练习题2

1. 单选题

1、下列语法特性中，C和C++都支持的是：B

a) 模板 b) 宏

c) 命名空间 d) 函数缺省参数

2、以下关于C++项目生成和管理说法错误的是：D

a) 一个C++项目可以有多个.cpp文件

b) C++项目的生成需要经过预编译、编译、连接三个步骤

c) 一般把声明放在.h中，而定义放在.cpp中

d) #include指令用于导入外部函数和变量，因此是由连接器负责处理

3、以下函数声明错误的是：C

a) void func(int p[], int size);

b) void func(int p[5]);

c) void func(int a=0, float b);

d) void func(int a=0, float b=0);

4、下列同名函数中，哪一个不能与void func(int \*); 一起重载：C

a) void func(const int \*);

b) int func(int \*, int);

c) int func(int\*);

d) void func();

5、以下对class A的构造函数声明错误的是：C

a) A();

b) A(int size, int val=0);

c) A(const A val);

d) A(const A &val);

6、以下对class B的析构函数声明错误的是：D

a) ~B();

b) ~B() throw();

c) virtual ~B();

d) virtual ~B(int v);

7、通常,拷贝构造函数的参数是: D

a) 某个对象名

b) 某个对象的成员名

c) 某个对象的指针名

d) 某个对象的引用名

8、以下关于C++动态内存分配的说法正确的是：C

a) 使用动态内存分配的优点之一是速度快

b) 使用new动态分配的内存会被系统自动回收，因此不需要手动释放

c) 使用new创建对象时，也会调用类的构造函数

d) 使用delete的目的是删除动态分配的内存，因此可以用free函数代替

9、以下关于class A及其派生类class D的代码，会导致编译错误的是：A

a) const A a; A &b=a;

b) A a; const A &b=a;

c) const D d; A a=d;

d) A a; D \*d=dynamic\_cast<D\*>(&a);

10、如果class D是class B的派生类，则：B

a) D可以访问B的私有成员；

b) D可以访问B的保护成员；

c) 如果D是私有继承，则不能访问B的公有成员；

d) 如果D是公有继承，则可以访问B的所有成员；

11、以下关于异常的说法错误的是：C

a) 当异常发生时，已经创建的自动变量会被析构

b) c++中用throw语句抛出异常，用catch语句捕获异常

c) 使用异常的主要作用是辅助程序调试

d) 如果异常没有被catch到，程序会非正常退出

12、以下对于class A的运算符重载声明中，一定会导致编译错误的是：B

a) A& operator++(int);

b) int operator[](int i, int j);

c) friend void operator<<(A&a, int val);

d) A operator-();

13、对class A，如果 A a; 这条语句导致了编译错误，则可能：

a) A是一个抽象类；

b) A的默认构造函数是私有成员；

c) A没有默认构造函数；

d) 以上情况都有可能；

14、Matrix是一个矩阵类，可以被用于存贮任意大小的矩阵。如果要对其重载operator==，则考虑易用性和效率等，最佳的函数接口应该设计为：A

a) friend bool operator==(const Matrix &a, const Matrix &b);

b) friend bool operator==(const Matrix a, const Matrix b);

c) friend bool operator==(Matrix &a, Matrix &b);

d) friend bool operator==(Matrix a, Matrix b);

15、如果int v=A::B()；是一条合法语句，则A, B不可能是：D

a) A是一个命名空间，B是A中的函数；

b) A是一个命名空间，B是A中的一个类；

c) A是一个类，B是A的静态成员函数；

d) A是一个类，B是A的一个普通成员函数；

16、以下说法错误的是：A

a) 全局变量可以减少函数参数传递，简化程序设计，应该尽量加以利用；

b) 一个类的静态成员变量会被其所有对象所共享；

c) 静态成员函数不可以访问普通成员变量；

d) 普通成员函数也可以访问静态成员变量；

17、对class A; A \*p; 使用指针p调用A的成员函数size()，一定错误的写法是：D

a) p->size()；

b) (\*p).size()；

c) p[0].size()；

d) p.size()；

18、对class A进行输入流操作符重载的正确方式是：B

a) friend istream operator>>(istream , const A&);

b) friend istream& operator>>(istream&, A&);

c) friend istream& operator>>(istream&, const A&);

d) friend istream operator>>(istream , A);

19、对float f=3.1; int v; 如果要将f的值赋与v，使得v的值等于3，则下列哪条语句得到的结果不正确：

a) v=f;

b) v=static\_cast<int>(f);

c) v=reinterpret\_cast<int&>(f);

d) v=int(f)；

20、如果class D是class B的派生类，则无论D与B如何实现，下列代码中一定会导致编译错误的是：

a) B b ; D d(b);

b) D d; B b(d);

c) B b; D &d(b);

d) D d; B &b(d);

1. 阅读程序，写出运行结果。

1、

#include <iostream>

using namespace std;

class Object

{

int \_id;

public:

Object(int id) :\_id(id) {

if(\_id<0) throw "invalid argument ";

cout<<”C”<<\_id<<endl;}

~Object() { cout<< "D"<<\_id<<endl; }

};

int main()

{

try{

Object a(1), b(-1), c(2);

}

catch( const char \*err) {

cout<<err<<endl;

}

return 0;

}

2、

#include <iostream>

using namespace std;

class Object

{

int \_id;

public:

Object(int id=0):\_id(id) { cout<< "C "<<\_id<<endl;}

Object(const Object &r)

:\_id(r.\_id)

{ cout<< "M "<<\_id<<endl; }

~Object() {cout<< "D "<<\_id<<endl;}

};

Object\* func(Object &a, Object b)

{

static Object s=a;

return &s;

}

int main()

{

Object a(1), b;

Object &c(\*func(a, b));

return 0;

}

3、

#include <iostream>

using namespace std;

struct A

{

A(int id) { cout<< "A "<<id<<endl;}

};

struct B

{

B(int id) { cout<< "B "<<id<<endl;}

};

struct D

:public B, virtual public A

{

B b;

A a;

D()

:a(1), b(1), B(2), A(2) {}

};

int main() {

D d;

return 0;

}

4、

#include <iostream>

using namespace std;

class B

{

public:

B() {}

B(int i) {b=i;}

virtual void virfun()

{cout<<"B::virfun() called.\n";}

private:

int b;

};

class D:public B

{

public:

D(){}

D(int i,int j):B(i){d=j;}

private:

int d;

void virfun()

{cout<<"D::virfun() called.\n";}

};

void fun(B \*obj)

{

obj->virfun();

}

int main()

{

B \*pb=new B;

fun(pb);

D \*pd=new D;

fun(pd);

pb=pd;

fun(pb);

pd=dynamic\_cast< D\*> (pb);

fun(pd);

}

**5.**

#include <iostream>

using namespace std;

class Count

{

int \*\_count;

public:

Count():\_count(new int(1)) {}

Count(const Count &r):\_count(r.\_count){ ++\*\_count; }

~Count(){

if(--\*\_count==0) delete \_count;

}

operator int() const {

return \*\_count;

}

};

Int main()

{

Count a, b, c(a), &d(b);

{

Count \*p=new Count(c);

}

cout<<a<<endl<<b<<endl<<c<<endl<<d<<endl;

return 0;

}

**三、编写程序**

1. 设计一个汽车类Vehicle，包含数据成员车轮数和车重，由它派生出类Car和类Truck，前者包含载客数，后者包含载人、载重量。类的主要的成员函数中包括show()显示所含属性。编写程序实现，并测试Car和Truck的对象建立和属性显示。

2. 设计一个Three\_d类，含有三个数据成员, 用成员函数重载与类Three\_d有关的增量运算符”++”与减量运算符”--”，建立它的前缀形式和后缀形式。

。

3. 仔细阅读下面的程序，填写下划线处的空缺代码，实现完整的功能：

#include<iostream>

#include<cstring>

using namespace std;

class Error{};

class String

{

char \*m\_ptr;

public:

String(char \*ptr){

m\_ptr = new char[];

strcpy(m\_ptr,ptr);

}

~String(){

delete m\_ptr;

}

String &operator+=(const String &str){

char \*s = new char[];

if(m\_ptr){

strcpy(s,m\_ptr);

delete[] m\_ptr;

}

strcat(s,str.m\_ptr);// appends str.m\_ptr to s

m\_ptr = s;

return ;

}

bool operator==(const String &str) const

{

return (strcmp(m\_ptr,str.m\_ptr)==0);

}

char operator[] (int i) {

if(i < strlen(m\_ptr))

return m\_ptr[i];

throw Error();

}

};

operator<<(ostream &out,const String &str)

{

return out<<str.m\_ptr;

}

int main()

{

String s1("Hello "),s2("world!");

if(s1==s2)

cout<<"S1==S2\n";

s1+=s2;

cout<<s1<<endl;

int k=0;

{

while(s1[k])

cout<<s1[k++];

}

{

cout<<"\nIndex is out of range!"<<endl;

}

return 0;

}

4.

设计Shape是一个代表所有图形的抽象类，成员length()是统计图形边长的虚函数。一般的 total()函数用来统计所有不同图形的边长总和。请从Shape派生出Triangle、Rectangle类，并扩展它们自己的属性和具体的length()实现。并编写main()测试输出各个图形各自的边长和总边长。