趋势科技笔试题专辑（含答案）

本专辑目录

本专辑由逍遥游(http://blog.sina.com.cn/xiaoyaoyoucn)整理
趋势科技笔试题专辑
http://blog.sina.com.cn/s/blog_684bb6fb0100jtve.html

趋势科技笔试题(一)

趋势科技的笔试题
1、#include
class A{
public:
A() {func(0);};
virtual void func(int data) {printf("A1 :%d\n",data);}
virtual void func(int data) const {printf("A2 :%d\n",data);}
void func(char *str) {printf("A3 %s)\n",str);}
};

class B:public A{
public:
void func() {printf("B1 :%s\n","");}
void func(int data) {printf("B2 :%d\n",data);}
void func(char *str) {printf("B3 %s)\n",str);}
};

int main()
{
A *pA;
B b;
const A *pcA;
pA=&b;
pA->func(1);
pA->func("test");
A().func(1);
pcA=&b;

pcA->func(2);
return 0;
}
程序运行的结果:
A1 :0
B2 :1
A3 test)
A1 :0
A1 :1
A2 :2
1) 基类的指针指向派生类对象：那么该指针只能够调用基类所定义的函数，但是如果该函数为虚函数，则调用该派生类自己的成员

函数。(B2 :1)
2) 如果以派生类的指针指向基类对象，则必须事先做明显的转型操作，但是这种做法很危险。
2、#include
template
void func(const int &t)
{
cout<}
template
void func(const T&t)
{
cout<}

int main()
{
func(10.3);
func(1000);
return 0;
}
程序运行结果：
10.3
1000
如果上述函数改为
＃i nclude
void func(const int &t)
{
cout<}
template
void func(const T&t)
{
cout<}
int main()
{
func(10.3);
func(1000);
return 0;

}
则程序的运行结果为：
10.3
1100
如果使用函数的非模板形式，不能在前面加上 template 关键字。
3、改错：
#i nclude
class klass
{
public:
klass(){}
private:
~klass(){}
void func(int n){
cout<<"klass!!"< }
public:
void test()
{
func(100);
}
};
int main()
{
klass k;
k.test();
return 0;
}
运行后程序显示：error C2248: 'klass::~klass' : cannot access private member declared in class 'klass'
证明析构函数的属性必须为 public。
但是，如果把 klass k 改为 klass* pk; pk=new klass; pk->test();程序通过，但是 klass 不能释放.
趋势的笔试题(zz)
趋势的笔试题(zz)
1,你有 5 瓶药，每个药丸重 10 克，只有一瓶受到污染的药丸重量发生了变化，每个药丸重 9 克。给你一个天平，你怎样一次就能测出哪

一瓶是受到污染的药呢？
2. 十个苹果,有一个不同.或轻或重.称三次.
3,有 4 个女人要过一座桥。她们都站在桥的某一边，要让她们在 1 7 分钟内全部通过这座桥。这时是晚上。她们只有一个手电筒。最

多只能让两个人同时过桥。不管是谁过桥，不管是一个人还是两个人，必要带着手电筒。手电筒必须要传来传去，不能扔过去。每个

女人过桥的速度不同，两个人的速度必须以较慢的那个人的速度过桥。第一个女人：过桥需要 1 分钟；第二个女人：过桥需要 2 分钟

；第三个女人：过桥需要 5 分钟；第四个女人：过桥需要 1 0 分钟。

答案 1. 调试好天平后,天平 左右盘各放两瓶,有下列情况:
1,天平平衡,则余下那瓶的是受污染的药;
2,天平左倾,则目标瓶在右盘,现在同时从左右盘拿下一瓶,观察到:若天平恢复平衡,则目标瓶就是右盘拿下的那瓶,若天平依旧左倾则

目标就是右盘余下的那瓶; 3.天平右倾,用上述 2 类似的方法判断. 整个过程只用天平一次,只是注意拿下药瓶时从左右盘同时拿一瓶

,然后稍作分析即可.
答案 2. 1) 先取 4 个苹果,一边 2 个放天平.结果有 A 和 B2 种: A) 重量相同,另取 4 个来称.结果有 a 和 b2 种: a) 如重量相同,留 4 个其中的 1

个,取剩余 2 个中的 1 个放天平另一边来称.如不同,那个就是了.如相同,最后的那一个就是了. b) 如重量不同,参照 B 方案. B) 重量不

同,取任意一边的 2 个,一边 1 个来称.结果有 c 和 d2 种: c) 相同,则留其中一个,取剩余 2 个中的 1 个放天平另一边来称.如不同,那个就是

了.如相同,最后的那一个就是了. d) 不同,则留其中一个,从已称过的中取 1 个放天平另一边来称.如不同,那个就是了.如相同,另一个

就是了. (faint，这样就称了四次了)
答案 3 1,2 go 2 1 back 1 5,10 go 10 2 back 2 1,2 go 2

1.8 Which virtual function redeclarations of the Derived class are correct?
a. Base* Base::copy(Base*); Base* Derived::copy(Derived*);
b. Base* Base::copy(Base*); Derived* Derived::copy(Base*);
c. ostream& Base::print(int,ostream&=cout); ostream& Derived::print(int,ostream&); d. void Base::eval() const; void

Derived::eval();

#include #include
using namespace std;
int main(int argc,char * argv[])
{
string strArr1[]={"Trend","Micro","soft"};
string *p=new string[2];
p[0]="US";
p[1]="CN";
cout < ;
for(i="0;i" ;

#include
using namespace std;
class CDemo
{
public:
CDemo() : str(NULL) {};

```
    ~CDemo() { if (str) delete [] str;}
    char *str;
};
void main (void)
{
CDemo d1;
d1.str = new char[32];
strcpy (d1.str, "trend micro");
cout << d1.str; vector *a1 = new vector();
a1->push_back(d1); delete a1;
```

} 哪里有问题?

vector *a1 = new vector(); 这句没看懂 new vector() 为什么后面是()跟着这样写是调用无参的构造函数，看看 VECTOR 原型就知

道了

```
class CDemo
{
public: CDemo() : str(NULL) {};
~CDemo() { if (str) delete [] str;}
CDemo(const CDemo& x):str(0)
{
if(x.str)
{
str = new char[strlen(x.str)+1];
strcpy(str, x.str);
}
}

CDemo& operator=(const CDemo& x)
{
if(this == &x)
    return *this;
delete []str;
str = 0;
if(x.str)
{
    str = new char[strlen(x.str)+1];
    strcpy(str, x.str); } return *this;
}
char *str; };
void main (void) // int main()
```

两次析构其实是对 CDemo 中的 str 引用的内存进行了两次析构，分别是 d1 的析构和 delete a1 发生的析构。只要在 CDome 中，当 CDome 的对象发生拷贝构造或拷贝时，就将 str 指向不同的内存即可。所以会在拷贝构造函数和复制操作符中，从新申请一块内存并把源字符串拷贝到新的内存中就 OK 了

1.5 Which of the following class DOES NOT need a copy constructor?

a. A matrix class in which the actual matrix is allocated dynamically within the constructor and is deleted within its destructor.

b. A payroll class in which each object is provided with a unique ID.

c. A word class containing a string object and vector object of line and column location pairs. d. A library class containing a list of book object.

1.6 What is the color of the pixel at x,y?

```
int x = 00,y = 100;
putpixel(x,y,RED);
if(getpixel(x,y)==RED)
{
putpixel(x,y,GREEN);
}
if(getpixel(x,y)==BLUE)
{
putpixel(x,y,YELLOW);
}

if(getpixel(x,y)==GREEN)
{
putpixel(x,y,BLUE);
}
```

a.RED b.GREEN c.YELLLOW d.BLUE

1.7 What is the output of the following code?

```
void turn_left(int dir)
{
      dir = (dir - 1);
}
void turn_right(int *p_dir)
{
*p_dir = (*p_dir + 1);
}
int main()
{ int dir1 = 3, dir2 = 3;
turn_left(dir1);
turn_right(&dir2);
printf("%d %d",dir1,dir2);
return 0;
}
```

a.3 3 b.3 4 c.2 3 d.none of the above 今年趋势的一个笔试题 呵呵，当时看了真是觉得搞笑啊，这简直是恶搞题…… 小明每周一到周五要去上学，周末休息。今天早上小明没有刷牙，这是为什么呢？请给出尽量多的原因。

```
#include class A
```

```
{
public:
      A() {func(0);};         virtual void func(int data)
     {
      printf("A1 :%d\n",data);
     }
      virtual void func(int data) const {printf("A2 :%d\n",data);}         void func(char *str) {printf("A3 %s)\n",str);}
};

class B:public A
{
public:
      void func() {printf("B1 :%s\n","");}         void func(int data) {printf("B2 :%d\n",data);}
      void func(char *str) {printf("B3 %s)\n",str);}
};

int main()
{
       A *pA;
       B b;
      const A *pcA;
      A1 :0 pA=&b;
      pA->func(1);
      B2 :1 pA->func("test");
      A3 test) A().func(1);
      A1 :0 A1 :1 pcA=&b;
      pcA->func(2);
      A2 :2
      return 0;
}
```

程序运行的结果: A1 :0 B2 :1 A3 test) A1 :0 A1 :1 A2 :2 1) 基类的指针指向派生类对象：那么该指针只能够调用基类所定义的

函数，但是如果该函数为虚函数，则调用该派生类自己的成员函数。(B2 :1) 2) 如果以派生类的指针指向基类对象，则必须事先做明显的转型操作，但是这种做法很危险。

趋势科技笔试面试经历 1 可以说，趋势科技是今年校园招聘开始的最早的公司之一了。10 月 15 号，就收到了趋势的笔试通知。怀着激=动的心情，参加了人生第一次笔试。试题是全英文的，分两部分，第一部分是 45 分钟的 IQ 和 EQ 题，这部分对我来说比较容易，平时看的多，做的很顺利。第二部分是技术题，120 分钟，涵盖了 c,c++，数据结构，数据库，网络（特别是 tcp/ip）等知识，还有几道科技短文阅读。由于好久没碰过英语，很多关键的英语单词忘了意思，严重影响了我的发挥。考官说两周以内等面试通知。我也没

放在心上。一个星期后，收到了趋势的面试电话，通知我去地铁大厦参加面试。当然很高兴啦。后来发现，同去笔试的几个兄弟都没收到面试通知，嘿嘿，看来我发挥的还不错嘛。当天下午，我就买了一件比较象样的衣服，毕竟是人生的处女面啊。晚上上网了解了下趋势科技，网络安全方面做的比较好的一个公司，新浪，hotmail 用的都是趋势的产品。我们校园网的杀毒软件用的也是趋势的。

趋势的待遇很好，听说有 8000 左右一个月，流口水 ing。上午 10：30 的面试，我提前半小时就到了，遇到了很多

其他高校的帅哥，在和他们的聊谈过程中，我知道，他们竟都是硕士生，参加这次面试的有 100 位，最后只招二三十个。顿感压力巨大啊。 很准时,10:30,我被叫进去开始面试了.面试官是个四十出头的很和蔼先生,穿的也很随便,也许这就是趋势的企业文化?整个面试过程感觉还不错，技术问题基本都答上来了，一些开放性的问题答的也还可以，只是我有点紧张,感觉得出我地声音有点颤抖.下面我简单罗列

下我碰到的问题 1、简单介绍一下你做的项目
2、sturct 和 class 的区别
3、虚函数和纯虚函数的区别，作用
4、有没有碰到过内存泄漏，怎么解决 5、老师对你的评价
6、你认为哪个老师比较牛，为什么？
7、同学们怎么评价你的，你同学认为你哪方面比较牛？
8、平时有什么兴趣和爱好
8、简单谈谈你对趋势的看法走出地铁大厦,我还是很自信的.回到宿舍后突然我发现我那装简历的档案袋没拿,我顿时后悔起来,这是

一个很严重的错误，从这个小小的疏忽，可以看出我的紧张，大意。这在工作中是万万要不得的.我想,就凭这一点,我是无望进入二面了.结果如我所料,没有收到下一轮的面试通知.这次失败的面试给了我惨痛的教训,以后无论做什么事,都要小心谨慎。 呵呵，凭我记忆写的首先是一堆智力题目，看图形选出不一样的，或者填出下一个然后是选择题，有一部分关于网络的，包括数据通信，什么路由器交换机之类的然后还有一些关于的，包括编程，呵呵考了一个原始套接字的，记住选就然后是一些大题。一道是数据结构的，链表的节点的反转，然后还有一道是写还是的原型来着。就记得这么多了，希望对师弟师妹有帮助。

2、sturct 和 class 的区别 STRUCT 中的所有成员都是 PUBLIC 的，类中可以有 PUBLIC，PRIVATE，PROTECTED。STRUCT 中不能包含成员函

数但是 CLASS 中可以包括成员函数.class 和 struct 的最基本的区别就是：在 class 中，在默认的情况下，它的成员是私有的，而在 struct 中，默认的情况下，它的成员是公有的!但是在一般编程的习惯中，在 struct 中一般只定义数据部分，而在 class 中一般都定义了数据部分和对这些数据操作的方法！

3、虚函数和纯虚函数的区别，作用纯虚函数就是没有函数体的，必须在派生类中重载德函数。虚函数可以有函数体，如果派生类中没有重载函数，则调用父类的函数 class a { virtual b()=0;//纯虚函数 virtual c(){...}//虚函数 }；虚函数主要实现多态机制避免二义性问题至于纯虚函数是抽象类机制，基类提供接口，派生类提供实现抽象类不能定义对象？最大区别：有纯虚函数的类不能定义对象。 基本来说纯虚函数是必须被重载的，因此在被用来做基类的抽象类中肯定有一个或多个纯虚函数。 而虚函数可以在继承类中被重载，也可以不。 二者都是体现了 c++的多态性
纯虚函数:
1.表明该纯虚函数所属的类仅做为接口使用,不能实例化(即不能生成一个该类的对象).
2.接口类的子类(derived class)必须 overriden 每个纯虚函数,使之成为非纯虚函数后,该类才能产充许产生实例;
3.纯虚函数可以有该函数的定义,有可以没有.但纯虚析构函数必须要有定义. 如果一个接口类只有它的析构函数的纯虚的,那么,该接

口类的子类可以不必显式 overriden 接口类的析构函数,编译器自动产生的析构函数就可以 overriden 接口类的析构函数,使子类自动成为一个可实例化的普通类. 4。有没有碰到过内存泄漏，怎么解决就是使用内存资源后没有被回收。 在 java 中，用 new 在堆上分配的内存资源都会被 java 的 garbage collector 自动回收当一个类的实例不再被其它的任何变量引用的时候，它就有资格被回收，但是并不是一定会被回收，如果没有被回收，则酒会产生内存泄露 对，单元测试，一个一个得试，看是哪个模块出了问题。有点费时间，但这个最保险了代码的规范性很重要。

趋势科技笔试题(二)

1、#include <stdio.h>

```
class A{
    public:
        A(){func(0);};
        virtual void func(int data){printf("A1 :%d\n",data);}
        virtual void func(int data) const{printf("A2 :%d\n",data);}
        void func(char *str){printf("A3 :(%s)\n",str);}
    };
class B:public A{
public:
        void func(){printf("B1 :%s\n","");}
        void func(int data){printf("B2 :%d\n",data);}
        void func(char *str){printf("B3 :(%s)\n",str);}
    };
int main()
{
        A *pA;
```

```
        B b;
        const A *pcA;
        pA=&b;
        pA->func(1);
        pA->func("test");
        A().func(1);
        pcA=&b;
        pcA->func(2);
        return 0;

}
```

程序运行的结果:

```
A1 :0
B2 :1
A3 :(test)
A1 :0
A1 :1
A2 :2
```

1)

基类的指针指向派生类对象：那么该指针只能够调用基类所定义的函数，但是如果该函数为虚函数，则调用该派生类自己的成员函数

。(B2 :1)

2) 如果以派生类的指针指向基类对象，则必须事先做明显的转型操作，但是这种做法很危险。

2、

```
include <iostream.h>
template <typename T>
void func(const int &t)
{
        cout<<t+100<<endl;
}

template<typename T>
void func(const T&t)
{
        cout<<t<<endl;
}

int main()
{
        func(10.3);
        func(1000);
        return 0;
}
```

程序运行结果：

10.3
1000
如果上述函数改为

include <iostream.h>
void func(const int &t)

{

cout<<t+100<<endl;

}

template<typename T>

void func(const T&t)

{

cout<<t<<endl;

}

int main()

{

func(10.3);

func(1000);

return 0;

}

则程序的运行结果为：
10.3
1100

如果使用函数的非模板形式，不能在前面加上 template 关键字。

3、
改错：
include <iostream.h>
class klass

```
{

public:
        klass(){}
private:
        ~klass(){}

        void func(int n){

            cout<<"klass!!"<<endl;

            }           public:

        void test(){
            func(100);
        }
};

int main()
{
        klass k;
        k.test();
        return 0;

}
```

运行后程序显示：error C2248: 'klass::~klass' : cannot access private member declared in class 'klass'

证明析构函数的属性必须为 public。

但是，如果把 klass k 改为 klass* pk; pk=new klass; pk->test();程序通过，但是 klass 不能释放