

所有单选题

1、已知函数原型"void fun(int &a);",则下列关于fun函数的调用形式,正确的是( )。

A: int x=3;fun(x);

B: const int y=3;fun(y);

C: fun(3);

D: 以上三种都正确

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的引用调用

2、已知"int \*p; fun(p);",其中,函数fun没有返回值,且其形参定义为引用调用方式,则下列给出的fun函数原型中正确的是( )。

A: void fun(int &a[]);

B: void fun(int[] &a);

C: void fun(int \*&a);

D: void fun(int \*&a);

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的引用调用

3、已知函数定义"void fun(int

&a) { a+=10; }",则执行"int x=5; fun(x);"后,变量x的值为( )。

A: 5

B: 10

C: 15

D: 不确定

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的引用调用

4、指针函数是指( )。

A: 所有参数都为指针类型的函数

B: 至少有一个参数为指针类型的函数

C: 返回值为指针类型的函数

D: 以上三种都是指针函数

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针函数

5、语句"int \*p0;"的含义是( )

A: 定义p是函数指针变量,p可以指向一个无参数而返回值类型为int的函数

B: 这是一个函数声明,p是函数名,没有参数,返回值类型为整型指针

C: 定义p是一个指向int型的指针变量,可以让p指向一个一维数组

D: 以上说法都不对

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针函数

6、已知"int a[5][3];fun(a);",则下列给出的fun函数原型中正确的是( )。

A: void fun(int (\*p)[5]);

B: void fun(int (\*p)[3]);

C: void fun(int \*p[5]);

D: void fun(int \*p[3]);

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针作为函数参数

7、已知函数调用"fun("abc");",  
则下列给出的fun函数原型中正确  
的是 ( )。

- A: void fun(char);
- B: void fun(char (\*p)[4]);
- C: void fun(char \*p[4]);
- D: void fun(char \*p);

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针作  
为函数参数

8、已知"double x[5];fun(x);",  
fun函数的形参变量名为s,则在  
fun函数的函数体中通过  
sizeof(s)计算得到的值为  
( )。

- A: 40
- B: 20
- C: 5
- D: 4

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针作  
为函数参数

9、已知"char x[5][3];fun(x);",  
fun函数的形参变量名为s,则在  
fun函数的函数体中通过  
sizeof(s[0])计算得到的值为  
( )。

- A: 5
- B: 4
- C: 3
- D: 1

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针作  
为函数参数

10、已知函数原型"double  
fun();",并定义一函数指针变量  
p=fun,则下列调用fun函数的语  
句中错误的是 ( )。

- A: p();
- B: \*p();
- C: (\*p)();
- D: fun();

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数指  
针

11、以下不是重载函数的一组  
函数声明是 ( )。

- A: void f1(int); int f1(int);
- B: void f2(double); int  
f2(float);
- C: void f3(int); void f3();
- D: void f4(double, double);  
void f4(double);

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数重  
载

12、下列有关内联函数的描述  
中,错误的是 ( )。

- A: 内联函数必须是一个小函  
数,代码不能太多
- B: 内联函数中不能含有循环  
语句
- C: 内联函数中不能含有  
switch语句
- D: 内联函数在调用时同样需  
要运行环境的保存、恢复

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/内联函数

13、下列关于内联函数声明位置的描述中,正确的是( )。

A: 有函数声明则必须在函数声明处将函数声明为内联函数, 没有函数声明时才在函数定义处将函数声明为内联函数

B: 只能在函数定义处将函数声明为内联函数

C: 内联函数声明既可以写在函数声明处也可以写在函数定义处, 但只允许写在一处, 即在函数声明处写了在函数定义处就不要再写了, 在函数定义处写了在函数声明处就不要再写了

D: 内联函数声明既可以写在函数声明处也可以写在函数定义处, 并且允许在两处同时写

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/内联函数

14、下列关于静态函数声明位置的描述中,正确的是( )。

A: 有函数声明则必须在函数声明处将函数声明为静态函数, 没有函数声明时才在函数定义处将函数声明为静态函数

B: 只能在函数定义处将函数声明为静态函数

C: 静态函数声明既可以写在函数声明处也可以写在函数定义处, 但只允许写在一处, 即

在函数声明处写了在函数定义处就不要再写了, 在函数定义处写了在函数声明处就不要再写了

D: 静态函数声明既可以写在函数声明处也可以写在函数定义处, 并且允许在两处同时写

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的作用域

15、下列关于函数的描述中,正确的是( )。

A: 外部函数和静态函数都可以在程序的所有源文件中调用

B: 外部函数可以在程序的所有源文件中调用, 静态函数不可以

C: 静态函数可以在程序的所有源文件中调用, 外部函数不可以

D: 以上描述都不正确

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的作用域

16、下列关于默认形参值指定位置的描述中, 正确的是

( )。

A: 有函数声明则必须在函数声明处指定默认形参值, 没有函数声明时才在函数定义处指定

B: 只能在函数定义处指定默认形参值

C: 默认形参值既可以在函数声明处指定, 也可以在函数定义处指定, 但只允许在一处指定, 即在函数声明处指定了在函数定义处就不要再指定了, 在函数定义处指定了在函数声明处就不要再指定了

D: 默认形参值既可以在函数声明处指定也可以在函数定义处指定, 并且允许在两处同时指定

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/默认形参

17、下列关于默认形参值的描述中, 正确的是( )。

A: 只能在函数声明处指定默认形参值

B: 函数调用时, 必须使用默认的参数值, 不能给出新的实参值

C: 函数调用时, 对于有默认值的形参, 如果给出相应的实参, 则函数会优先使用传入的实参值

D: 以上描述都不正确

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/默认形参

18、已知函数原型"void fun(int n, double d=3.0, char \*s="abc");",则下面的函数调用中,不合法的调用是( )。

A: fun(5);

B: fun(5, 8.0);

C: fun(5, 0, "#");

D: fun(5, , );

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/默认形参

19、以下说法正确的是( )。

A: 函数的定义可以嵌套,但是函数的调用不可以嵌套

B: 函数的定义不可以嵌套,但是函数的调用可以嵌套

C: 函数的定义和函数的调用都可以嵌套

D: 函数的定义和函数的调用都不可以嵌套

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的递归调用

20、下列关于非递归方式和递归方式的描述中,正确的是( )。

A: 非递归方式实现的程序比递归方式实现的程序具有更好的可读性

B: 非递归方式实现的程序比递归方式实现的程序运行效率更高

C: 非递归方式实现的程序中不能出现函数调用语句

D: 以上说法都不正确

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的递归调用

21、有关预处理命令的描述中正确的是( )。

A: 预处理命令都是以"&"开头

B: 预处理命令是运行时进行处理的命令

C: 预处理命令是在运行前进行处理的命令

D: 预处理命令必须有include

答案: C

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改

人: admin 审核人:

admin

创建时间: 2011-6-2

最后修改时间: 2011-6-14

知识点: C++部分/编译预处理/编译预处理基本概念

22、有关宏定义的描述中错误的是( )。

A: 宏定义包括无参宏定义和带参宏定义

B: 无参宏定义的功能是定义一个常量,相当于用const定义常量

C: 带参宏定义的功能类似于内联函数

D: 无参宏定义在编译之前进行宏替换,带参宏定义类似于函数在运行时进行调用

答案: D

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改

人: admin 审核人:

admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/编译预处理/宏定义

23、有关文件包含的描述中错误的是( )。

A: #include后面指定包含的文件必须是系统的头文件

B: #include后面指定包含的文件可以是系统的头文件,也可以是自定义的头文件

C: 预处理时,将指定头文件的内容替换这条文件包含命令

D: 包含的文件名用尖括号或用双引号括起决定寻找头文件的方式

答案: A

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改人: admin

审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: CPP部分/编译预处理/文件包含

24、在文件包含预处理语句的使用形式中,当#include后面的文件名用双引号括起时,寻找被包含文件的方式是( )。

A: 直接按系统设定的标准方式搜索目录

B: 先在当前目录搜索,再按系统设定的标准方式搜索

C: 仅仅搜索源程序所在目录

D: 仅仅搜索当前目录

答案: B

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改人: admin

审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: CPP部分/编译预处理/

文件包含

25、在文件包含预处理语句的使用形式中,当#include后面的文件名用尖括号括起时,寻找方式是( )。

A: 直接按系统设定的标准方式搜索目录

B: 先在当前目录搜索,再按系统设定的标准方式搜索

C: 仅仅搜索源程序所在目录

D: 仅仅搜索当前目录

答案: A

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改人: admin

审核人: admin

admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: CPP部分/编译预处理/文件包含

26、以下各项中,不能正确把c1定义成结构体变量的是( )。

A: typedef struct

```
{
    int red;
    int blue;
    int green;
}COLOR;
COLOR c1;
```

B: struct

```
{
    int red;
    int blue;
    int green;
}c1;
```

C: struct Color

```
{
```

```
int red;
int blue;
int green;
}c1;
```

D: struct Color c1

```
{
    int red;
    int blue;
    int green;
};
```

答案: D

评析:

空

等级: 1 难度: 2

区分度: 1

出题人: admin 修改人: admin

审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: CPP部分/构造数据类型/结构体

27、下列关于结构体的说法,错误的是( )。

A: 结构是用户自定义的一种数据类型

B: 结构中可设定若干个不同数据类型的成员

C: 在定义结构时,不可以为成员设置默认值

D: 结构中的成员的数据类型不可以是另一个已经定义的结构

答案: D

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改人: admin

审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: CPP部分/构造数据类型/结构体

28、以下程序的输出结果为：WangJun，则横线处应填入（ ）。

```
#include <iostream>
using namespace std;
struct Student
{
    char num[7];
    char name[10];
    int score;
}
stu1={"100001","LiuKai",87},stu2={"100002","WangJun",92};
int main()
{
    cout<<_____ ;
    return 0;
}
```

- A: stu2.name
- B: Student.name
- C: stu2.name[10]
- D: WangJun

答案：A  
评析：  
空  
等级：1 难度：1  
区分度：1  
出题人：admin 修改人：admin 审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/构造数据类型/结构体

29、假定有"struct Book{char title[40]; float price;}book;"，则正确的语句为（ ）。

- A: Book &x= &book;
- B: Book &x=book;
- C: Book &x=new Book;
- D: Book \*x=Book;

答案：B  
评析：  
空  
等级：1 难度：2  
区分度：1  
出题人：admin 修改人：admin 审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/构造数据类型/结构体

30、设有如下定义：  
struct sk  
{ int a; float b; } data, \*p;  
若有p=&data;，则对data中的a域的正确引用是（ ）。

- A: p.a
- B: (\*p).a
- C: p->data.a
- D: p.data.a

答案：B  
评析：  
空  
等级：1 难度：2  
区分度：1  
出题人：admin 修改人：admin 审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/构造数据类型/结构体

31、设有如下定义：  
struct sk  
{ int a; float b; } data, \*p;  
若有p=&data;，则对data中的a域的不正确引用是（ ）。

- A: \*p.a
- B: (\*p).a

C: p->a

D: data.a

答案：A  
评析：  
空  
等级：1 难度：2  
区分度：1  
出题人：admin 修改人：admin 审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/构造数据类型/结构体

32、下列对结构及其变量定义错误的是（ ）。

- A: struct Student  
{  
 int m\_num;  
 char m\_name[10];  
};
- B: struct Student  
{  
 int m\_num;  
 char m\_name[10];  
}my;
- C: struct  
{  
 int m\_num;  
 char m\_name[10];  
};
- D: struct  
{  
 int m\_num;  
 char m\_name[10];  
}Student;

答案：C  
评析：  
空  
等级：1 难度：1  
区分度：1  
出题人：admin 修改人：admin 审核人：admin  
创建时间：2011-6-2

最后修改时间: 2011-6-10  
知识点: C++部分/构造数据类型/结构体

33、下面对枚举类型的描述正确的是( )。

A: 枚举类型的定义为:  
`enum{Monday,Tuesday,Wednesday,Thursday,Friday}Day;`

B: 枚举类型的定义为: `enum Day{Monday,Tuesday,Wednesday=5,Thursday,Friday=5};`

C: 在C++语言中, 用户自定义的枚举类型的第一个常量的默认值是1

D: 以上说法都不正确

答案: B

评析:

空

等级: 1 难度: 2

区分度: 1

出题人: admin 修改人: admin

审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-14

知识点: C++部分/构造数据类型/枚举

34、定义如下枚举类型:

```
enum Day{Monday,Tuesday,Wednesday,Thursday,Friday=2};
```

则下列语句正确的是( )。

A: `Day day;day=Monday;`

B: `Day day;day=3;`

C: `Day day;day=Monday+3;`

D: 表达式 `Wednesday==Friday` 的值是false

答案: A

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改人: admin

审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/枚举

35、`"typedef int LONG;"` 的作用是( )。

A: 定义了一个长整型变量

B: 定义了一个整型变量

C: 定义了一个新的数据类型标识符

D: 定义了一种新的数据类型

答案: C

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改人: admin

审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/类型重定义

36、下列关于成员访问权限的描述中, 不正确的是( )。

A: 公有数据成员和公有函数成员都可以被类对象直接处理

B: 类的私有数据成员只能被同类的公有成员函数以及该类的友元类或友元函数访问

C: 只有类或派生类的成员函数和友元类或友元函数可以访问保护成员

D: 保护成员在派生类中可以被派生类的函数成员访问, 而私有成员不可以

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-14

知识点: C++部分/类与对象/类和对象的声明与定义

37、下列关于构造函数的描述中, 错误的是( )。

A: 构造函数可以设置默认参数

B: 构造函数在定义类变量时被自动执行

C: 构造函数可以对静态数据成员进行初始化

D: 构造函数可以重载

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造析构函数作用和声明

38、对于下面定义类 `MyClass`, 在函数 `f()` 中将对象成员 `n` 的值修改为50的语句应该是( )。

```
class MyClass
```

```
{
```

```
public:
```

```
    MyClass(int x)
```

```
{
```

```
    n=x;
```

```
}
```

```

void SetValue(int n1)
{
    n=n1;
}
private:
    int n;
};
int f()
{
    MyClass *ptr=new
MyClass(45);
    _____;
}

```

- A: MyClass(50)
- B: SetValue(50)
- C: ptr->SetValue(50)
- D: ptr->n=50

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/类和对象的声明与定义

39、下列关于类与对象的说法中, ( ) 是不正确的。

- A: 对象是类的一个实例
- B: 任何一个对象只能属于某个类
- C: 一个类只能有一个对象
- D: 类与对象的关系和数据类型与变量的关系相似

答案: C

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/类和对象的声明与定义

40、以下不是构造函数的特征的是 ( )。

- A: 构造函数的函数名与类名相同
- B: 构造函数可以重载
- C: 构造函数可以设置默认参数
- D: 构造函数可以指定返回类型说明

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造析构函数作用和声明

41、有关析构函数的说法不正确的是 ( )。

- A: 析构函数有且只有一个
- B: 析构函数无任何返回类型
- C: 析构函数可以有形参
- D: 析构函数的作用是在对象被删除前进行一些需要的清理工作

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造析构函数作用和声明

42、下列有关类的说法, ( ) 是不正确的。

- A: 类是一种用户自定义的数据类型
- B: 只有类中的成员函数才能存取类中的私有数据
- C: 在类中如果不作特别声明, 所有数据成员均为私有类型
- D: 类的成员包括成员函数和成员数据

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/类和对象的声明与定义

43、假定MyClass为一个类名, 那么下列的函数说明中, ( ) 为该类的析构函数的原型

- A: ~MyClass;
- B: ~MyClass(int n);
- C: MyClass();
- D: ~MyClass();

答案: D



评析：  
略  
等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/构造析构函数作用和声明

44、若有一个MyClass类，则执行语句MyClass a, b(2), \*p;后，自动调用该类的构造函数( )次。

- A: 1
- B: 2
- C: 3
- D: 4

答案：B  
评析：  
略  
等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/构造析构函数作用和声明

45、下列关于成员函数特征的描述中，( )是错误的。

- A: 成员函数一定是内联函数
- B: 成员函数可以重载
- C: 成员函数可以设置参数的默认值
- D: 成员函数可以是静态的

答案：A  
评析：

略  
等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类和对象的声明与定义

46、友元的作用是( )。

- A: 提高程序的运行效率
- B: 加强类的封装性
- C: 实现数据的隐藏
- D: 增加成员函数的种类

答案：A  
评析：  
略  
等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/友元

47、一个类的友元函数或友元类能够访问该类的( )。

- A: 私有成员
- B: 保护成员
- C: 公有成员
- D: 公有成员、保护成员和私有成员

答案：D  
评析：  
略  
等级：1                      难度：1  
区分度：1  
出题人：文本导入

修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/友元

48、下面对于静态成员的描述中，正确的是( )。

- A: 静态数据成员是类的所有对象共享的数据
- B: 类的每个对象都有自己的静态数据成员
- C: 类的不同对象有不同的静态数据成员值
- D: 静态数据成员不能通过类的对象访问

答案：A  
评析：  
略  
等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类的静态成员

49、下面对静态数据成员的描述中，( )是正确的。

- A: 静态数据成员可以在类体内进行初始化
- B: 静态数据成员不可以被类的对象来访问
- C: 静态数据成员不能设置为protected
- D: 静态数据成员可以直接用类名来访问

答案：D  
评析：

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/类的静态成员

50、( )是析构函数的特征。

A: 一个类中只能定义一个析构函数

B: 析构函数名与类名相同

C: 析构函数的定义只能在类声明体内

D: 析构函数可以有一个或多个参数

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造函数重载

51、下列的各类函数中,( )不是类的成员函数。

A: 构造函数

B: 析构函数

C: 友元函数

D: 拷贝构造函数

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造函数重载

52、关于拷贝构造函数的描述正确的是( )。

A: 通常的拷贝构造函数的参数是指针类型

B: 如果不自定义拷贝构造函数, 系统提供默认的拷贝构造函数

C: 如果有自定义的构造函数, 系统就不再提供拷贝构造函数

D: 如果需要用已有对象为新创建的对象初始化时, 就必须自定义拷贝构造函数

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/拷贝构造函数

53、下面关于友元函数的描述中,( )是正确的

A: 友元函数的实现必须在类的内部定义

B: 友元函数是类的成员函数

C: 友元函数破坏了类的封装性和隐藏性

D: 友元函数不能访问类的私有成员

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/友元

54、静态数据成员初始化在( )进行, 而且前面不加static, 以免与一般静态变量或对象相混淆。

A: 类体内

B: 类体外

C: 构造函数内

D: 内联函数内

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/类的静态成员

55、在下面的程序中, 错误的语句是( )。

```
class Sample
{
public:
    Sample(int val); //A
    ~Sample(); //B
private:
    int a=2; //C
```

```
float f;      //D
};
```

- A: A  
B: B  
C: C  
D: D

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造  
函数重载

56、在公有派生情况下, 有关  
派生类对象和基类对象的关系,  
下列叙述不正确的是()

- A: 派生类的对象可以直接访问  
基类的所有成员  
B: 派生类的对象可以初始化  
基类的引用  
C: 派生类的对象不可以直接  
访问基类中的私有成员  
D: 派生类的对象的地址可以  
赋给指向基类的指针

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/继承的  
基本概念

57、下列对派生类的描述中错  
误的是 ( )。

- A: 派生类至少有一个基类  
B: 派生类可作为另一个派生  
类的基类  
C: 派生类除了包含它直接定  
义的成员外, 还包含其基类的  
成员

D: 派生类对象的地址不可以  
赋给指向基类的指针

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/继承的  
基本概念

58、下面描述中, 错误的是 ( )

- A: 公有继承时, 基类中的  
public成员在派生类中仍是  
public的  
B: 公有继承时, 基类中的  
private成员在派生类中仍是  
private的  
C: 公有继承时, 基类中的  
protected成员在派生类中仍是  
protected的

D: 私有继承时, 基类中的  
public成员在派生类中是private  
的

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/继承方  
式

59、基类的 ( ) 在私有继承时  
在派生类中为私有成员函数,  
在公有和保护继承时在派生类  
中为保护成员函数。

- A: 私有成员函数  
B: 保护成员函数  
C: 公有成员  
D: 保护数据成员

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/继承方  
式

60、派生类可访问它基类成员  
的描述中, 正确的是 ( )

- A: 派生类是否可访问它基类  
中的成员与继承方式无关  
B: 派生类不可访问它基类中  
的成员

C: 私有继承方式下派生类可  
访问它基类的私有成员

D: 公有继承方式下派生类可  
访问它基类的私有成员

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/继承/继承方式

61、派生类对象可访问的基类成员是 ( )

- A: 公有继承方式基类的公有成员
- B: 公有继承方式基类的保护成员
- C: 保护继承方式基类的公有成员
- D: 保护继承方式基类的保护成员

答案: A  
评析: 略  
等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/继承/继承方式

62、下列描述不正确的是 ( )

- A: 任何继承方式基类中私有成员派生类都不可访问
- B: 保护继承方式基类中公有成员派生类对象可访问
- C: 派生类中包含了基类中的所有成员
- D: 私有继承方式基类中公有成员可被直接派生类访问, 而不可被间接派生类访问

答案: B  
评析: 略  
等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/继承/继承方式

63、下列关于继承的描述中, 错误的是 ( )

- A: 析构函数不能被重载
- B: 派生类所表示的事物是基类所表示事物的子集
- C: 派生类的成员除了它自己的成员外, 还包括了它的基类成员
- D: 基类成员的访问权限到派生类保持不变

答案: D  
评析: 略  
等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-14  
知识点: C++部分/继承/继承的基本概念

64、派生类的派生类不可访问的基类成员是 ( )

- A: 先私有继承方式再公有继承方式的基类中公有成员和保护成员
- B: 先公有继承方式再私有继承方式的基类中公有成员和保护成员

C: 先公有继承方式再保护继承方式的基类中公有成员和保护成员

D: 先保护继承方式再公有继承方式的基类中公有成员和保护成员。

答案: A  
评析: 略  
等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/继承/继承方式

65、设类B是基类A的派生类, 并有"A aa, \*pa=&aa; B bb, \*pb=&bb;"语句, 则下列语句正确的是 ( )

- A: pb=pa;
- B: bb=aa;
- C: aa=bb;
- D: \*pb=\*pa;

答案: C  
评析: 略  
等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/继承/类型兼容

66、下列虚基类的声明中, 正确的是 ( )。

- A: class virtual B:public A

B: virtual class B:public A

C: class B:public A virtual

D: class B:virtual public A

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/多重继承

67、创建派生类对象时构造函数的调用顺序是先调用( )。

A: 派生类的构造函数

B: 基类的构造函数

C: 派生类的析构函数

D: 基类的析构函数

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/派生类的构造函数

68、多继承的构造函数顺序可分为如下4步:

①所有非虚基类的构造函数按照它们被继承的顺序构造。

②所有虚基类的构造函数按照它们被继承的顺序构造。

③所有子对象的构造函数按照它们声明的顺序构造。

④派生类自己的构造函数体

这四个步骤的正确顺序是( )。

A: ④③①②

B: ②④③①

C: ②①③④

D: ③④①②

答案: C

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/多重继承

69、下列关于派生类构造函数的描述中, 错误的是( )

A: 派生类构造函数成员初始化列表中, 应包含直接基类和所有间接基类的构造函数的调用

B: 派生类构造函数的成员初始化列表中, 仅包含直接基类构造函数和其他(如子对象)类构造函数的调用

C: 派生类构造函数通常带有成员初始化列表

D: 派生类默认构造函数中隐含直接基类的无参构造函数的调用

答案: A

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-14

知识点: C++部分/继承/派生类的构造函数

70、下列有关类型的描述中, 错误的是( )

A: 类型兼容规则是在公有继承下才成立

B: 公有继承下, 基类对象可赋值给派生类对象

C: 公有继承下, 派生类对象可赋值给基类对象

D: 在保护继承方式下类型兼容是不成立的

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/类型兼容

71、下列关于多重继承的二义性描述中, 错误的是( )

A: 不是所有情况的多重继承都会出现二义性

B: 多重继承中出现的二义性使用虚基类的方式可以消除

C: 派生类中对它多个基类中的同名成员访问时可能出现二义性

D: 程序中出现二义性是不必消除的

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/多重继承

72、下列关于虚基类的描述中, 错误的是 ( )

A: 虚基类需要用关键字 virtual 说明

B: 带有虚基类的类的派生类的构造函数与不带有虚基类的类的派生类的构造函数没有区别

C: 使用虚基类可以解决公有基类成员的二义性问题

D: 虚基类后继类层次中的类都需要对虚基类进行初始化

答案: B

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-14

知识点: C++部分/继承/多重继承

73、派生类的构造函数的成员初始化列表中, 不能包含 ( )

A: 基类的构造函数

B: 派生类中子对象的初始化

C: 派生类中一般数据成员的初始化

D: 基类的子对象初始化

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/派生类的构造函数

74、下列关于继承的描述中, 正确的是()

A: 继承不是类之间的一种关系

B: C++语言仅支持单一继承

C: 继承会增加程序的冗余性

D: 继承是面向对象方法中一个很重要的特性

答案: D

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/继承/继承的基本概念

75、下列关于虚函数的描述中, 正确的是 ( )。

A: 虚函数是一个 static 类型的成员函数

B: 虚函数既可以是成员函数, 也可以是成员函数

C: 基类中说明一个虚函数后, 派生类中定义相同原型的函数可以不必加 virtual 说明

D: 派生类中的虚函数与基类

中相同原型的虚函数具有不同的参数个数或类型

答案: C

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/多态性/虚函数的工作方式

76、实现多态性的技术是 ( )。

A: 抽象类

B: 前期绑定

C: 友元函数

D: 动态绑定

答案: D

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/多态性/多态性的概念

77、下列关于虚函数的说明中, 正确的是 ( )。

A: 从虚基类继承的函数都是虚函数

B: 虚函数不得是静态成员函数

C: 只能通过指针或引用调用虚函数

D: 抽象类中的成员函数都是虚函数

答案: B  
评析:

等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/多态性/虚函数的工作方式

78、在派生类中重新定义虚函数时,除了( ),其他方面都应基类中相应的虚函数保持一致。

- A: 函数名
- B: 参数
- C: 函数体
- D: 返回值类型

答案: C  
评析:

等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/多态性/虚函数的工作方式

79、关于纯虚函数和抽象类的描述中,错误的是( )。

- A: 纯虚函数是一种特殊的虚函数,它没有具体的实现
- B: 抽象类是指具有纯虚函数的类
- C: 一个基类中说明有纯虚函数

数,该类的派生类一定不再是抽象类

D: 抽象类作为基类来使用,其纯虚函数的实现由派生类给出

答案: C  
评析:  
等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/多态性/抽象类的基本概念

80、虚函数必须是类的( )。

- A: 友元函数
- B: 成员函数
- C: 构造函数
- D: 析构函数

答案: B  
评析:

等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/多态性/虚函数的概念

81、多态性提高了系统的( )。

- A: 一致性
- B: 灵活性
- C: 可维护性

D: 包括以上三项

答案: D  
评析:

等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/多态性/多态性的概念

82、无论是虚函数还是实函数,在派生类被重定义后,原来的函数版本会( )。

- A: 自动删除
- B: 被新函数覆盖
- C: 被隐藏
- D: 无法操作

答案: B  
评析:

等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/多态性/虚函数的工作方式

83、抽象类的主要作用是( )。

- A: 直接用作基类
- B: 直接用作指针
- C: 引用的基类型
- D: 包括以上各项

答案: D

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/多态性/抽象类的工作方式

84、假定要对类AB定义加号运算符重载成员函数,实现两个AB类对象的加法,并返回相加结果,则该成员函数的声明语句为( )。

A: AB operator+(AB &a,AB &b)

B: AB operator+(AB &a)

C: operator+(AB a)

D: AB &operator+()

答案: B

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/运算符重载方法

85、关于运算符重载,下面的叙述错误的是( )。

A: 运算符预定义的操作数个数不能改变

B: 重载不能改变运算符的优先级顺序

C: 参数的类型没有限制

D: 尽量使其与原意保持一致

答案: C

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/运算符重载的基本概念

86、在一个类中可以对一个运算符进行( )重载。

A: 1种

B: 2种以下

C: 3种以下

D: 多种

答案: D

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/运算符重载的基本概念

87、重载赋值操作符时,应声明为( )函数。

A: 友元

B: 虚

C: 成员

D: 多态

答案: C

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/特殊运算符重载

88、要在类对象使用运算符,以下不必被重载的运算符是( )。

A: []

B: =

C: ++

D: --

答案: B

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/运算符重载的基本概念

89、下列运算符中,不能重载的是( )。

A: new

B: ()

C: ::

D: &&

答案: C

评析:

等级: 1 难度: 1



区分度: 1

出题人: 文本导入

修改人: admin

审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/  
运算符重载的基本概念

90、在表达式 $x+y*z$ 中,  $+$  是作为成员函数重载的运算符,  $*$ 是作为非成员函数重载的运算符。下列叙述中正确的是( )。

A:  $\text{operator}+$ 有两个参数,  
 $\text{operator}*$ 有两个参数

B:  $\text{operator}+$ 有两个参数,  
 $\text{operator}*$ 有一个参数

C:  $\text{operator}+$ 有一个参数,  
 $\text{operator}*$ 有两个参数

D:  $\text{operator}+$ 有一个参数,  
 $\text{operator}*$ 有一个参数

答案: C

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin

审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/  
运算符重载的基本概念

91、友元运算符 $\text{obj1} > \text{obj2}$ 被  
C++编译器解释为( )。

A:  $\text{operator}>(\text{obj1}, \text{obj2})$

B:  $>(\text{obj1}, \text{obj2})$

C:  $\text{obj1.operator}>(\text{obj2})$

D:  $\text{obj2.operator}>(\text{obj1})$

答案: A

评析:

等级: 1

难度: 1

区分度: 1

出题人: 文本导入

修改人: admin

审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/  
特殊运算符重载

92、已知某个类的友元函数重载了 $+=$ 和 $-$ ,  $a$ 、 $b$ 、 $c$ 是该类的对象, 则 $\text{"a+=b-c"}$ 被C++编译器解释为( )。

A:  
 $\text{operator}+=(a, \text{operator}-(b, c))$

B:  
 $a.\text{operator}+=(b.\text{operator}-(c))$

C:  
 $\text{operator}+=(a, b.\text{operator}-(c))$

D:  
 $a.\text{operator}+=(\text{operator}-(b, c))$

答案: A

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin

审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/  
运算符重载方法

93、下列运算符中, 必须使用成员函数进行重载的是( )。

A:  $==$

B:  $=$

C:  $>>$

D:  $++$

答案: B

评析:

空

等级: 1

难度: 1

区分度: 1

出题人: 文本导入

修改人: admin

审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/运算符重载/  
运算符重载规则

94、 $\text{cout}$ 是输入/输出流库预定义的( )。

A: 类

B: 对象

C: 常量

D: 包含文件

答案: B

评析:

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin

审核

人: admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/输入输出流/  
输入输出流简介

95、4个预定义流对象中不支持缓冲的用于出错信息标准输出的是( )。

A:  $\text{cin}$

B:  $\text{cout}$

C:  $\text{cerr}$

D:  $\text{clog}$

答案: C

评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流简介

96、下面哪个选项是istream类的对象 ( )。

- A: cin
- B: cout
- C: cerr
- D: clog

答案: A  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流简介

97、下面4个选项中, 专门用于读取单个字符的是 ( )。

- A: cin.read()
- B: cin.get()
- C: cin.put()
- D: cin.getline()

答案: B  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin

人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流的成员函数

98、下面输出字符变量k的方法中, 正确的是 ( )。

- A: cout<<put(k);
- B: cout<<'k';
- C: cout.put(k);
- D: cout.put('k');

答案: C  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流的成员函数

99、已知char str[20];cin.getline(str,6);从键盘输入abc def,则cout<<str;的输出结果为 ( )。

- A: abc def
- B: abc de
- C: abc d
- D: abc

答案: C  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2

最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流的成员函数

100、下列关于read()函数的叙述中, 正确的是 ( )。

A: read()函数是只用于从键盘输入字符串

B: read()函数所获取的字符多少是不受限制的

C: read()函数只能用于对文本文件的操作

D: read()函数只能按照规定读取所指定的字符数

答案: D  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流的成员函数

101、C++程序进行文件操作时, 需要包含头文件 ( )。

- A: iostream
- B: fstream
- C: cmath
- D: cstdlib

答案: B  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-2

最后修改时间：2011-6-10  
知识点：CPP部分/文件/文件与流

102、定义fstream的流对象并打开文件时，默认的打开方式为（ ）。

- A: ios::in
- B: ios::out
- C: ios::in|ios::out
- D: 没有默认方式，必须指定

答案：C  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/文件/文件的打开与关闭

103、下列语句中不能在C盘根目录下新建文件test.txt的为（ ）。

- A: ofstream fout("c:\\test.txt");
- B: fstream fout;  
fout.open("c:\\text.txt");
- C: fstream fout;  
fout.open("c:\\text.txt",ios::out);
- D: fstream fout;  
fout("c:\\text.txt");

答案：D  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin

创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/文件/文件的打开与关闭

104、打开文件的方式中，（ ）以追加方式打开文件。

- A: in
- B: out
- C: app
- D: ate

答案：C  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/文件/文件的打开与关闭

105、下列函数中，可用于向文本文件中进行读操作的是（ ）。

- A: get
- B: put
- C: write
- D: open

答案：A  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-14  
知识点：CPP部分/文件/文本文件的顺序读写

件的顺序读写

106、已知流对象fin和某文本文件关联，用于只读操作，下面说法错误的是（ ）。

A: fin.get()用于从文件中当前位置读取一个字符，然后文件指针后移一个字符。

B: fin.get()用于从文件中当前位置读取一个字符，而后文件指针位置不动。

C: 已知：char ch;则fin.get(ch)、ch=fin.get()、fin>>ch;都能实现从文件中读取字符赋值给ch。

D: 当文件指针指向EOF时，读取失败。

答案：B  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/文件/文本文件的顺序读写

107、有关二进制文件读写说法正确的是（ ）。

A: read()和write()函数只能用于二进制文件读写，不能用于文本文件读写。

B: read()和write()函数只能用于文本文件读写，不能用于二进制文件读写。

C: read()和write()函数既能用于二进制文件读写，也能用于文本文件读写。

D: 以上说法都不对

答案: C

评析:

等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/二进制文件的顺序读写

108、已知outfile是一个输出流对象,要想将outfile的文件指针定位到当前位置之前32个字节处,正确的函数调用语句是( )。

- A: outfile.seekp(32,ios::cur);
- B: outfile.seekp(32,ios::beg);
- C: outfile.seekp(-32,ios::beg);
- D: outfile.seekp(-32,ios::cur);

答案: D

评析:

等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的随机读写

109、下列说法正确的是( )。

- A: 使用seekg()函数时,默认参照位置为ios::beg
- B: 使用seekg()函数时,默认参照位置为ios::cur
- C: 使用seekg()函数时,默认参照位置为ios::end

D: 以上说法都不对

答案: A

评析:

空  
等级: 1                      难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的随机读写

110、( )是对不同数据类型完成相同操作的函数的描述。

- A: 重载函数
- B: 模板函数
- C: 函数模板
- D: 递归函数

答案: C

评析:

空  
等级: 1                      难度: 1  
区分度: 1  
出题人: admin                      修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-14  
知识点: C++部分/模板/函数模板定义

111、模板函数的执行代码是编译器发现( )时生成的。

- A: 执行函数
- B: 源程序中声明函数
- C: 源程序中定义函数
- D: 源程序中调用函数

答案: D

评析:

空

等级: 1                      难度: 1  
区分度: 1  
出题人: admin                      修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-14  
知识点: C++部分/模板/函数模板定义

112、下列有关函数模板和模板函数的说法中错误的是( )。

A: 函数模板只是对函数的描述,编译器不为其产生任何执行代码,所以它不是一个实实在在的函数

B: 模板函数是实实在在的函数,它由编译系统在遇到具体函数调用时生成,并调用执行

C: 函数模板需要实例化为模板函数后才能执行

D: 当函数模板和一般函数同名时,系统先去匹配函数模板,将其实例化后进行调用

答案: D

评析:

空  
等级: 1                      难度: 1  
区分度: 1  
出题人: admin                      修改人: admin                      审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/函数模板定义

113、下面函数模板的定义中,有( )处错误。

```
Template<class T1,T2>
T2 fl(T1 a,b)
{
    return (a>b) ? (a) : (b);
}
```

A: 1  
B: 2  
C: 3  
D: 4  
答案: B  
评析:  
空  
等级: 1 难度: 2  
区分度: 1  
出题人: admin 修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/函数模板定义

114、( ) 是对一批仅成员数据类型不同的类的抽象。

- A: 函数模板
- B: 模板函数
- C: 类模板
- D: 模板类

答案: C  
评析:  
空  
等级: 1 难度: 1  
区分度: 1  
出题人: admin 修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-14  
知识点: C++部分/模板/类模板定义

115、类模板的模板参数 ( )

- A: 只可作为数据成员的类型
- B: 只可作为成员函数的返回类型

C: 只可作为成员函数的参数类型

D: 以上三项均可以

答案: D  
评析:  
空  
等级: 1 难度: 1  
区分度: 1  
出题人: admin 修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

116、下面对模板的声明中, 正确的是 ( )

- A: `template<T>`
- B: `template <class T1,T2>`
- C: `template <class T1,class T2>`
- D: `template<class T1;class T2>`

答案: C  
评析:  
空  
等级: 1 难度: 1  
区分度: 1  
出题人: admin 修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

117、关于类模板, 描述错误的是 ( )

- A: 类模板以成员类型为参数
- B: 类模板可以直接使用
- C: 根据声明对象时的实际数

据类型, 编译器把类模板实例化为模板类

D: 模板类可以用来作函数的参数

答案: B  
评析:  
空  
等级: 1 难度: 1  
区分度: 1  
出题人: admin 修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-14  
知识点: C++部分/模板/类模板定义

118、以下关于函数模板叙述正确的是 ( )

- A: 函数模板也是一个具体类型的函数
- B: 函数模板的类型参数与函数的参数是同一个概念
- C: 通过使用不同的类型参数, 函数模板可以生成不同类型的函数
- D: 用函数模板定义的函数没有类型

答案: C  
评析:  
空  
等级: 1 难度: 1  
区分度: 1  
出题人: admin 修改人: admin 审核人: admin  
创建时间: 2011-6-2  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/函数模板定义

119、有如下函数模板定义:  
`template <class T>`  
`T func(T x, T y)`

```
{ return x*x*x+y*y*y; }
```

在下列对func的调用中,错误的是( )。

A: func(3, 5);

B: func(3.0, 5.5);

C: func (3, 5.5);

D: func<int>(3, 5.5);

答案: C

评析:

空

等级: 1 难度: 3

区分度: 1

出题人: admin 修改

人: admin 审核人:

admin

创建时间: 2011-6-2

最后修改时间: 2011-6-10

知识点: C++部分/模板/函数模板使用

120、有关二进制文件读写说法正确的是( )。

A: 二进制文件的扩展名必须是.dat

B: 只有数值型数据使用二进制文件进行存储

C: 要用ios::binary指定二进制文件的打开方式

D: 只有结构体数据使用二进制文件进行存储

答案: C

评析:

空

等级: 1 难度: 1

区分度: 1

出题人: admin 修改

人: admin 审核人:

admin

创建时间: 2011-6-14

最后修改时间: 2011-6-14

知识点: C++部分/文件/二进制

文件的顺序读写

所有是非题

1、已知函数原型"void fun(int &a);",则调用fun函数时系统需要为形参a分配内存空间。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数的引用调用

2、只有返回引用的函数可以作为赋值语句的操作数。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/函数/返回引用的函数

3、函数返回的指针可以是静态局部变量的地址。

答案: 正确

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针函数

4、将数组作为函数实参时,则在函数中可以操作数组中的元素并更改元素的值。

答案: 正确

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-14

知识点: C++部分/函数/指针作为函数参数

5、一个函数的函数名可以作为另一个函数的实参。

答案: 正确

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/函数/指针作为函数参数

6、已知函数原型"double fun();",并在主函数中定义一函数指针变量p=fun,则主函数之前必须给出fun函数的定义。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/函数/函数指针

7、函数"void f1(int);"和函数"int f2();"满足函数重载的条件，所以它们是重载函数。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/函数重载

8、函数"void fun(double);"和函数"int fun(int, double=3.0);"满足函数重载的条件，所以它们是重载函数。

答案： 正确

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-14

知识点：CPP部分/函数/函数重载

9、重载函数在调用时由系统根据函数的返回值类型自动确定调用哪一个重载函数。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-14

知识点：CPP部分/函数/函数重载

10、若多个重载函数的形式与函数调用形式匹配，则在执行函数调用语句时会按照重载函数的定义顺序依次执行各重载函数。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/函数重载

11、任何一个带默认形参值的函数都可以改用一组不带默认形参值的重载函数来实现。

答案： 正确

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/函数重载

12、使用内联函数关键字将一个函数定义为内联函数后，系统在编译时肯定会将其作为内联函数处理。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/内联函数

13、将功能比较简单且被频繁调用的函数定义为内联函数可以提高程序的执行效率。

答案： 正确

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/内联函数

14、静态函数不可以声明为内联函数。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/函数的作用域

15、静态函数在调用时不需要进行运行环境的保存、恢复。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/函数的作用域

16、默认形参值可以通过函数调用指定。

答案： 正确

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/默认形参

17、默认形参值可以通过全局变量指定。

答案： 正确

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/默认形参

18、默认形参值可以通过静态局部变量指定。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/默认形参

19、已知函数原型"void fun(int

a, int b=0);", 则通过"fun();"调用fun函数时, 形参a和b的值都为0。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/默认形参

20、递归方式实现的函数必须有返回值, 即函数返回类型不能为void。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/函数/函数的递归调用

21、编译预处理与注释一样, 编译器不会将其编译成目标代码。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/编译预处理/编译预处理基本概念

22、顾名思义, 编译之前的处理过程就称为编译预处理。

答案： 正确

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/编译预处理/编译预处理基本概念

23、编译预处理命令都是以"#"开头, 并且命令末尾有分号。

答案： 错误

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/编译预处理/编译预处理基本概念

24、可以声明带参数宏定义, 也可以声明不带参数宏定义。

答案： 正确

评析：

略

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-1

最后修改时间：2011-6-10

知识点：CPP部分/编译预处理/宏定义

25、带参数的宏定义与函数一样, 在程序运行时, 程序流程都要发生改变。



答案： 错误  
评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/编译预处理/  
宏定义

26、条件编译是在程序运行时，  
运行满足条件的代码。

答案： 错误  
评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/编译预处理/  
条件编译

27、条件编译命令只能在函数  
外出现，不能在函数中使用条  
件编译命令。

答案： 错误  
评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/编译预处理/  
条件编译

28、文件包含是将程序需要的  
头文件包含在当前文件中。

答案： 正确

评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-14  
知识点：CPP部分/编译预处理/  
文件包含

29、#include后面指定包含的文  
件必须是系统的头文件

答案： 错误  
评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/编译预处理/  
文件包含

30、在头文件中使用条件编译  
命令#ifndef，可以避免头文件  
被重复包含。

答案： 正确  
评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-14  
知识点：CPP部分/编译预处理/  
多文件结构

31、定义结构体类型的关键字  
是struct，整个定义的最后以"}"  
结束。

答案： 错误  
评析：

略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/构造数据类  
型/结构体

32、结构体类型的定义一般放  
在程序开始的文件包含语句后  
面，不可以放到某个函数内部。

答案： 错误  
评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/构造数据类  
型/结构体

33、结构体中的成员也可以是  
结构体类型。

答案： 正确  
评析：  
略  
等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核  
人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/构造数据类  
型/结构体

34、结构体中的成员不是变量，  
不能单独使用，它们从属于某  
个结构体变量，必须通过"结构  
体变量名.成员名"的形式来引  
用。

答案： 正确

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/结构体

35、可以定义结构体的数据,但结构体类型的数据不可以作为函数参数。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/结构体

36、枚举常量默认值从1开始,依次增1。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/枚举

37、下面的语句:

```
enum
```

```
Color{red,green,blue,white,black};
```

```
co1=red,co2;
```

```
co2=2;
```

是正确的。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/枚举

38、在C++中,可以使用typedef关键字为已有的数据类型定义一个别名,增强程序的可读性。

答案: 正确

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/类型重定义

39、使用typedef对程序的通用和移植没有帮助。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/类型重定义

40、下面的语句:

```
typedef int Array[10];
```

```
Array array;
```

是正确的。

答案: 正确

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/构造数据类型/类型重定义

41、如果一个类没有申请动态内存空间,则可以不定义析构函数。

答案: 正确

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造析构函数作用和声明

42、系统提供的缺省构造函数没有参数,所以自定义构造函数必须带有参数。

答案: 错误

评析:

略

等级: 1 难度: 1

区分度: 1

出题人: 文本导入

修改人: admin 审核

人: admin

创建时间: 2011-6-1

最后修改时间: 2011-6-10

知识点: C++部分/类与对象/构造析构函数作用和声明

43、面向对象程序设计的英文缩写为OOP。

答案： 正确

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类和对象的声明与定义

44、除了私有成员以外，一个类中的其他成员都可以在任何地方访问。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类和对象的声明与定义

45、通过类对象访问类成员需要用"-"进行访问。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类和对象的声明与定义

46、除void类型以外，其它任何类型不能作为构造函数的返回类型。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/构造析构函数作用和声明

47、有类Sample的两个对象s1和s2，如果要执行语句"s1=s2"，则Sample类内必须要定义拷贝构造函数。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/构造析构函数作用和声明

48、对象是类的实例。

答案： 正确

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类和对象的声明与定义

49、在类的定义的内部定义的函数体，全部作为内联函数出现。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类和对象的声明与定义

50、类的静态数据成员不能通过对象来访问。

答案： 错误

评析：

略

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/类的静态成员

51、类的构造函数可以通过返回类型或参数的个数与类型的不同进行重载。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/类与对象/构造函数重载

52、静态成员函数中不能访问非静态成员。

答案： 正确

评析：

等级：1                      难度：1

区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/类的静态成员

53、如果类A是类B的友类,那么类A中的所有成员函数都可以访问类B中的所有成员。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/友元

54、类的静态成员属于类,而不属于任何一个对象。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/类的静态成员

55、类的非静态成员函数不可以访问类的静态成员。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入

修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/类的静态成员

56、友元提供了一般函数与类的成员之间、不同类的成员之间进行数据共享的机制。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/友元

57、this指针是一个隐含于每一个成员函数中的特殊指针。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/this指针

58、this指针是一个隐含的指针,不能显式地引用this指针。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin

创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/this指针

59、静态数据成员不能是私有的。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/类与对象/类的静态成员

60、私有继承中,对于基类中的所有成员,派生类的成员函数都不可直接访问。

答案: 错误  
评析: 略

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/继承/继承方式

61、派生类至少有一个基类。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/继承/继承的

## 基本概念

62、派生类对象可以出现在任何基类对象可以出现的地方。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/类型兼容

63、当用公有继承从基类派生一个类时，基类的保护成员为派生类的公有成员。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承方式

64、C++语言中，派生类只能有一个基类。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承的基本概念

65、派生类只能继承基类的公有成员和保护成员，不能继承私有成员。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承的基本概念

66、在保护继承的方式下，派生类可以访问基类成员的保护成员。

答案： 正确

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承方式

67、在私有继承方式下，派生类的对象可以访问基类中的公有成员。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承方式

68、一个派生类可以作另一派生类的基类。

答案： 正确

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承的基本概念

69、派生类的成员除了它自己的成员外，还包含了它的基类的成员。

答案： 正确

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承的基本概念

70、在C++中，允许从一个类派生出任意多个类，但不允许某个类同时从多个类派生。

答案： 错误

评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/多重继承

71、构造函数不能被派生类继承。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/派生类的构造函数

72、如果基类的构造函数不带参数，则定义一个派生类对象时，就不会调用基类的构造函数。

答案： 错误  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-14  
知识点：CPP部分/继承/派生类的构造函数

73、基类数据成员的初始化只能在基类中完成。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/派生类的构造函数

74、在定义派生类的构造函数

的语句中不一定包含基类的构造函数。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/派生类的构造函数

75、类B从类A私有派生，类C从类B派生，不管类C如何继承于类B，从此以后类C的成员函数不可以访问A类的任何成员。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/继承方式

76、根据类型兼用规则，公有派生类对象可以赋值给基类的引用，这时，通过引用可以调用在派生类中新增加的公有成员函数。

答案： 错误  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/类型兼容

77、设置虚基类的目的是提高程序的运行效率。

答案： 错误  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/多重继承

78、定义一个虚基类后，其后继层次中的类都需要对虚基类进行初始化。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/继承/多重继承

79、带有虚函数的类称为抽象类。

答案： 错误  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/抽象类的基本概念

80、抽象类虽然也是一种数据类型，但它不能建立对象，只能作为基类被其他类继承。

答案： 正确  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/抽象类的工作方式

81、在C++中,可以声明虚构造函数和虚析构造函数。

答案： 错误  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/虚函数的工作方式

82、抽象类不能定义对象，但可以声明抽象类的指针或引用。

答案： 正确  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/抽象类的工作方式

83、对虚函数的调用，系统都是采用动态联编。

答案： 错误  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/虚函数的工作方式

84、声明虚函数是体现多态性的基础。

答案： 正确  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-14  
知识点：CPP部分/多态性/虚函数的工作方式

85、基类的指针或引用调用虚函数时采用后期绑定。

答案： 正确  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/虚函数的工作方式

86、由抽象基类继承得到的派

生类肯定是具体类。

答案： 错误  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/虚函数的工作方式

87、析构造函数中调用虚函数采用静态联编。

答案： 正确  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/虚函数的工作方式

88、构造函数可以声明为虚函数。

答案： 错误  
评析：

等级：1 难度：1  
区分度：1  
出题人：文本导入  
修改人：admin 审核人：admin  
创建时间：2011-6-1  
最后修改时间：2011-6-10  
知识点：CPP部分/多态性/虚函数的工作方式

89、ios类是istream类、ostream类的虚基类。

答案： 正确

评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流简介

90、4个预定义流对象中支持缓冲的用于出错信息标准输出的是cerr。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流简介

91、cin>>x;等价于operator>>(cin,x);

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-14  
知识点: C++部分/输入输出流/输入输出流简介

92、已知char str[20];cin.getline(str,80);输入abc def, 则cout<<str;输出为abc def。

答案: 正确

评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流的成员函数

93、cin.read()用于输入字符串。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流的成员函数

94、getline()函数中使用的终止符只能是换行符。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/输入输出流的成员函数

95、提取运算符重载函数的参数是istream类对象的引用和自定义类对象的引用, 返回为istream类对象的引用。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/自定义数据类型的输入输出

96、重载插入运算符和提取运算符的函数需要定义为类的非成员函数或友元函数。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/输入输出流/自定义数据类型的输入输出

97、fstream类是iostream类的派生类

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件与流

98、C++中, 打开一个文件, 就是将该文件与流对象建立关联; 关闭一个文件, 就是将该关联断开。

答案: 正确  
评析:



等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的打开与关闭

99、利用只读方式打开D盘根目录下的abc.txt文件可以使用如下命令: ifstream  
fin("d:\\abc.txt");

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的打开与关闭

100、当使用ifstream流类定义一个流对象并打开一个磁盘文件时, 文件的隐含打开方式为 ios::out。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的打开与关闭

101、ios::out|ios::binary表示以输出方式打开一个二进制文件。

答案: 正确

评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的打开与关闭

102、以只读方式打开文件时, 文件指针指向文件任意位置。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的打开与关闭

103、以追加方式打开文件时, 文件指针指向文件尾。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的打开与关闭

104、只使用ios::out方式打开已有文件时, 将清空原有文件。

答案: 正确  
评析:

等级: 1 难度: 1

区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的打开与关闭

105、C++中的文本文件以ASCII形式存储数据。

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文本文件的顺序读写

106、C++中, 随机读/写方式不适用于文本文件。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/文件/文件的随机读写

107、函数模板就是模板函数

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin

创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/函数模板定义

108、类模板的类型参数只能是C++中的基本类型。

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

109、模板类的实例是一个实际的对象,可以像其他对象一样使用

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

110、函数模板不能含有常规形参

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10

知识点: C++部分/模板/函数模板定义

111、函数模板的一个实例就是一个函数定义

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/函数模板使用

112、类模板的成员函数不能是模板函数

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

113、用类模板定义对象时,可以省略类型实参

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-14

知识点: C++部分/模板/类模板定义

114、类模板的成员函数都是模板函数

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

115、类模板描述了一组类

答案: 正确  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

116、类模板中只允许有一个类型参数

答案: 错误  
评析:

等级: 1 难度: 1  
区分度: 1  
出题人: 文本导入  
修改人: admin 审核人: admin  
创建时间: 2011-6-1  
最后修改时间: 2011-6-10  
知识点: C++部分/模板/类模板定义

117、重载的运算符至少有一个参数的类型与自定义的类型有关。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/运算符重载/运算符重载方法

118、系统会为用户定义类自动提供一个默认的赋值运算符，但当类具有指针类型的数据成员时，一般需要重新定义赋值运算符。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/运算符重载/运算符重载规则

119、假定要对类AB定义加号操作符重载成员函数，实现两个AB类对象的加法，并返回相加结果，则该成员函数的声明语句为：AB operator+(AB &,AB &);

答案： 错误  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2

最后修改时间：2011-6-10  
知识点：CPP部分/运算符重载/特殊运算符重载

120、重载插入运算符"<<"必须采用成员函数重载。

答案： 错误  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-14  
知识点：CPP部分/运算符重载/特殊运算符重载

121、对二元运算符采用成员函数重载时，只需要一个参数，而且该参数可以是基本数据类型。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-14  
知识点：CPP部分/运算符重载/运算符重载规则

122、对一元运算符采用某个类的友元函数重载时需要一个参数，参数为该类的对象，不能是其他类型。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin

创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/运算符重载/运算符重载规则

123、在一个类中，可以对一个操作符进行多次重载。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/运算符重载/运算符重载的基本概念

124、C++中所有的运算符都可以被重载。

答案： 错误  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10  
知识点：CPP部分/运算符重载/运算符重载的基本概念

125、如果在某个类中使用成员函数对运算符重载，其左操作数必须是该类的对象。

答案： 正确  
评析：

等级：1                      难度：1  
区分度：1  
出题人：文本导入  
修改人：admin                      审核人：admin  
创建时间：2011-6-2  
最后修改时间：2011-6-10

知识点：CPP部分/运算符重载/  
运算符重载规则

126、重载赋值运算符"="时，  
为了保持原有特性，重载运算  
符函数中应该使用返回语句  
"return \*this;"。

答案： 正确

评析：

空

等级：1 难度：1

区分度：1

出题人：文本导入

修改人：admin 审核

人：admin

创建时间：2011-6-2

最后修改时间：2011-6-10

知识点：CPP部分/运算符重载/  
特殊运算符重载

所有简答题

1、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
void fun(int &a, int b)
{
    int t = a;
    a = b;
    b = t;
}
int main()
{
    int a = 3, b = 5;
    fun(a, b);
    cout<<a<<endl;
    cout<<b<<endl;
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/函数的引用调用  
2、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
void fun(int a, int b, int &sum, int sub)
{
    sum = a+b;
    sub = a-b;
}
int main()
{
    int a=5, b=10, sum=0, sub=0;
    fun(a, b, sum, sub);
    cout<<sum<<endl;
    cout<<sub<<endl;
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/函数的引用调用  
3、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
void fun(int *p, int n)
{
    for (int i=0; i<n; i++)
        p[i+1]=p[i-1]+p[i];
}
int main()
{
    int a[] = {1, 2, 3, 4, 5, 6};
    fun(a+1, 2);
    cout<<a[3]<<','<<a[4]<<endl;
    fun(a+2, 2);
    cout<<a[3]<<','<<a[4]<<endl;
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/指针作为函数参数

4、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
void fun(char *p, int n)
{
    char t;
    for (int i=0; i<n/2; i++)
    {
        t=p[i];
        p[i]=p[n-i-1];
        p[n-i-1]=t;
    }
}
int main()
{
    char s[] = "abcdefg";
    fun(s, strlen(s));
    cout<<s<<endl;
    fun(s+2, 2);
    cout<<s<<endl;
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/指针作为函数参数

5、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
float divide(int x, int y)
{
    return x/y;
}
double divide(double x, double y=2.0)
{
    return x/y;
}
int main()
{
    cout<<divide(5)<<endl;
    cout<<divide(5, 2)<<endl;
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/函数重载  
6、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
void fun(char *s)
{
    cout<<s<<endl;
}
void fun()
{
    cout<<"default"<<endl;
}
int main()
{
    fun();
    fun("abc");
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/函数重载

7、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
int fun(int a[], int n)
{
    int f;
    if (n==1)
        return a[0];
    f = fun(a+1, n-1);
    if (f<a[0])
        return f;
    return a[0];
}
int main()
{
    int a[] = {2, 3, 1, 22, 5, 26, 7, 9, 6, 6};
    cout<<fun(a,
    sizeof(a)/sizeof(a[0]))<<endl;
    cout<<fun(a+5, 3)<<endl;
    return 0;
}
```

等级: 1 难度: 3  
知识点: C++部分/函数/函数的递归调用  
8、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
int fun(int n)
{
    if (n<3)
        return 1;
    return 3*fun(n-1)+2*fun(n-2);
}
int main()
{
    cout<<fun(2)<<endl;
    cout<<fun(5)<<endl;
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/函数的递归调用  
9、下面程序的运行结果是\_\_\_\_①\_\_\_\_。  
\_\_\_\_②\_\_\_\_。

```
#include <iostream>
using namespace std;
int f(int n)
{
    if (n==0)
        return 1;
    return (f(n-1)+1)*2;
}
int main()
{
    cout<<f(0)<<endl;
    cout<<f(5)<<endl;
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/函数/函数的递归调用  
16、下面程序的输出结果为\_\_\_\_\_①\_\_\_\_\_  
\_\_\_\_\_②\_\_\_\_\_。

```
#include<iostream>
using namespace std;
class Sample
{
    int x,y;
public:
    Sample(){ x=y=0; }
    Sample(int a,int b) { x=a,y=b; }
    void disp()

{ cout<<"x="<<x<<" ,y="<<y<<endl; }
};
int main()
{
    Sample s1,s2(2,3);
    s1.disp();
    s2.disp();
    return 0;
}
```

等级: 1 难度: 1  
知识点: C++部分/类与对象/类和对象的声明与定义  
17、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_  
\_\_\_\_\_②\_\_\_\_\_。

```
#include<iostream>
using namespace std;
class Sample
{
private:
    int x;
    int y;
public:
    void get(int a,int b)
    {
        x=a;
        y=b;
    }
    void disp()
    {

cout<<"x="<<x<<" ,y="<<y<<endl;
    }
};
int main()
{
    Sample object1;
    object1.get(10,20);
    object1.disp();
    Sample object2(object1);
    object2.disp();
    return 0;
}
```

等级: 1 难度: 1  
知识点: C++部分/类与对象/类和对象的声明与定义  
18、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_  
\_\_\_\_\_②\_\_\_\_\_。

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

```
class point{
    int x,y;
public:
    point(int x1,int x2=0)
    {
        x=x1;
        y=x2;
    }
    void setpoint(int x1,int y1)
    {
        x=x1;
        y=y1;
    }
    void dispoint()
    {
        cout<<x<<endl<<y<<endl;
    }
};
int main()
{
    point *p=new point(12);
    p->setpoint(5,12);
    p->dispoint();
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/类与对象/构造析构函数作用和声明  
19、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_  
\_\_\_\_\_②\_\_\_\_\_。

```
#include<iostream>
using namespace std;
class Wealthy
{
public:
    char answer[10];
    void wantMoney()
    {
        char answer[10] = "Yes!";
        cout<<"Give me
$10?>"<<answer<<endl;
        cout<<"Would you like
$10?>"<<this->answer<<endl;
    }
};
int main()
{
    Wealthy w;
    strcpy(w.answer,"No!");
    w.wantMoney();
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/类与对象/this指针  
20、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_  
\_\_\_\_\_②\_\_\_\_\_。

```
#include<iostream>
using namespace std;
class A
{
private:
    int x;
public:
```

```
A(int a){x=a;}
friend class B;
};
class B
{
public:
    void print1(A a)
    {
        a.x--;cout<<"x="<<a.x<<endl;
    }
    void print2(A a)
    {
        a.x++;cout<<"x="<<a.x<<endl;
    }
};
int main()
{
    A a(10);
    B b;
    b.print1(a);
    b.print2(a);
    return 0;
}
```

等级: 1 难度: 1  
知识点: C++部分/类与对象/构造函数重载  
21、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_  
\_\_\_\_\_②\_\_\_\_\_。

```
#include<iostream>
using namespace std;
class X
{
private:
    int a;
    int b;
public:
    X()
    {
        a=0;
        b=0;
    }
    X(int a,int b)
    {
        X::a=a;
        X::b=b;
    }
    void disp()
    {
        cout<<"a="<<a<<" ,b="<<b;
    }
};
class Y
{
private:
    X x;
    int y;
public:
    Y()
    {
        y=0;
    }
    Y(int a,int b,int c):x(a,b)
    {
        y=c;
    }
}
```

```

void disp()
{
    x.disp();
    cout<<"y="<<y<<endl;
}
};
int main()
{
    Y object1;
    Y object2(1,2,3);
    object1.disp();
    object2.disp();
    return 0;
}

```

等级: 1 难度: 1

知识点: C++部分/类与对象/构造函数重载

22、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```

#include<iostream>
using namespace std;
class Member
{
public:
    Member(char *name,char *address)
    {
        strcpy(Member::name,name);

        strcpy(Member::address,address);
        friend void Show(Member);
        friend void SetName(Member,char
*);
private:
    char name[20];
    char address[20];
};
void Show(Member obj)
{
    cout<<obj.name<<endl;
}
void SetName(Member m,char *name)
{
    strcpy(m.name,name);
}
int main()
{
    Member a("张强","南开大学宿");
    Show(a);
    SetName(a,"刘晓征");
    Show(a);
    return 0;
}

```

等级: 1 难度: 3

知识点: C++部分/类与对象/友元

23、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```

#include<iostream>
using namespace std;
class SavingAccount
{
private:
    char m_name[40];

```

```

    char m_addr[60];
    double total;
public:
    static double m_rate;
    static void setRate(double newRate)
    {
        m_rate = newRate;
    }
};
double SavingAccount::m_rate = 0.0075;
int main()
{
    SavingAccount::setRate(0.0074);
    SavingAccount
a1,a2;cout<<a1.m_rate<<endl;
a2.setRate(0.0072);
cout<<a1.m_rate<<endl;
return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/类与对象/类的静态成员

24、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```

#include<iostream>
using namespace std;
class Sample
{
    int A;
    static int B;
public:
    Sample(int a)
    {
        A=a,B+=a;
    }
    static void func(Sample s);
};
void Sample::func(Sample s)
{
    cout<<s.A<<" "<<s.B<<endl;
}
int Sample::B=0;
int main()
{
    Sample s1(2),s2(5);
    Sample::func(s1);
    Sample::func(s2);
    return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/类与对象/类的静态成员

25、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```

#include<iostream>
using namespace std;
class parent
{
    int i;
protected:
    int x;
public:
    parent() {x=0; i=0;}

```

```

    void change() {x++; i++;}
    void display();
};
class son: public parent
{
public:
    void modify();
};
void parent::display()
{cout<<"x="<<x<<endl; }
void son::modify() {x++; }
int main()
{
    son A;
    parent B;
    A.change();
    A.modify();
    A.display();
    B.change();
    B.display();
    return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/继承/继承的基本概念

26、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```

#include<iostream>
using namespace std;
class base
{
public:
    void show() {cout<<"class base
show() is called\n"; }
};
class derived: public base
{
public:
    void show() {cout<<"class
derived show() is called\n"; }
};
int main()
{
    base demo1;
    derived demo2;
    demo1.show();
    demo2.show();
    return 0;
}

```

等级: 1 难度: 1

知识点: C++部分/继承/继承的基本概念

27、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```

#include<iostream>
using namespace std;
class A
{
public:
    A(int i, int j)
    {
        a = i;
        b = j;
    }
    void move(int x, int y)

```

```

{
    a+=x;
    b+=y;
}
void show()
{
    cout<<a<<" "<<b<<endl;
}
private:
    int a,b;
};class B:private A
{
public:
    B(int i, int j):A(i,j)
    {}
    void fun()
    {
        move(3,5);
    }
    void fl()
    {
        show();
    }
};int main()
{
    B d(3,4);
    d.fl();
    d.fun();
    d.fl();
    return 0;
}

```

等级：1 难度：1  
 知识点：CPP部分/继承/继承的基本概念  
 28、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_

\_\_\_\_\_②\_\_\_\_\_。  
 \_\_\_\_\_③\_\_\_\_\_。

```

#include<iostream>
using namespace std;
class Table
{
public:
    Table(){cout<<"table"<<endl;}
};
class Circle
{
public:
    Circle(){cout<<"circle"<<endl;}
};
class RoundTable:public Circle,public Table
{
public:
    RoundTable():Table(),Circle()
    { cout<<"roundtable"<<endl; }
};
int main()
{
    RoundTable rt;
    return 0;
}

```

等级：1 难度：1  
 知识点：CPP部分/继承/多重继承  
 29、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_

```

#include<iostream>
using namespace std;
class A
{
public:
    A()
    {a=0;}
    A(int i)
    {a = i;}
    void print()
    {
        cout<<a<<';
    }
private:
    int a;
};class B: public A
{
public:
    B()
    {b1=b2=0;}
    B(int i)
    {b1=0;b2=i;}
    B(int i,int j,int k):A(i),b1(j),b2(k)
    {}
    void print()
    {
        A::print();
        cout<<b1<<','<<b2<<endl;
    }
private:
    int b1,b2;
};int main()
{
    B b1;
    B b2(15);
    B b3(11,12,13);
    b1.print();
    b2.print();
    b3.print();
    return 0;
}

```

等级：1 难度：3  
 知识点：CPP部分/继承/派生类的构造函数

30、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_。  
 \_\_\_\_\_②\_\_\_\_\_。

```

#include<iostream>
using namespace std;
class A
{
public:
    A()
    {cout<<"A";}
    ~A()
    {cout<<"~A";}
    int n;
};class B: public A
{
public:
    B()
    {
        cout<<"B";
    }
}

```

\_\_\_\_\_②\_\_\_\_\_。  
 \_\_\_\_\_③\_\_\_\_\_。

```

~B()
{
    cout<<"~B";
}
};class C: public B
{
public:
    A a;
    C()
    {
        cout<<"C\n";
    }
    ~C()
    {
        cout<<"~C";
    }
};int main()
{
    C c;
    return 0;
}

```

等级：1 难度：2  
 知识点：CPP部分/继承/派生类的构造函数

35、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_。  
 \_\_\_\_\_②\_\_\_\_\_。

```

#include<iostream>
using namespace std;
int main()
{
    char *str="南开大学";
    cout.write(str,8)<<endl;
    char c='c';
    cout.put(c).put('c')<<endl;
    return 0;
}

```

等级：1 难度：1  
 知识点：CPP部分/输入输出流/输入输出流的成员函数

36、从键盘输入abcdefg，运行结果是\_\_\_\_\_①\_\_\_\_\_。

```

#include<iostream>
using namespace std;
int main()
{
    char s[20]="";
    cin.read(s,6);
    cout<<s<<endl;
    cout<<cin.gcount()<<endl;
    return 0;
}

```

等级：1 难度：1  
 知识点：CPP部分/输入输出流/输入输出流的成员函数

37、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_。  
 \_\_\_\_\_②\_\_\_\_\_。

```

#include<iostream>
using namespace std;

```



```
#include<fstream>
int main()
{
    fstream f1("a1.txt",ios::out);
    fstream f2("a2.txt",ios::out);
    char
s1[80]="abcd",s2[80]="defg",ch,s[80];
    f1<<s1<<endl;
    f2<<s2<<endl;
    f1.close();
    f2.close();
    f1.open("a1.txt",ios::app);
    f2.open("a2.txt",ios::in);
    while(f2>>ch)
        f1<<ch;
    f1.close();
    f2.close();
    f1.open("a1.txt",ios::in);
    while(!f1.eof())
    {
        f1.getline(s,sizeof(s));
        cout<<s<<endl;
    }
    f1.close();
    return 0;
}
```

等级: 1 难度: 1  
知识点: C++部分/文件/文本文件的顺序读写

38、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```
#include<iostream>
using namespace std;
#include<fstream>
class Point
{
    int x,y;
public:
    Point(int i=0,int j=0)
    {
        x=i;y=j;
    }
    void Print()
    {
        cout<<"x="<<x<<"y="<<y<<endl;
    }
};
int main()
{
    Point p1(1,2),p2(3,4),p3;
    ofstream file1("abc.dat");
    file1.write((char*)&p1,sizeof(p1));
    file1.write((char*)&p2,sizeof(p2));
    file1.close();
    ifstream file2("abc.dat");
    file2.read((char*)&p3,sizeof(p3));
    p3.Print();
    file2.read((char*)&p3,sizeof(p3));
    p3.Print();
    file2.close();
    return 0;
}
```

等级: 1 难度: 1  
知识点: C++部分/文件/二进制文件的顺

序读写

39、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```
#include<iostream>
using namespace std;
#include<fstream>
int main()
{
    int f[20]={0,1,1},x,sum=0;
    for(int i=3;i<20;i++)
        f[i]=f[i-1]+f[i-2];
    fstream
fio("abc.dat",ios::in|ios::out|ios::binary|ios::
trunc);
    fio.write((char*)f,sizeof(f));
    for(int i=1;i<=3;i++)
    {
        fio.seekg((i)*sizeof(int),ios::beg);
        fio.read((char*)&x,sizeof(int));
        sum=sum+x;
    }
    cout<<x<<endl;
    cout<<sum<<endl;
    fio.close();
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/文件/文件的随机读写

40、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```
#include<iostream>
using namespace std;
#include<fstream>
int main()
{
    ofstream outfile("abc.txt");
    char str[]="abcdefg",ch;
    outfile<<str;
    outfile.close();
    ifstream infile("abc.txt");
    infile>>ch;
    cout<<infile.tellg()<<ch<<endl;
    infile.seekg(0,ios::end);
    infile.seekg(-1,ios::cur);
    infile>>ch;
    cout<<infile.tellg()<<ch<<endl;
    infile.close();
    return 0;
}
```

等级: 1 难度: 2  
知识点: C++部分/文件/文件的随机读写

44、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```
#include<iostream>
using namespace std;
class Increase
{
public:
    Increase(int x):value(x){}
    Increase & operator++();
    Increase operator++(int);
    void display(){ cout<<"the value is "

```

```
<<value <<endl; }
private:
    int value;
};
Increase & Increase::operator++()
{
    value++;
    return *this;
}
Increase Increase::operator++(int)
{
    Increase temp(*this);
    value++;
    return temp;
}
int main()
{
    Increase n(20);
    ++(++n);
    n.display();
    (n++)++;
    n.display();
}
```

等级: 1 难度: 3  
知识点: C++部分/运算符重载/特殊运算符重载

45、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```
#include<iostream>
using namespace std;
class myCla
{
    int x,y;
public:
    myCla(int x0,int y0){x=x0;y=y0;}
    friend int operator+(myCla&
myCla&);
    int operator-(myCla&);
};
int operator+(myCla& ob1,myCla& ob2)
{
    return ob1.x+ob2.x;
}
int myCla::operator-(myCla& ob2)
{
    return y-ob2.y;
}
int main()
{
    myCla a(1,2),b(6,8);
    cout<<a+b<<operator+(a,b)<<endl;
    cout<<b-a<<b.operator-(a)<<endl;
    return 0;
}
```

等级: 1 难度: 3  
知识点: C++部分/运算符重载/运算符重载方法

46、下面程序的运行结果是\_\_\_\_①\_\_\_\_  
\_\_\_\_②\_\_\_\_。

```
#include<iostream>
using namespace std;
class Fun
{
```

```
public:
    int operator()(int x,int y)
    { return (x*x+5*x-8)*(y+5); }
};
int main()
{
    Fun f1,f2;
    cout<<f1(3,2)<<endl;
    cout<<f2(2,3);
    return 0;
}
```

等级: 1 难度: 2

知识点: C++部分/运算符重载/特殊运算符重载

47、下面程序的运行结果是\_\_\_\_\_①\_\_\_\_\_

\_\_\_\_\_②\_\_\_\_\_

\_\_\_\_\_③\_\_\_\_\_。

```
#include<iostream>
using namespace std;
class CArray
{
public:
    CArray(int i)
    {
        Length=i;
        Buffer=new char[Length+1];
    }
    ~CArray()
    { delete []Buffer; }
    int GetLength()
    { return Length; }
    char& operator[](int i);
private:
    int Length;
    char* Buffer;
};
char& CArray::operator[](int i)
{
    static char ch;
    if(i<Length&&i>=0)
        return Buffer[i];
    else
    {
        cout<<"Index out of range.";
        return ch;
    }
}
int main()
{
    int cnt;
    CArray string1(6);
    char *string2="Nankai";
    for(cnt=0;cnt<7;cnt++)
        string1[cnt]=string2[cnt];
    cout<<endl;
    for(cnt=0;cnt<6;cnt++)
        cout<<string1[cnt];
    cout<<"\n";
    cout<<string1.GetLength()<<endl;
    return 0;
}
```

等级: 1 难度: 3

知识点: C++部分/运算符重载/特殊运算符重载

符重载  
所有问答題  
17、请将下面程序补充完整。

```
#include<iostream>
using namespace std;
class Person
{
private:
    int no;
    char name[20];
public:
    _____①_____ (int no,char name[])
    {
        this->no=no;
        strcpy(this->name,name);
    }
    int getno() {return no;}
    char *getname() {return name;}
};
int main()
{
    Person *p;
    p=_____②_____;
    p->setvalue(1,"zhang");
```

```
cout<<p->getno()<<": "<<(*p).getname()<<endl;
return 0;
}
```

等级: 1 难度: 2

知识点: C++部分/类与对象/类与对象的声明与定义

18、请补充完整下面类说明中的拷贝构造函数。

```
class MyClass
{
public:
    int *p;
    int size;
    MyClass(MyClass& mc)
    {
        size=_____①_____;
        p=new int[size];
        for(int i=0;i<size;i++)
            _____②_____;
    }
};
```

等级: 1 难度: 2

知识点: C++部分/类与对象/构造函数作用与声明

19、ARRAY类的定义如下，构造函数把参数n的值赋给s，给v动态分配长度为n的数组空间，然后利用数组参数a初始化v所指向的数组。请将类定义补充完整。

```
class ARRAY
{
    int *v;
    int s;
public:
    ARRAY( int a[], int n );
    ~ARRAY() { delete []v; }
    int size() { return s; }
```

```
int& operator[](int n);
};
_____①_____ ARRAY(int a[], int n)
{
    if( n<=0 )
    {
        v=NULL;
        s=0;
        return;
    }
    s=n;
    v=_____②_____;
    for(int i=0; i<n; i++)
        v[i]=a[i];
}
```

等级: 1 难度: 2

知识点: C++部分/类与对象/类与对象的声明与定义

20、请将下面程序补充完整。

```
#include<iostream>
using namespace std;
class Student
{
private:
    double score;
    _____①_____;
};
class Teacher
{
public:
    void setScore(Student &stu,double s)
    {
        stu.score=s;
    }
    //函数isPass判断stu是否及格
    //score在60以上(包括60)值为1, 否则值为0
    int isPass(Student &stu)
    {
        _____②_____;
    }
};
```

等级: 1 难度: 2

知识点: C++部分/类与对象/友元

21、请将下面程序补充完整。

```
#include<iostream>
using namespace std;
class Sample
{
private:
    int num;
public:
    Sample(int num)
    {
        _____①_____; //将形参num赋值给成员变量num
    }
    _____②_____ { num=0; }
};
int main()
{
```

```

Sample s;
return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/类与对象/构造函数重载

22、请将下面程序补充完整。

```

#include<iostream>
using namespace std;
class Sample
{
private:
    int num;
public:
    ①
    {
        num=n;
    }
    ②
    {
        num=0;
    }
};
int main()
{
    Sample s1(5),s2;
    return 0;
}

```

```

    ①
    {
        num=n;
    }
    ②
    {
        num=0;
    }
};
int main()
{
    Sample s1(5),s2;
    return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/类与对象/构造函数重载

23、请将下面程序补充完整。

```

#include<iostream>
using namespace std;
class Sample
{
    ①
    static int num;
};
②; //将静态成员num初始化为0
int main()
{
    Sample::num=1;
    return 0;
}

```

等级: 1 难度: 1

知识点: C++部分/类与对象/类的静态成员

24、请将下面程序补充完整。

```

#include<iostream>
using namespace std;
class Student
{
private:
    double score;
public:
    void setScore(double score)
    {
        ①; //将形参score赋值给成员score
    }
}

```

```

}
//compareScore用来比较成员score与形参score的大小
//成员score大于或等于形参score, 返回1, 否则返回0
int compareScore(double score)
{
    ②;
}
};
int main()
{
    Student s1;
    s1.setScore(50);
    cout<<s1.compareScore(60)<<endl;
    return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/类与对象/this指针

25、请将下面程序补充完整。

```

#include<iostream>
using namespace std;
class Student
{
private:
    double score;
public:
    void setScore(double s)
    {
        score=s;
    }
    //compareScore用来比较两个Student对象中score的大小
    //返回score较大的那个对象 (假设两个score不会相等)
    Student& compareScore(Student &s)
    {
        if(①)
            ②;
        else
            return s;
    }
};
int main()
{
    Student s1,s2,s3;
    s1.setScore(50);
    s2.setScore(60);
    s3 = s1.compareScore(s2);
    return 0;
}

```

等级: 1 难度: 1

知识点: C++部分/类与对象/this指针

31、请将下面的程序补充完整, 使得程序输出结果为:

```

x=2
x square=4
x=3
x cube=27#include <iostream>
#include <cmath>
using namespace std;

```

```

class Power
{
public:
    Power(int i){x=i;}
    ①
protected:
    int x;
};
class Square:public Power
{
public:
    Square(int i):Power(i) {}
    void display()
    {
        cout<<"x="<<x<<endl;
        cout<<"x square="<<x*x<<endl;
    }
};
class Cube:public Power
{
public:
    Cube(int i) : Power(i){}
    void display()
    {
        cout<<"x="<<x<<endl;
        cout<<"x cube="<<x*x*x<<endl;
    }
};
void fun(②)
{
    p.display();
}
int main()
{
    Square squ(2);
    Cube cub(3);
    fun(squ);
    fun(cub);
    return 0;
}

```

等级: 1 难度: 3

知识点: C++部分/多态性/虚函数的工作方式

32、请将下面的程序补充完整。

```

#include<iostream>
using namespace std;
class Instrument
{
public:
    virtual void play(){cout<<"play instrument"<<endl;}
    void display(){cout<<"instrument"<<endl;}
};
class Wind: ①
{
public:
    void play(){cout<<"play wind"<<endl;}
    void display(){cout<<"wind"<<endl;}
};
② :public Instrument
{
public:

```

```

void play(){cout<<"play
stringed"<<endl;}
void
display(){cout<<"stringed"<<endl;}
};
int main()
{
    Wind w;
    Stringed s;
    Instrument *p=&w;
    p->play();
    w.play();
    p=&s;
    p->display();
    s.display();
    w.display();
    return 0;
}

```

等级: 1 难度: 3

知识点: C++部分/多态性/虚函数的工作方式

33、请将下面的程序补充完整。

```

#include<iostream>
using namespace std;
class A
{
public:
    A()
    { cout<<"A::A()called.\n"; }
    virtual ~A()
    { cout<<"A::~A()called.\n"; }
};
class B :public A
{
public:
    B(int i)
    {
        cout<<"B::B()called.\n";
        buffer=new char[i];
    }
    virtual ~B()
    {
        ①;
        cout<<"B::~B()called.\n";
    }
private:
    char* buffer;
};
void fun(A* a)
{ delete a; }
int main()
{
    A *b=new B(10);
    fun(②); // 将b所指向的
    对象销毁
    return 0;
}

```

等级: 1 难度: 3

知识点: C++部分/多态性/虚函数的工作方式

34、请将下面的程序补充完整,使得程序输出结果为:

Nankai University

```

Nankai University#include<iostream>
using namespace std;
class Nan
{
public:
    ① void speak()
    { cout<<"Nankai"<<" "; }
};
class Uni:public Nan
{
public:
    void speak()
    { cout<<"University"<<endl; }
};
int main()
{
    Nan a, ②;
    Uni p;
    a.speak();
    p.speak();
    pa=&a;
    pa->speak();
    pa=&p;
    pa->speak();
    return 0;
}

```

等级: 1 难度: 3

知识点: C++部分/多态性/虚函数的工作方式

35、下面程序从键盘输入一行字符,使用cin.get()从中逐个提取字符并赋值给变量ch,然后将ch输出到屏幕,请将程序填写完整。

```

#include<iostream>
using namespace std;
int main()
{
    char ch;
    while((ch= ①)!='\n')
    {
        ②;
    }
    return 0;
}

```

等级: 1 难度: 1

知识点: C++部分/输入输出流/输入输出流的成员函数

36、下面程序重载插入运算符,实现对坐标点对象的输出,请将程序填写完整。

```

#include<iostream>
using namespace std;
class Point
{
    double x;
    double y;
public:
    Point(double a=0,double b=0)
    {
        x=a;
        y=b;
    }
    friend ①(ostream&,Point &);
};

```

```

};
①(ostream& output,Point &p)
{
    output<<p.x<<" "<<p.y<<endl;
    ②;
}
int main()
{
    Point p1,p2(3,5);
    cout<<p1<<p2;
    return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/输入输出流/自定义数据类型输入输出

37、下面程序重载提取运算符,实现对坐标点对象的输入,请将程序填写完整。

```

#include<iostream>
using namespace std;
class Point
{
    double x;
    double y;
public:
    friend ①(istream&,Point &);
    void display()
    {
        cout<<x<<" "<<y<<endl;
    }
};
①(istream& input,Point &p)
{
    input>>p.x>>p.y;
    ②;
}
int main()
{
    Point p1,p2;
    cin>>p1>>p2;
    p1.display();
    p2.display();
    return 0;
}

```

等级: 1 难度: 2

知识点: C++部分/输入输出流/自定义数据类型输入输出

45、请将下面程序补充完整。

```

#include<iostream>
using namespace std;
class Point
{
    double x,y;
public:
    Point(double a,double b)
    {
        x=a;
        y=b;
    }
    ① int
    operator==(Point&,Point&);
    Point& operator+=(Point&);
};

```

```

void display()
{ cout<<x<<" "<<y<<endl; }
};
Point& __②__ (Point& c)
{
    x=x+c.x;
    y=y+c.y;
    return *this;
}
int operator==(Point& a,Point& b)
{
    if(a.x==b.x&& a.y==b.y)
        return 1;
    else
        return 0;
}
int main()
{
    Point p1(3,4),p2(4,5);
    if(p1==p2)
        cout<<"相等"<<endl;
    else
        cout<<"不相等"<<endl;
    p1+=p2;
    p1.display();
    p2.display();
    return 0;
}

```

等级: 1 难度: 3

知识点: C++部分/运算符重载/特殊运算符重载

46、请将下面程序补充完整,使得程序输出结果为:

```

n=10
n=30#include<iostream>
using namespace std;
class Sample
{
public:
    Sample() {}
    Sample(int i){n=i;}
    friend Sample operator-(Sample
    &,Sample &);
    friend Sample operator+(Sample
    &,Sample &);
    void disp(){cout<<"n="<<n<<endl;}
private:
    int n;
};
Sample operator-(__①__)
{
    int m=s1.n-s2.n;
    Sample tmp(m);
    return tmp;
}
__②__
{
    int m=s1.n+s2.n;
    Sample tmp(m);
    return tmp;
}
int main()
{
    Sample s1(10),s2(20),s3;

```

```

s3=s2-s1;
s3.disp();
s3=s2+s1;
s3.disp();
return 0;
}

```

等级: 1 难度: 3

知识点: C++部分/运算符重载/运算符重载方法

47、请将下面程序补充完整。

```

#include<iostream>
using namespace std;
class complex
{
    float REAL, IMAGE;
public:
    complex()
    { REAL=0.0; IMAGE=0.0; }
    complex(float real, float image)
    { REAL=real; IMAGE=image; }
    float getR()
    { return REAL; }
    float getI()
    { return IMAGE; }
    __①__
    complex operator-(complex& arg);
};
complex complex::operator+(complex&
arg)
{
    complex s(REAL+arg.REAL,
    IMAGE+arg.IMAGE);
    return s;
}
complex complex::operator-(complex&
arg)
{
    complex m(REAL-arg.REAL,
    IMAGE-arg.IMAGE);
    return m;
}
int main()
{
    float real, image;
    cout<<"n input complex A(real,
    image):";
    cin>>real>>image;
    complex A(real, image);
    cout<<"n input complex B(real,
    image):";
    cin>>real>>image;
    complex B(__②__); // 用real和
    image的值初始化B中的成员变量
    complex C;
    C=A+B;
    cout<<"complex
    A+B=("<<C.getR()<<")+"
    i("<<C.getI()<<")n";
    C=A-B;
    cout<<"complex
    A-B=("<<C.getR()<<")-"
    i("<<C.getI()<<")n";
    return 0;
}

```

等级: 1

难度: 3

知识点: C++部分/运算符重载/运算符重载方法

48、请将下面程序补充完整。

```

#include<iostream>
using namespace std;
class A
{
public:
    A()
    { X=Y=0; }
    A(int a, int b)
    { X=a; Y=b; }
    A& operator=(A& p);
    int GetX()
    { return X; }
    int GetY()
    { return Y; }
private:
    int X, Y;
};
__①__
{
    X=p.X;
    Y=p.Y;
    cout<<"Assignment operator
    Called.\n";
    return *this;
}
int main()
{
    A a1(72, 81);
    A a2;
    __②__; // 用a1对象给a2对象赋
    值,调用a2对象的赋值运算符重载

    cout<<a2.GetX()<<","<<a2.GetY()<<endl
    ;
    return 0;
}

```

等级: 1

难度: 3

知识点: C++部分/运算符重载/特殊运算符重载

1、 5 5

2、 15 0

3、 5,5 5,8

4、 gfedcba gfdecba

5、 2.5 2

6、 default abc

7、 1 7

8、 1 61

9、 1 94

16、 x=0,y=0 x=2,y=3

17、 x=10,y=10 x=10,y=20

18、 5 12

19、 。 。 。 yes 。 。 。 。 no

20、 x=9 x=11

21、 a=0,b=0,y=0 a=1,b=2,c=3

22、 张 张

23、 0.0074 0.0072

24、 2,7 5,7

25、 x=2 x=1

26、

27、 3,4 6,9

28、 cir tab rou

29、 0,0,0 0,0,15 11,12,13

30、 ABAC ~C~A~B~A

35、 南开大学 cc

36、 abcdef 6

37、 abcd defg

38、 x=1y=2 x=3y=4 ! ! !

39、 2 4

40、 1a7g

44、 the 22 theval。 。 23

45、 77 66

46、 112 48

47、 inde。 。 。 Nankai。 。 6