_data(data) {

//

// }

//

// int get() const {

// this->\_data = 100;

// return \_data;

// }

//};

//

//void main() {

// Test t(10);

//}

class Computer {

char\* model;

char\* vendor;

double cpu\_power;

int core;

public:

Computer() :model(nullptr), vendor(nullptr), cpu\_power(0),

core(0) {

}

Computer(const char\* model, const char\* vendor, const double& cpuPower, const int& core)

{

SetModel(model);

SetVendor(vendor);

SetCpuPower(cpuPower);

SetCore(core);

}

Computer(const Computer& other) {

cout << "Copy constructor" << endl;

SetModel(other.GetModel());

SetVendor(other.GetVendor());

SetCpuPower(other.GetCpuPower());

SetCore(other.GetCore());

}

//copy assignment

Computer& operator=(const Computer& other) {

cout << "Copy assignment" << endl;

SetModel(other.GetModel());

SetVendor(other.GetVendor());

SetCpuPower(other.GetCpuPower());

SetCore(other.GetCore());

return \*this;

}

char\* GetModel() const {

return model;

}

char\* GetVendor()const {

return vendor;

}

double GetCpuPower()const {

return cpu\_power;

}

int GetCore()const {

return core;

}

void SetModel(const char\* model) {

int l = strlen(model);

this->model = new char[l + 1];

strcpy\_s(this->model, l + 1, model);

}

void SetVendor(const char\* vendor) {

int l = strlen(vendor);

this->vendor = new char[l + 1];

strcpy\_s(this->vendor, l + 1, vendor);

}

void SetCpuPower(const double& cp) {

this->cpu\_power = cp;

}

void SetCore(const int& core) {

this->core = core;

}

void ShowInfo() const {

cout << "Model : " << GetModel() << endl;

}

~Computer()

{

delete[]model;

delete[]vendor;

}

};

void foo(const Computer& a) {

cout << a.GetModel();

//a.SetModel("A");

}

void main() {

/\*Computer c("Rog", "Asus", 10, 4);

Computer d(c);

d = c;\*/

//Personal => fullname,image,age,HasPractise,SalaryRequirement

}