

[回到课程](#)

模拟测试系列2

1 【填空题】

请将下列计算学生平均成绩的程序补充完整:

```
#include<iostream>
#include<string>
using namespace std;
class student {
public:
    student(string n, int s)_____(1)_____ {
        num++;
        _____(2)_____
    }
    _____(3)_____ { return sum / num; }
    void display() { cout << num << ":" << name << "," << score << endl; }
    ~student() { num--; sum -= score; }
private:
    string name;
    int score;
    static int num;
    static float sum;
};
_____(4)_____
_____(5)_____

int main()
{
    student p[3]={ { "Mary" ,60.1},{ "Tom" ,80.1}, { "Jack" ,40.1}};
    p[2].display();
    cout<<student::avg()<<endl;
    return 0;
}

程序运行输出:
Jack,40.1
60.1
```

(10.0分)

第一空:

第二空:

第三空:

第四空:

第五空:

2

【简答题】

阅读程序写运行输出

```
#include <iostream>
using namespace std;
class Sample
{
protected:
    int x;
public:
    Sample() { x = 0; }
    Sample(int val) { x = val; }
    Sample& operator++() { x++; return *this; }
};
class Derived :public Sample
{
    int y;
public:
    Derived() :Sample() { y = 0; }
    Derived(int val1, int val2) :Sample(val1) { y = val2; }
    Sample& operator--() { x--; y--; return *this; }
    void disp()
    {
        cout << "x=" << x << ",y=" << y << endl;
    }
};
int main()
{
    Derived d(3, 5);
    d.disp();
    ++d;
    d.disp();
    --d;
    --d;
    d.disp();
    return 0;
}
```

(5.0分)

填写答案

3

【简答题】

阅读程序写运行输出

```
#include <iostream>
using namespace std;
class base {
public:
    virtual int func() { return 10; }
};
class derived:public base {
public:
    int func() { return 20; }
};
int main()
{
    derived d;
    base & b = d;
    cout << b.func() << endl;
    cout << b.base::func() << endl;
    cout << d.func() << endl;
    cout << d.base::func() << endl;
    return 0;
}
```

(5.0分)

填写答案

4

【简答题】

阅读程序写运行输出

```
#include <iostream>
using namespace std;
```

```

class A {
public:
    virtual void f() { cout << "A::f()\n"; }
    virtual ~A() { cout << "Destroy A object" << endl; }
};

class B :public A
{
private:
    char *buf;
public:
    B(int i) {
        buf = new char[i];
    }
    void f() {
        cout << "B::f()\n";
    }
    ~B() { cout << "Destroy B object" << endl; delete[]buf; }
};

int main()
{
    A *a = new A;
    a->f();
    delete a;
    a = new B(15);
    a->f();
    delete a;
    return 0;
}

```

(5.0分)

填写答案

5 【简答题】

阅读程序写运行输出

```

# include<iostream>
using namespace std;
class Location {
    int X, Y;
public:
    Location(int x = 1, int y = 3) :X(x), Y(y) { }
    void init(int initX, int initY);
    int GetX() const ;
    int GetY() const ;
};

void Location::init(int initX, int initY){
    X = initX;
    Y = initY;
}

int Location::GetX()const {    return X;  }

int Location::GetY()const {    return Y;  }

void display(const Location& rL){
    cout << rL.GetX() <<"," << rL.GetY() << "\n";
}

int main()
{
    Location A[5] = { {0,0},{1,1},{2,2},{3,3},{4,4} };
    Location *rA = A;
    display(*rA);
    A[3].init(5, 6);
    rA->init(7, 8);
    for (int i = 0; i < 5; i++)
        display(*(rA++));

    return 0;
}

```

(5.0分)

填写答案

6 【简答题】

阅读程序写运行输出

```

# include<iostream>

```

```
using namespace std;
template<class X>
void testF1(X x, X y) {
    cout << "I'm testF1:" << x << ", "<< y << endl;
}

void testF1(int x, int* y) {
    int temp = x;
    x = *y;
    *y = temp;
    cout << "I'm testF2:" << x<< ", "<< *y << endl;
}

void testF1(int x, char y) {
    cout << "I'm testF3:" << x << ", "<< y << endl;
}

int main()
{
    int x = 49, y = 98;
    char m = 'D', n = '\x32';
    testF1(x, y);
    testF1(x, n);
    testF1(m, n);
    testF1(x, &y);
    testF1(m,&y);
    return 0;
}
```

(5.0分)

填写答案

7 【简答题】

1、 程序设计：（10分）
现有存放整数的一般单向链表head，现在要求你处理链表，使得这个链表中最后只保留偶数值的结点。请你编程实现这个处理要求。
给出的结点定义为：
class node{
public:
node(int v, node* n=nullptr):value(v){ }
int value;
node* next;
};
处理工作的函数原型为：node* DeleteOdd(node *head);

(10.0分)

填写答案

8 【简答题】

2、 程序设计：
现在需要设计一个班级管理系统，请你来设计相关的类型。已经了解到的情况为：
（1） 班级中的学生人数30-50人不等；班级是按系、专业、年级、班级名来区分的，如：计算机科学与技术系软件工程专业2年级1班；每个班级有一名班长。
（2） 学生需要管理的信息有：学号、姓名、性别、出生年月，5门功课的成绩。
（3） 可以按班级显示班级的所有信息，能够显示班级中所有学生的信息，能够按照班级中学生的成绩总分排序。

请你使用自定义类型机制来实现上述系统相关的类型，给出类型声明即可，不需要实现，并给出必要的数据和接口的注释。（5分）

(5.0分)

填写答案

暂时保存

提交

