练习题1参考解答

一、**单项选择题(20分)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** | **9** | **10** | **11** | **12** | **13** | **14** | **15** | **16** | **17** | **18** | **19** | **20** |
| **a** | **c** | **c** | **a** | **a** | **d** | **c** | **b** | **c** | **d** | **d** | **a** | **d** | **d** | **d** | **d** | **c** | **d** | **d** | **c** |

二、（10分）

1）

1

3

5

4

2

三、**编程题（20分）**

1) （10分）

#include<iostream>

using namespace std;

class student {

private:

int num;

int score;

public:

student(int num, int score) :num(num), score(score){

cout << "Constructor called." << endl;

}

student(): num(0), score(0) {

cout << "Default Constructor called." << endl;

}

~student() { cout << "Destructor called." << endl; }

int getnum() const { return num; }

int getscore() const { return score; }

};

int Max(student\* p)

{

int max = p->getscore();

int j = 0;

for (int i = 0; i < 5; i++)

{

if (max < (p + i)->getscore())

{

max = (p + i)->getscore();

j = i;

}

}

cout << "最高分是：" << max << endl;

return (p + j)->getnum();

}

int main()

{

student a[] = {

student(1,67),student(2,57),student(3,96),student(4,76),student(5,46)

};

cout << "得最高分的num是：" << Max(a) << endl;

return 0;

}

2) （10分）

class Range

{

public:

double beg, end;

Range() {}

Range(double \_beg, double \_end):beg(\_beg),end(\_end) {}

};

Range overlappedRange(const Range v[], int n)

{

Range r(0,0);

if (n > 0)

{

r = v[0];

for (int i = 1; i < n; ++i)

{

if (v[i].beg > r.beg)

r.beg = v[i].beg;

if (v[i].end < r.end)

r.end = v[i].end;

}

}

return r;

}

void main()

{

Range v[] = { Range(0.2, 3.5), Range(1.1, 4.8), Range(-1, 3.0), Range(1.8, 3.4) };

Range r;

//在此处，用一行代码调用overlappedRange函数，计算数组v中区间的交集，结果存入r

r = overlappedRange(v, 4);

}