1:

缺省析构函数的函数体是（C）

Ａ不存在

Ｂ随机产生的

Ｃ空的

Ｄ无法确定的

用于释放内存等，若不做系统默认生成析构函数。没有返回值，也没有参数，默认为空函数。

2:

当+运算符被重载为友元函数时，友元运算符obj1+obj2被C++编译器解释为（A）.

A．operator+(obj1,obj2)

B. +(obj1,obj2)

C. obj2.operator+(obj1)

D. obj1.operator(obj2)

3:

下列选项中不属于C++的类型的是（D）。

A.signed short int

B unsigned long int

C unsigned int

D long short

4:

已知print（）函数是一个类的常量成员函数，它无返回值，下列表示中（A）是正确的。

A void print() const

B const void print()

C void const print()

D void print(const)

5:

基类和派生类可以分别叫做（B）

A.“大类”和“小类”

B.“父类”和“子类”

C.“小类”和“大类”

D.“子类”和“父类”

6:

设a=1, b=2, c=3, d=4，则条件表达式a<b ? a : c<d ? c : d的值为（A）。

A 1

B 2

C 3

D 4

7:

在类定义的外部，可以被访问的成员有（C）

A)所有类成员

B)private的类成员

C)public的类成员

D)public或private的类成员

8:

在公有继承的情况下，基类的成员（私有的除外）在派生类中的访问权限（B）

A 受限制

B 保持不变

C 受保护

D 不受保护

9:

下列叙述中，正确的是(D)。

A 派生类可以使用private派生

B 对基类成员的访问必须是无二义性的

C 基类成员的访问能力在派生类中维持不变

D 派生类继承基类的所有成员

10:

假设i=2，执行下列语句后i的值为（B）

switch(i)

{

case 1: i++;

case 2: i--;

case 3: ++i; break;

case 4: --i;

default: i++;

}

A 1

B 2

C 3

D 4

11:

（B）是不可以作为该类的成员的。

A．自身类对象的指针；

B. 自身类的对象；

C. 自身类对象的引用；

D. 另一个类的对象。

12:

C++中程序不总是从主函数开始执行（A）

A 错

B 对

13:

有定义语句：

char s[10];

若要从终端给s输入5个字符，错误的输入语句是（）。

A cin.getline(s,6);

B cin>>s+1;

C cin.gets(s,6);

D cin>>s[1];

14:

下列字符串常量表示中,(B)是错误的。

A "abcd\n"

B "\" yes\"or\"No\"

C "\'OK\'"

D "ABC\0 "

15:

下列与流程控制有关的关键字是（A） 。

A goto

B static

C int

D sizeof

16:

局部变量可以隐藏全局变量，那么在有同名全局变量和局部变量的情形时，可以用（A）提供对全局变量的访问

A域运算符

B类运算符

C重载

D引用

17:

若析构函数是虚函数，则delete对析构函数的调用采用（B）

A.静态联编

B.动态联编

C.重载函数

D.多重函数

18:

下面程序的输出结果 (B)。

＃i nclude<iostream.h>

int fun(char \*s)

{

char \*p=s;

while(\*p!='\0')

p++;

return (p-s);

}

void main()

{

Cout<<fun("ABCDEF")<<endl;

}

A 3

B 6

C 8

D 0

19:

在私有继承中,基类中所有成员对派生类的对象都是不可见的.(A)

A 对

B 错

20:

在下列C++标点符号中，表示复合语句结束的标记符是（C）。

A #

B //

C }

D ;

21:

有如下程序：

#include&lt;iostream&gt;

using namespace std;

class MyClass{

public:

MyClass(int i=0){cout<<1;}

MyClass(const MyClass&x){cout<<2;}

MyClass operator=(const MyClass&x){cout<<3; return \*this;}

~MyClass(){cout<<4;}

};

int main(){

MyClass obj1(1),obj2(2),obj3(obj1);

obj1=obj2;

return 0;

}运行时的输出结果是（）。

A 1123444

B 11114444

C 121444

D 11314444

22:

有关类和对象的说法不正确的是(B)。

A 对象是类的一个实例

B 一个类只能有一个对象

C 任何一个对象只能属于一个具体的类

D 类与对象的关系和数据类型和变量的关系相似

23:

假定一个类的构造函数为 A(int aa,int bb){a=aa++;b=a\*bb++;} ，则执行Ax(4,5); 语句后， x.a 和 x.b 的值分别为 (C) 。

A. 4 和 5

B. 5 和 4

C. 4 和 20

D. 20 和 5

24:

函数模板中模板参数只能有一个(A).

A 错

B 对

25:

下面程序的正确的输出是（C）。

#include<iostream>

void main()

{ enum team{my,your=4,his,her=his+10};

Cout<<my<<’ ‘<<your<<’ ’<<his<<’ ‘<<her<<endl;

}

A 0 1 2 3

B 0 4 0 10

C 0 4 5 15

D 1 4 5 15

26:

指针可以用来表示数组元素，下列表达式中(A)是错误的。??????

已知：int a[3][7];

A \*(a+1)[5]

B \*(\*a+3)

C \*(\*(a+1))

D \*(&amp;a[0][0]+2)

27:

下列关于对象的描述中,错误的是(B)。

A 一个类可以定义多个对象

B 对象是类的实例

C 对象之间通过消息进行通信

D 对象是类的别名

28:

友元的作用是（A）

**A** 提高程序的运行效率

B 加强类的封装性

C 实现数据的隐藏性

D 增加成员函数的种类

29:

关于构造函数的说法，不正确的是：（A）

A没有定义构造函数时，系统将不会调用它

B 其名与类名完全相同

C 它在对象被创建时由系统自动调用

D 没有返回值

30:

下面关于类的对象性质的描述，正确的是（D）。

A 对象之间不可以相互赋值

B 不可以说明指向对象的指针

C 对象不能用作数组元素

D 一个对象能用作另一个对象的成员

31:

关于delete 运算符的下列描述中，（B）是错误的。

A 它必须用于new返回的指针

B 它也适用于空指针

C 对一个指针可以使用多次该运算符

D指针名前只用一对方括号符，不管所删除数组的维数

32:

下列for语句的循环次数为（B）。

for(int i=0, x=0; !x &&i<=5; i++);

A 5次

B 6次

C 7次

D 无穷次

33:

析构函数是一种函数体为空的成员函数.(A)

A 错

B 对

34:

在基类和派生类之间定义，虚函数的函数原型必须完全一致？()？？？？

A 错

B 对

35:

不属c++源程序的翻译过程的是(A)。

A 解释分析

B 词法分析

C 语法分析

D 链接

36:

定义数组长度时，其“元素个数”允许的表示方式是（A）。

A 整型常量

B 整型表达式

C 整型常量或整型表达式

D 任何

37:

下列程序输出的结果是（B）。

#include<iostream>

using namespace std;

void fun();

int main()

{

for(int i=1; i<3; i++)

fun();

return 0;

}

void fun()

{

static int n = 10;

n++;

cout<<"n="<<n<<endl;

}

A n=11 n=11

B n=11 n=12

C n=10 n=11

D n=10 n=10

38:

在结构化程序设计中,模块划分的原则是(A)

A 模块内具有高内聚度,模块间具有低耦合度

B 各模块应包括尽量多的功能

C 各模块之间的联系应尽量紧密

D 各模块的规模应尽量大

39:

关于类模板的说法正确的是：（B）

A 类模板的主要作用是生成抽象类

B 类模板实例化时，编译器将根据给出的模板实参生成一个类

C 在类模板中的数据成员具有同样类型

D 类模板中的成员函数没有返回值

40:

有以下程序：

void main()

{

char s[]="abcde";

s+=2;

cout<<s[0]<<endl;

}

程序运行后的输出结果是（D）。

A 输出字符a的ASCII

B 输出字符c的ASCII码

C 输出字符c

D 程序出错

41:

采用函数重载的目的在于 ( B)。

A实现共享

B减少空间

C提高速度

D使用方便，提高可靠性

42:

C++流中的重载运算符<<;,它是一个（D）

A 用于输出操作的成员函数。

B 用于输入操作的成员函数。

C 用于输出操作的非成员函数。

D 用于输入操作的非成员函数。

43:

类MyA的拷贝初始化构造函数是（C）

A. MyA()

B. MyA(MyA\*)

C. MyA(MyA&)

D. MyA(MyA)

44:

在语句cin<<data;中，cin是(C)

A C++的关键字

B 函数名

C 类名

D 对象名

45:

在下列的各类函数中，（C）不是类的成员函数。

A 构造函数；

B 析构函数；

C 友元函数；

D 拷贝初始化构造函数

46:

下列有关重载函数的说法中正确的是（C）

A. 重载函数必须具有不同的返回值类型

B. 重载函数参数个数必须相同

C. 重载函数必须有不同的形参列表

D. 重载函数名可以不同

47:

Which one of the following is not a valid identifier in C++(D)?

A myName

B little

C X123Y

D Go Home

E IdEnTiFiEr

48:

下列字符串中不可以用作C++标示符的是（C） 。

A\_521

B computer123

C int

D flower

49:

下面程序的输出结果为 (B)。

＃i nclude<iostream.h>

void prt(int \*x,int \*y,int \*z)

{

Cout<<++\*x<<","<<+ + \*y<<","<<\*(z++)<<endl;

}

int a=10;

int c=20;

int b=40;

void main()

{

prt(&a,&b,&c);

prt(&a,&b,&c);

}

A 11,42,31 12,21,41

B 11,41,20 12,42,20

C 11,21,40 11,21,41

D 11,41,21 12,42,22

50:

x为int型，s为float型,x=3,s=2.5。表达式s+x/2的值为(B)。

A 4

B 3.5

C 2.5

D 3

51:

关于下面程度段说法正确的是（A）.

class X

{ private: int n; punblic: X(X&); } X::X(X&x) {n=x.n} X obj1,obj2;

obj2(obj1);

A 语句obj2（obj1）；的功能是用对象obj1初始化具有相同类类型的对象obj2

B 语句obj2（obj1）；的功能是用对象obj2初始化具有相同类类型的对象obj1 C X(X&amp;x)也可以不使用引用参数而仍然是拷贝构造函数

D X(X&amp;x)不能访问对象的私有成员

52:

当一个类的某个函数被说明为virtual时，在该类的所有派生类中的同原型函数(A)

A 都是虚函数

B 只有被重新说明时才是虚函数

C 都不是虚函数

D 只有被重新说明为virtual时才是虚函数

53:

不能作为重载函数的调用的依据是(D)。

A 参数个数

B 参数类型

C 函数类型

D 函数名称

54:

在int b[][3]={{1},{3,2},{4,5,6},{O}};中b[2][2]的值是( C)。

A 5

B 0

C 6

D 2

55:

友元运算符obj++被C++编译器解释为（B ）。

A operator++(obj)

B operator++(obj,0)

C obj.operator++( )

D obj.operator++(0)

56:

如果你不需要返回任何值，你将选择下列哪个函数声明？ (B)

A myfunction(void)

B void myfunction()

C myfunction()

57:

已知: int m=10；下列表示引用的方法中，正确的是：（A）

A)int &Z；

B)int &t=10；

C)int &X=m；

D)float &f=&m；

58:

下列正确的字符常量为（D）

A ”a”

B ’\81’

C a

D’\101’

59:

有如下模板定义：

Template<class T>T fun(T x,T y){return x\*x+y\*y；} 在下列对fun的调用中，正确的是：(A)

A fun(2.3,8)

B fun(2,8)

C fun(2.0,8.2)

D fun<float>(2,8.3)

60:

( B)是给对象取了一个别名，他引入了对象的同义词。

A 指针

B 引用

C 枚举

D结构

61:

类模板的使用实际上是类模板实例化成一个具体的（A）.

A 类

B 对象

C 函数

D 模板类

62:

若Sample类中的一个成员函数说明如下：void set(Sample& a)，则Sample& a的含义是（C）

A. 指向类Sample的名为a的指针

B. a是类Sample的对象引用，用来作函数Set（）的形参

C. 将a的地址赋给变量Set

D. 变量Sample与a按位与的结果作为函数Set的参数

63:

对于基类型相同的两个指针变量之间，不能进行的运算是（C）。

A &lt;

B =

C +

D －

64:

假定下列语句都是程序运行后首次执行的输出语句，其中输出结果与另外三条语句不同的语句是(C)

A cout<<setfill('\*')<<123<<setw(9)<<321;

Bcout<<setfill('\*')<<setw(6)<<left<<123<<setw(6)<<right<<321;

C cout<<123<<setfill('\*')<<setw(6)<<321;

D cout<<setfill('\*')<<setw(9)<<left<<123<<321

65:

在编译指令中，宏定义使用哪个指令（B）

A #include

B #define

C #if

D #els;

66:

所谓多态性是指 （B）

A 不同的对象调用不同名称的函数

B 不同的对象调用相同名称的函数

C 一个对象调用不同名称的函数

D 一个对象调用不同名称的对象

67:

已知：p是一个指向类A数据成员m的指针，A1是类A的一个对象。如果要给m赋值为5，（C）是正确的。

A A1.p = 5

B A1->p = 5

C A1.\*p = 5

D \*A1.p = 5

68:

适宜采用inline定义函数情况是（A）

A 函数体含有循环语句

B 函数体含有递归语句

C 函数代码少、频繁调用

D 函数代码多、不常调用

69:

删除动态对象，需要使用操作符（B）

A new

B delete

C .

D \*

70:

在面对对象方法中，实现信息隐蔽是依靠(A)

A 对象的封装

B 对象的分类

C 对象的继承

D 对象的多态

71:

设int a=1,b=2,c=3,则表达式a<b!=c的值是(A)

A 0

B 1

C 2

D 3

72:

在下列函数原型中，可以作为类AA构造函数的是(D)。

A void AA(int);

B int AA();

C AA(int)const;

D AA(int);

73:

关于成员函数的特征的下述描述中，错误的是（A）

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

74:

如果你希望让一个外部函数访问类的私有和保护成员，你需要这样声明()？？？？

A myClass myFunction(friend myClass);

B friend myClass myFunction(myClass);

C void myClass myFunction(myClass);

75:

下列字符常量表示中,(C)是错误的。

A '\a'

B '\*'

C '\4f'

D '\105'

76:

下列关于C++与C语言关系的描述中错误的是 (C) 。

A C++是C语言的超集

B C++是C语言进行了扩充

C C++和C语言都是面向对象的程序设计语言

D C++包含C语言的全部语法特征

77:

设置虚基类的目的是(B)。

A 简化程序

B 消除二义性

C 提高运行效率

D 减少目标代码

78:

下列程序中说明的公有成员是(D).

class Location{

int X;

public:

void init(int initX,int initY);

private:

int Y;

public:

int GetX();

int GetY();

}

A X

B init(int initX,int initY)

C GetX(),GetY()

D B 和C都是

79:

有如下程序段，请指出\*\*p所代表的是(D)。

int \*\*p,\*s,k=20;

s=&k;

p=&s;

A s

B &k

C \*p

D k

80:

不能作为函数重载判断依据的是：（D）

A 参数个数

B 参数类型

C 函数名字

D 返回类型

81:

下面对静态数据成员的描述中，正确的是（B）

A 静态数据成员可以在类体内进行初始化

B 静态数据成员可以直接用类名或者对象名来调用

C 静态数据成员不能用private控制符修饰

D 静态数据成员不可以被类的对象调用

82:

编写C++程序一般需经过的几个步骤依次是(B)

A编辑，调试，编译，连接

B编辑，编译，连接，调试

C编译，调试，编辑，连接

D编译，编辑，连接，运行

83:

类example 的析构函数为（D）

**A** example（）；

**B** example（）

**C** ~example（）；

**D** ~example（）

84:

关于对象概念的描述中，说法错误的是（C）

A 对象是类的一个变量

B 对象代表着正在创建的系统中的一个实体

C 对象就是C语言中的结构变量

D 对象之间的信息传递是通过消息进行的

85:

在C++中，合法的长整型常数是（A）。

A 0L

B 4962710

C 0.054838743

D 2.1869e10

86:

一部分用指定的数据类型和长度来执行特殊功能的代码被称作(D)

A 链表

B 类

C 函数

D 数组

87:

以下对i, j, k进行说明并赋值的方法中, 只有（D）是错误的。

A int i=1,j=1,k=1;

B int i,j,k; i=j=k=1;

C int i=1,j=1,k=1;

D int i=j=k=1;

88:

下面叙述不正确的是（D）

A 派生类一般都用公有派生

B 对基类成员的访问必须是无二义性的

C 赋值兼容规则也适用于多重继承的组合

D 基类的公有成员在派生类中仍然是公有的

89:

设int x[] = {1，2，3，4，5，6}，\* p = x；则值为3的表达式是（B）

A p + = 2, \* + + p

B p + = 2, \* p + +

C p + = 3, \* p

D p + = 2, + + \* p

90:

下面叙述不正确的是（A）

A 基类的保护成员在派生类中仍然是保护的

B 基类的保护成员在公有派生类中仍然是保护的

C 基类的保护成员在私有派生类中仍然是私有的

D 对基类成员的访问必须是无两义性

91:

在 C++程序中出现（C）情况时，将引发一个异常。

a 在 main 函数的结尾少了一个大括号

b for 循环语法错误

c 被零除或数组溢出

d 数组没有被赋满值

92:

以下程序的输出结果是(B)

#include<iostream.h>

void main()

{

char s[]="123",\*p;

p=s;

cout<<\*p++<<endl;

}

A 123

B 1

C 3

D 321

93:

以下定义，

struct st1{int a,b;float x,y;};

struct st2{int a,b;st1 s1; } ss;

对成员变量x的引用形式正确的是：(A)。

A ss.s1.x

B st2.st1.x

C st2.s1.x

D ss.x

94:

在C++中，关于下列设置参数默认值的描述中，正确的是( C)。

A 不允许设置参数的默认值

B 设置参数默认值只能在定义函数时设置

C 设置参数默认值时，应该事先设置右边的再设置左边的

D 设置参数默认值时，应该全部参数都设置

95:

下面程序的输出结果 (B)。

＃include<iostream.h>

int fun(char \*s)

{

char \*p=s;

while(\*p!='\0')

p++;

return (p-s);

}

void main()

{

Cout<<fun("ABCDEF")<<endl;

}

A 3

B 6

C 8

D 0

96:

若派生类的成员函数不能直接访问基类中继承来的某个成员，则该成员一定是基类中的（D）

A私有成员

B公有成员

C保护成员

D保护成员或私有成员

97:

以下程序的输出结果是：（D）

#include<iostream.h>

int fun(char \*s)

{

char \*p=s;

while (\*p!='\0') p++;

return (p-s);

}

void main()

{

cout<<fun("abc")<<endl;

}

A 0

B 1

C 2

D 3

98:

设int x[]={1,3,5,7,9,11,13},\*p=x;则不能正确引用数组元素的表达式是（B）。

A \*(p――)

B \*(――p)

C \*(p++)

D \*(++p)

99:

关于类概念的描述中，（D）是错误的。

A 类是抽象数据类型的实现

B 类是具有共同行为的若干对象的统一描述体

C 类是创建对象的样板

D 类不是C＋＋中的一种数据类型

100:

有如下程序：执行后的输出结果应该是：（A）

#include<iostream.h>

class A

{

public:

A(){cout&lt;&lt;"A"；}

}；

class C:public A

{

public:

C(){cout<<"C"；}

}；

void main()

{

C cobj；

}

A) AC

B) CA

C) A

D) C

101:

缺省析构函数的函数体是（C）

Ａ不存在

Ｂ随机产生的

Ｃ空的

Ｄ无法确定的

102:

定义派生类时，若不使用关键字显式地规定采用何种继承方式，则默认方式为(A)

A 私有继承

B 非私有继承

C 保护继承

D 公有继承

103:

关于虚基类，下面说法正确的是(D)

A 虚基类可以用来解决二义性问题

B 虚基类不能实例化

C 带有虚函数的类称为虚基类

D 带有纯虚函数的类称为虚基类

104:

下列字符串中不可以用作C++标示符的是（C）

A \_521

B computer123

C int

D flower

105:

下面选项中不属于面向对象程序设计特征的是(C)

A继承性

B多态性

C相似性

D封装性

106:

如果B类继承A类， A::x()被声明为虚函数， B::x()覆盖A::x()哪一种方法可以使x()被B的对象b调用(A)

A B::x()

B A::x()

C A::x() followed by B::x()

D B::x() followed by A::x()

107:

设int x[] = {1，2，3，4，5，6}，\* p = x；则值为3的表达式是（B）

A p + = 2, \* + + p

B p + = 2, \* p + +

C p + = 3, \* p

D p + = 2, + + \* p

108:

函数调用时，下列说法中不正确的是（A）。

A 若用值传递方式，则形式参数不予分配内存

B 实际参数和形式参数可以同名

C 主调函数和被调函数可以不在同一个文件中

D 函数间传递数据可以使用全局变量

109:

下面说法正确的是（B）

A 类的内联函数必须在类体内定义

B 内联函数在运行时是将该函数的目标代码插入每个调用该函数的地方

C 类的内联函数必须在类体外关键字inline定义

D 内联函数在编译时是将该函数的目标代码插入每个调用该函数的地方

110:

假定 AB 为一个类，则执行 AB x ；语句时将自动调用该类的 (B) 。

A 有参构造函数

B 无参构造函数

C 拷贝构造函数

D 赋值构造函数

111:

可用作C++语言用户标识符的一组标识符是(B)。

A void define +WORD

B a3\_b3 \_123 YN

C for -abc Case

D 2a DO sizeof

112:

C++程序中的语句必须以（B）结束。

A 冒号

B 分号

C 空格

D 花括号

115:

下列有关继承和派生的叙述中，正确的是(B)

A 派生类不能访问通过私有继承的基类的保护成员

B 多继承的虚基类不能够实例化

C 如果基类没有默认构造函数，派生类就应当声明带形参的构造函数

D 基类的析构函数和虚函数都不能够被继承，需要在派生类中重新实现

116:

记录流的当前格式化状态的标志字中每一位用于记录一种格式,这种格式是不能被设置或清除的.（A）

A 错

B 对

117：

C++语言允许使用友元,友元不会破坏封装性.(B)

A 对

B 错

118:

下列运算符中可用于实数的是（B）

A %

B ――

C <<

D &

119:

有如下程序段，请指出\*\*p所代表的是(D)。

int \*\*p,\*s,k=20;

s=&k;

p=& s;

A s

B &k

C \*p

D k

120:

设有定义int i;double j＝5;，则10+i+j值的数据类型是（B）

A int

B double

C float

D 不确定

121:

C++源程序文件的缺省扩展名为(A)。

A. cpp

B. exe

C. obj

122:

设int n=11，i=4； 则赋值运算执行后n的值是（D）。

A 0

B 3

C 2

D 11

123:

在下列八进制整型常量的表示中，（B）是正确的。

A 0a0

B 015

C 080

D 0x10

124:

在私有继承中,基类中所有成员对派生类的对象都是不可见的.(A)

A 对

B 错

125:

下列关于do…while语句的叙述正确的是（D）。

A do…while语句所构成的循环不能用其他语言构成的循环来代替。

B do…while语句所构成的循环只能用break语言跳出。

C do…while语句所构成的循环只有在while后面的表达式非零时才结束。

D do…while语句所构成的循环只有在while后面的表达式为零时才结束。

126：

下列for语句的循环次数为（B）。

for(int i=0, x=0; !x && i<=5; i++);

A 5次

B 6次

C 7次

D无穷次

127:

下列选项中，用于清除基数格式位置以十六制输出的语句是（B）

A cin>>setf(ios::hex, ios::basefield);

B cout<<setf(ios::dec, ios::basefield);

C cout<<setf(ios::oct, ios::basefield);

D cout<<setf(ios::hex, ios::basefield);

128：

下面程序的输出结果是 (D)。

＃i nclude<iostream.h>

void fun(int \*s,int \*y)

{

static int t=3;

\*y=s[t];

t--;

}

void main()

{

int a[]={1,2,3,4};

int i,x=10;

for(i=0;i<4;i++)

{

fun(a,&x);

cout<<x<<",";

}

cout<<endl;

}

A 1,2,3,4

B 0,0,0,0

C 4,4,4,4

D 4,3,2,1

129: 类的构造函数被自动调用执行的情况是在定义该类的(C)

A 成员函数时

B 数据成员时

C 对象时

D 友元函数时

130: 下列选项中属于C++语句的是（A）。

A ;

B a = 17

C i + 5

D cout << ‘\n’

131: 下列程序的输出结果是（B）

#include <iostream.h>

void main()

{int n［］［3］={10,20,30,40,50,60};

int (\*p)［3］;

p=n;

cout<<p［0］［0］<<"，"<<\*(p［0］+1)<<"，"<<(\*p)［2］<<endl;}

A 10，30，50

B 10，20，30

C 20，40，60

D 10，30，60

132: 如何声明无类型的指针（D）

A int\* data

B void data

C null\* data

D void \* data

133:

使用地址作为实参传给形参，下列说法正确的是（D）

A 实参是形参的备份

B 实参与形参无联系

C 形参是实参的备份

D 实参与形参是同一对象

134:

在下列关于C++与C语言关系的描述中，错误的是（D）。

A C++是C语言的超集

B C++对C语言进行了扩充

C C++包含C语言的全部语法特征

D C++与C语言都是面向对象的程序设计语言

135:

在C++中，关于下列设置参数默认值的描述中，正确的是(C)。

A 不允许设置参数的默认值

B 设置参数默认值只能在定义函数时设置

C 设置参数默认值时，应该事先设置右边的再设置左边的

D 设置参数默认值时，应该全部参数都设置

136:

下列关于析构函数的描述中，不正确的是（D）

A析构函数有且只有一个

B析构函数可以有参数

C析构函数没有函数类型

D析构函数的作用是在对象被撤销时收回其内存空间

137:

语句以（B）符号作为结束标记。

A ,

B ;

C 空格

D :

138:

执行下列语句后的输出结果是（B）。

int hot=100;

int &rad=hot;

hot+=100;

cout<< rad << endl ;

A 0；

B 100；

C 200；

D 300；

139:

下列（D）表达式不满足：当x的值为偶数时值为“真”，为奇数时值为“假”。

A (x/2\*2-x)==0

B !(x%2)

C x%2==0

D !x%2!=0

140:

下列关于运算符重载的描述中，错误的是(C)

A ::运算符不能重载

B 类型转换运算符只能作为成员函数重载

C 将运算符作为非成员函数重载时必须定义为友元

D 重载[ ]运算符应完成“下标访问”操作

141:

已知example是已定义的类,则程序结束前,共存在(B)个对象.

#include<example.h>

main()

{ example x(5);

example arr[2]={1,2};

x=10;

x.print();

x=example(15);

x.print();

}

A 3

B 4

C 7

D 8

142:

在面向对象的程序设计中，首先在问题域中识别出若干个 （B）

A 函数

B 类

C 文件

D 过程

143:

下面程序的输出结果是 (A)。

#i nclude<iostream.h>

int fun(char \*s)

{

char \*p=s;

while(\*p!='\0')

p++;

return (p-s);

}

void main()

{

cout<<fun("ABCDEFGHIJKLMNIPQRST")<<endl;

}

A 20

B 6

C 19

D 0

144: 以下为死循环的程序段是（B）。

A for ( int x=0; x<3; x-- ) x+=1;

B int k = 0;do { ++k; } while ( k>=0 );

C int a = 5; while ( a ) { a--; }

D int i = 3; for ( ; i; i-- );

145:

下列程序段的输出结果为(C).

#include <iostream.h>

void main()

{

int x=3,y=2;

cout<<(x-=y,x\*=y+8/5);

}

A 1

B 7

C 3

D 5

146:

对定义重载函数的下列要求中， (D)是错误的。

A要求参数的个数不同

B要求参数中至少有一个类型不同

C要求参数个数相同时，参数类型不同

D要求函数的返回值不同

147:

#include<iostream>

using namespace std;

class Test{

public:

Test(){}

~Test(){cout<<'#';}

};

int main(){

Test temp[2], \*pTemp[2];

return 0;

}

执行这个程序输出星号（#）的个数为（B）。

A 1

B 2

C 3

D 4

148:

类TM和类TN是两个独立的类，那么类TM中哪种形式的数据成员一定是不允许的：(C)

A TM \* a

B TN & a

C TM a

D TN a；

149:

对于语句 cout<<endl<<x; 中的各个组成部分，下列叙述中错误的是(D)。

A cout 是一个输出流对象

B endl 的作用是输出回车换行

C x 是一个变量

D << 称作提取运算符

150:

以知AA是一个类，则执行语句AA a；后（A）

A 自动调用无参构造函数

B 自动调用有参构造函数

C 自动调用拷贝构造函数

D 产生一个对象a,但不调用任何函数

151:

语句 int (\*p)( ); 的含义是（A）

A p 是一个指向函数的指针变量，该函数的返回值是一个整型数据

B p 是指针变量，指向一个整型数据

C p 是一个指向一维数据的指针变量

D p 是指针函数

152:

有关构造函数的说法不正确的是（D）。

A 构造函数名字和类的名字一样

B 构造函数在定义类的对象时自动执行

C 构造函数无任何函数类型

D 构造函数有且只有一个

153:

下列定义中，x是一个类,(B)是定义指向对象数组的指针p.

A x \*p[4]

B x (\*p)[4]

C (x\*)p[4]

D x \*p[]

154:

在多继承中，公有派生和私有派生对于基类成员的派生类中的可访问性与单继承的规则(A)

A 完全相同

B 完全不同

C 部分相同，部分不同

D 以上都不对

155:

关于常数据成员的说法不正确的是(C)

A常数据成员的定义形式与一般常量的定义形式相同，只不过常数据成员的定义必须出现在类体中.

B常数据成员必须进行初始化,并且不能被更新

C常数据成员通过构造函数的成员初始化列表进行初始化

D常数据成员可以在定义时直接初始化

156:

若有以下说明

int a[10]={0,1,2,3,4,5,6,7,8,9,10}, \*p=a;

则数值为6的表达式是 （C）

A \*a+5

B \*(a+6)

C \*(p+5)

D p+5

157:

有如下类声明“class A{int x； ……}；”，则A类的成员x是：（B）

A 公有数据成员

B 私有数据成员

C 公有成员函数

D 私有成员函数

158:

函数模板中模板参数只能有一个.（）????

A 错

B 对

159:

下列函数中,(D)不能重载.

A 成员函数

B 构造函数

C 非成员函数

D 析构函数

160:

在下列八进制整型常量的表示中，（B）是正确的。

A 0a0

B 015

C 080

D 0x10

161:

函数调用func((exp1,exp2),(exp3,exp4,exp5))中所含实参的个数为（B）

A 1

B 2

C 4

D 5

162:

以下程序段的结果是（B）。

int m=0, n=0;

if (++m && ++n )

cout << m << "," << n << endl;

A 0,0

B 1,1

C 1,0

D 0,1

163:

在C++中，要实现动态联编，必须使用(D)调用虚函数。

A 类名

B 派生类指针

C 对象名

D 基类指针

164:

在以下给出的表达式中,与 do--while(E)语句中的(E)不等价的

表达式是(C).

A (!E==0)

B (E>0||E<0)

C (E==0)

D (E!=0)

165:

已知int i=3; 下面do\_while语句执行时循环次数为（B）。

do {i--; cout << i << endl;} while(i != 1);

A 1

B 2

C 3

D 无限

166：

下列关于常对象和常成员的描述中，正确的是(A)

A类的普通成员函数不能对常对象进行操作

B类的成员函数可以对常对象进行操作

C除常成员函数外，其他函数不可以对常对象进行操作

D在常成员函数中，常对象可以被更新

167：

成员函数的特征的下述描述中，错误的是（A）

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

168:

已知：int a, b; 下列switch语句中，（D）是正确的。

A switch(a){case a: a++; break; case b: b++; break;}

B switch(a + b){case 1: a + b; break; case 1: a - b;}

C switch(a \* a){case 1, 2: ++a; case 3, 4: ++b;}

D switch(a/10 + b){case 5: a/5; break; default : a + b;}

169:

下列引用的定义中,(D)是错误的。

A int i; int &j=i;

B float i; float &j=i;

C char d; char &k=d;

D int i; int &j; j=i

170:

对于类定义

class A{

public:

virtual void funcl(){}

void func2(){}

};

class B:public A{

public:

void func1(){cout<<”class B func1”<<endl;}

virtual void func2(){cout<<”class B func2”<<endl;}

};

下面正确的叙述是（A）

A A::func2()和B::func1()都是虚函数

B A::func2()和B::func1()都不是虚函数

C B::func1()是虚函数，A::func2()不是虚函数

D B::func1()不是虚函数，A::func2()是虚函数

171:

下面字符序列中，合法的常量是(B)。

A 078

B 5L

C 9e7.5

D ’a1’

172:

C++语言建立类族是通过（B）

A 类的嵌套

B 类的继承

C 虚函数

D 抽象类

173:

假设ClassY:publicX，即类Y是类X的派生类，则说明一个Y类的对象时和删除Y类对象时 ，调用构造函数和析构函数的次序分别为（A）

A Y,X；Y,X

B X,Y；X,Y

C X,Y；Y,X

D Y,X；X,Y

174:

字符串常量“ME”的字符个数是（C）。

A 1

B 2

C 3

D 4

175:

有如下程序：

#include<iostream>

using namespace std;

class MyClass{

public:

MyClass(int i=0){cout<<1;}

MyClass(const MyClass&x){cout<<2;}

MyClass& operator=(const MyClass&x){cout<<3; return \*this;}

~MyClass(){cout<<4;}

};

int main(){

MyClass obj1(1),obj2(2),obj3(obj1);

obj1=obj2;

return 0;

}运行时的输出结果是（A）。

A 1123444

B 11114444

C 121444

D 11314444

176:

设a和b为整型变量，执行语句b=(a=2+3,a\*4),a+5;后a和b的值为（A） 。

A 5，10

B 20，25

C 5，25

D 5，20

177:

要求打开文件"d:\file.dat"，可写入数据，正确的语句是：（D）

A ifstream infile("d:\file.dat", ios::in)；

B ifstream infile("d:\\file.dat", ios::in)；

C ofstream infile("d:\file.dat", ios::out)；

D fstream infile("d:\\file.dat", ios::in|ios::out)；

178:

下列字符串中不可以用作C++标示符的是 （C） 。

A \_521

B computer123

C int

D flower

179:

当在使用类模板时，类模板本身被实例化成了一个具体的（C）。

A 对象

B 实例

C 模板类

D 类

180:

下面关于模版说法正确的是（A）。

a 类模版提供了一种对类中类型进行参数化的方法; 在实例化模版类时，实际的数据类型会代替 与类成员或方法相关联的类型参数

b 类模版中必须包含类成员与类方法

c 不可以用自定义的数据类型实例化一个模版类

d 类模版中类方法的参数必须用占位符替代，而不能使用实际数据类型

181:

下面程序的输出结果是（C）

#include <iostream>

using namespace std;

class Sample

{

friend long fun (Sample s)

{

if (s.x < 2) return 1;

return s.x \* fun(Sample(s.x-1));

}

public:

Sample (long a) { x = a; }

private:

long x;

};

void main()

{

int sum = 0;

for (int i=0; i<4; i++)

{

sum += fun(Sample(i));

}

cout << sum;

}

A 12

B 16

C 10

D 34

182:

已知f1(int)是类A的公有成员函数，P是指向成员函数f1()的指针，采用下面哪种赋值方式是正确的。(C)

A p=A.f1();

B p=f1()；

C P=A::f1();

D p=fl；

183:

设有如下定义int (\*ptr)();，下面关于ptr正确叙述是（C）。

A ptr是指向一维数组的指针变量。

B ptr是指向int 型数据的指针变量。

C ptr是指向函数的指针，该函数返回一个int型数据。

D ptr是一个函数名，该函数的返回值是指向int型数据的指针。

184:

下列关于构造函数说法不正确的是(C)。

A 构造函数必须与类同名

B 构造函数可以省略不写

C 构造函数必须有返回值

D 在构造函数中可以对类中的成员进行初始化

185:

某结构体变量定义如下，对此结构体变量的成员的引用形式正确的是（B）

struct st{int a,b; float x,y;} s1,\*p;

p=&s1;

A s1->a

B p->b

C p.x

D \*p.y

186:

下面类模板的定义中，不正确的是（）？？？？？？

A template<class T>

class A{T x,y;};

B template<class T>

class A{private:T x;};

C template<class T>

class A{public:T x;};

D template<class T>

class A{protected:t x;};

187:

以下程序段的结果是（C）。

int m=0, n=0;

if(m++ && n++)

m++;

cout << m << “,” << n << endl;

A 0,0

B 1,1

C 1,0

D 2,1

188:

文件包含命令中被包含的文件的扩展名(D)。

A 必须为.h

B 必须是.c

C 不能用.h

D 不一定是.h

189:

在下面有关析构函数特征的描述中，正确的是：（C）

A 一个类中可以定义多个析构函数

B 析构函数名与类名完全相同

C 析构函数不能指定返回类型

D 析构函数可以有一个或多个参数

190:

所谓数据封装就是将一组数据和与这组数据有关数据有关操作组装在一起，形成一个实体，这实体也就是（A）

A类

B 对象

C 函数体

D 数据块

191:

有如下表达式19/3\*sqrt(4.0)/5，则该表达式值数据类型为（C）。

A int

B float

C double

D 不确定

192:

当一个函数无返回值时，函数的类型应定义为 (A)。

A void

B 任意

C int

D 无

193:

下面程序的输出结果是(C)。

＃include<iostream.h>

int x=10;

void fun(int c);

void main()

{

fun(x);

cout<<"x="<<x<<endl;

}

void fun(int x)

{

x=1;

}

A)x=0

B)x=1

C)x=10

D)语法错误，不能输出正确结果

194:

（C）是一个在基类中说明的虚函数，它在该基类中没有定义，但要求任何派生类都必须定义自己的版本。

A 纯虚函数

B 虚析构函数

C 虚构造函数

D 静态成员函数

195:

对象引用作为函数参数比用对象指针效率更高些(B)

A 错

B 对

196:

在 C++中，可以重载的运算符有（D）。

a sizeof()

b ::

c \*

d ++

197:

已知一个类X,(C)是定义指向类X成员函数的指针,假设类有3个公有成员: void f1(int), void f2(int)和int a.

A X\*P

B int X::\*pc=&X::a

C void(X::\*pa)()

D X\*p[10]

198:

已知：p是一个指向类A数据成员m的指针，A1是类A的一个对象。如果要给m赋值为5，（C）是正确的。

A A1.p = 5

B A1->p = 5

C A1.\*p = 5

D \*A1.p = 5

199: ．

设存在函数int max(int，int)返回两参数中较大值，若求22，59，70三者中最大值，下列表达式不正确的是：（C）

A int m = max(22，max(59，70))；

B int m = max(max(22，59)，70)；

C int m = max(22，59，70)；

D int m = max(59，max(22，70))；

200:

下面描述中，表达错误的是：（B）

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

201:

有一个int型变量，在程序中频繁使用，最好定义它为 (A)。

A register

B auto

C extern

D static

202:

定义变量int a=6，则a++的值是(B)

A 7

B 6

C 5

D 4

203:

以下能定义一维的选项是(B)

A int a[5] ={0,1,2,3,4,5,}

B char a[]={0,`1,2,3,4,5}

C char a={‘A’, ‘B,’ ‘C’}

D int a[5] =”123”

204:

有如下程序段：

int i=4; int j=1;

int main() {

int i=8, j=i;

cout<<i<<j<<endl;

}

运行时的输出结果是(C)

A 44

B 41

C 88

D 81

205:

下面对于析构函数的描述错误的是（B）。

A 类中有且仅有一个析构函数。

B 析构函数可以有形参。

C 析构函数没有函数类型。

D 析构函数在对象消失时被自动执行

206:

类的析构函数是对一个对象进行以下哪种操作时自动调用的是（B）

A 建立

B 撤销

C 赋值

D 引用

207:

对类的构造函数和析构函数描述正确的是 (A)。

A 构造函数可以重载，析构函数不能重载

B 构造函数不能重载，析构函数可以重载

C 构造函数可以重载，析构函数也可以重载

D 构造函数不能重载，析构函数也不能重载

208:

在C++中，能正确表示a≥10或a≤0的关系表达式是(C)

A a>=10 or a<=0

B a>=10│a<=0

C a>=10 || a<=0

D a>=10 && a<=0

209:

下面程序的输出结果是(A)。

＃i nclude<iostream.h>

int fun(char \*s)

{

char \*p=s;

while(\*p!='\0')

p++;

return (p-s);

}

void main()

{

cout<<fun("ABCDEFGHIJKLMNIPQRST")<<endl;

}

A 20

B 6

C 19

D 0

210:

类的静态成员(C)

A 是指静态数据成员

C 是指静态函数成员

C 为该类的所有对象共享

D 遵循类的其他成员做遵循的所有访问权限

211:

下列对结构及其变量定义错误的是：（D）

A struct MyStruct{ int num; char ch; }; struct MyStruct my;

B struct MyStruct{ int num; char ch; }; MyStruct my;

C struct{ int num; char ch; } my;

D struct{int num;char ch;};struct my;

212:

如果没有为一个类定义任何构造函数的情况下，下列描述正确的是：（A）

A 编译器总是自动创建一个不带参数的构造函数

B 这个类没有构造函数

C 这个类不需要构造函数

D 该类不能通过编译

213:

在一个派生类的对象结束其生命周期时 (A)

A先调用派生类的析构函数后调用基类的析构函数

B先调用基类的析构函数后调用派生类的析构函数

C如果基类没有定义析构函数，则只调用派生类的析构函数

D如果派生类没有定义析构函数，则只调用基类的析构函数

214:

表达式的结果是a = 11 % 3（D）

A 0

B 1

C 3

D 2

215:

已知fun(int)是类Test的公有成员函数，p是指向成员函数fun()的指针，采用（B）是正确的:

A p=fun;

B p=Test::fun();

C p=fun();

D p=Test::fun;

216:

不属c++源程序的翻译过程的是(A)。

A 解释分析

B 词法分析

C 语法分析

D 链接

217:

友元关系不能（D）

A 提高程序的运行效率

B 是类与类的关系

C 是一个类的成员函数与另一个类的关系

D 继承

218:

在下列运算符中，（D）优先级最低。

A ！

B &&

C !=

D ?:

219:

有以下程序：

void sum(int \*a)

{ a[0]=a[1]; }

void main()

{

int aa[10]={1,2,3,4,5,6,7,8,9,10}, i;

for(i=2; i>=0; i--)

sum(&aa[i]);

cout<<aa[0]<<endl;

}

执行后的输出结果是(A)。

A 4

B 3

C 2

D 1

220:

（B）是不可以作为该类的成员的。

A 自身类对象的指针；

B 自身类的对象；

C 自身类对象的引用；

D 另一个类的对象。

221:

用于类中虚成员函数说明的关键字（A）

A virtual

B public

C protected

D private

222:

C++中数组元素的下标是从0开始,数组元素是连续存储在内存单元中的。(B)

A 错

B 对

223:

下面关于类概念的描述中，（D）是错误的。

A 类是抽象数据类型的实现

B 类是具有共同行为的若干对象的统一描述体

C 类是创建对象的样板

D 类就是C语言中的结构体类型

224:

指向类X中类型为type的数据成员的指针的说明形式为( )？？？？

A type X::\*pointer

B X::type\*pointer

C type \*X::pointer

D X::\*type pointer

225:

在函数的返回值类型与返回值表达式的类型的描述中，错误的是(B )。

A函数返回值的类型是在定义函数时确定，在函数调用时是不能改变的

B函数返回值的类型就是返回值表达式的类型

C函数返回值表达式类型与返回值类型不同时，函数表达式类型应转换成返回值类型

D函数返回值类型确定了返回值表达式的类型

226:

下列虚基类的声明中正确的是（D）

A． class virtual B:public A

B．virtual class B:public

C． class B:public A virtual

D．class B: virtual public A

227:

有关类和对象的说法不正确的是(B).

A 对象是类的一个实例

B 一个类只能有一个对象

C 任何一个对象只能属于一个具体的类

D 类与对象的关系和数据类型和变量的关系相似

228:

已知int a[] = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9} 和 \*p = a，则不能表示数组a中元素的式子是（A）。

A \*a

B \*p

C a

D a[p - a]

229:

设int a=3,b=4,c=5;表达式(a+b)>c&&b==c的值是(C)。

A 2

B -1

C 0

D 1

230：

关于for循环不正确的是是（A）。

A 在循环体中，不能改变循环控制变量值

B 在执行中循环控制变量初值、终值可变

C 循环控制变量可增可减

D 循环控制变量不必在说明部分说明

231:

（B）是不可以作为该类的成员的。

A 自身类对象的指针；

B.自身类的对象；

C.自身类对象的引用；

D.另一个类的对象。

232:

已知函数FA调用FB，若要把这两个函数定义在同一个文件中，则(C)

A 若FA定义在FB之后，则FA的原型必须出现在FB的定义之前

B FA必须定义在FB之前

C FB必须定义在FA之前

D 若FB定义在FA之后，则FB的原型必须出现在FA的定义之前

233:

可以用“px->\*p”来访问类A的对象a 的数据成员，则下面正确的（B）

A px是指向类数据成员的指针

B p是指向类数据成员的指针

C p是指向类的对象的指针

D 无法确定指针类型

234:

在公有继承中,基类中的公有成员和私有成员在派生类中都是可见的.(A)

A 错

B 对

235:

下面不属于类的继承方式的是(A)

A operator

B private

C protected

D public

236:

在一个C++程序中，main函数的位置（C）。

A 必须在程序的开头

B 必须在程序的后面

C 可以在程序的任何地方

D 必须在其它函数中间

237:

派生类的继承方式中有两种:公有继承和私有继承.(B)

A 对

B 错

238:

能表示a≥10或者a≤0的关系表达式是（C）。

A a>=10 or a<=0

B a>=10 | a<=0

C a>=10 || a<=0

D a>=10&&a<=0

239:

说明友元函数使用关键字(A)

A friend

B const

C colatile

D static

240:

以下关于this指针的叙述中正确的是：（D）

A 任何与类相关的函数都有this指针；

B 类的成员函数都有this指针；

C 类的友员函数都有this指针；

D 类的非静态成员函数才有this指针；

241:

下面对模板的声明，正确的是（C）

A template<T>

B template<class T1,T2>

C template<classT1,classT2>

D template<classT1;classT2>

242:

有以下程序：

void sum(int \*a)

{ a[0]=a[1]; }

void main()

{

int aa[10]={1,2,3,4,5,6,7,8,9,10}, i;

for(i=2; i>=0; i--)

sum(&aa[i]);

cout<<aa[0]<<endl;

}

执行后的输出结果是(A)。

A 4

B 3

C 2

D 1

243:

下面程序段中，说法正确的是

class location{

public :

int GetX();

};

class rectangle:private location {

public:

int GetW();

};(C)

A 类location中的public成员在类 rectangle中是不可访问的

B 类rectangle私有派生类 location

C viod f( ) { Rectangle r; Int X=r .GetX( ); } 其中int X = r .GetX( );是非法语句

D 类rectangle中的public成员在类 location中是私有的

244:

有下面类的说明，有错误的语句是（A）。

class X

{

A int a=2;

B X();

public:

C X(int val);

D ~X();

};

245: 关于动态关联的下列叙述中，（D）是错误的。

A 动态关联是以虚函数为基础的

B 动态关联调用虚函数操作是指向对象的指针或引用

C 动态关联是在运行时确定所调用的函数代码的

D 动态关联是在编译时确定操作函数的

246:

设有 int a[10], \*p=a ; 则下面哪种表示与a[5]不等价（D）。

A \*(a+5)

B \*(p+5)

C p[5]

D p+5

247:

有定义语句：

char s[10];

若要从终端给s输入5个字符，错误的输入语句是（B）。

A cin.getline(s,6);

B cin>>s+1;

C cin.gets(s,6);

D cin>>s[1];

248:

以下程序的输出结果是：（D）

#include <iostream.h>

void func(char \*\*m){

++m;

cout<<\*m<<endl;

}

void main(){

static char \*a[]={"morning", "afternoon", "evening"};

char \*\*p;

p=a;

func(p);

}

A 字符o的起始地址

B 字符a的起始地址

C 字符o

D afernoon

249:

作用域运算符“：：”的功能是：（D）

A 标识作用域的级别的

B 指出作用域的范围的

C 给定作用域的大小的

D 标识成员是属于哪个类的

250:

一个const对象（B）。

A 能访问所有成员函数

B 只能访问const成员函数

C 也能访问volatile成员函数

D 能访问const和volatile成员函数

251:

设有以下说明语句

typedef struct

{ int n;

char ch[8];

} PER;(B)

A PER是结构体变量名

B PER是结构体类型名

C typedef struct是结构体类型

D struct是结构体类型名

252:

对于类定义

class A{

public:

virtual void func1( ){ }

void func2( ){ }

};

class B:public A{

public:

void func1( ){cout<<〃class B func 1〃<<end1;}

virtual void func2( ){cout<<〃class B func 2〃<<end1;}

};

下面正确的叙述是(C)

A A::func2( )和B::func1( )都是虚函数

B A::func2( )和B::func1( )都不是虚函数

C B::func1( )是虚函数，而A::func2( )不是虚函数

D B::func1( )不是虚函数，而A::func2( )是虚函数

253:

以下叙述中正确的是 (A)

A.C++语言程序的基本单位是类

B. 可以在一个函数中定义另一个函数。

C.main()函数必须放在其他函数之前

D.所有被调用的函数一定要在调用之前进行定义

254:

设有以下定义：enum t1 {a1,a2=7,a3,a4=15} time;则枚举常量a1和a3的值分别是(D)

A 1和2

B 6和3

C 1和8

D 0和8

255:

基类中构造函数说明为纯虚函数是没有意义的.(B)

A 错

B 对

256:

已知： union a { int x; short y; }; a var[20]; var占了多少字节？(A)

A 4

B 80

C 这是不合法的C++代码

D 120 E 不确定

257:

下列不具有访问权限属性的是（A）

A. 非类成员

B. 类成员

C. 数据成员

D. 函数成员

258:

在C++中使用流进行输入输出，其中用于屏幕输出的对象是：（C）

A cerr

B cin

C cout

D cfile

259:

执行语句k=30;j=k++;后k和j的值分别为（C）

A k=30,j=30

B k=30,j=31

C k=31,j=30

D k=31,j=31

260:

在C++语言中，函数返回值的类型是由（B）定义的。

A return 语句中的表达式类型

B 定义该函数时所指定的函数类型

C 调用该函数时的主调用函数类型

D 调用函数时系统临时

261:

下列关键字中，(C)不是类中定义数据成员使用的关键字

A double

B float

C extern

D static

262:

以下能正确定义字符串的语句是（D）。

A char str[]={′\064′};

B char str="\x43";

C char str=′′;

D char str[]="\0";

263:

设有以下定义和语句：  
char str[20]="Program", \*p;  
p=str;  
则以下叙述中正确的是（A）。  
A \*p与str[0]中的值相等  
B str与p类型完全相同  
C str数组长度和p所指向的字符串长度相等  
D 数组str中存放的内容和指针变量p中存放的内容相同

264:

关于虚函数的描述中，（A）是正确的。

A 派生类的虚函数与基类的虚函数具有不同的参数个数和类型

B 基类中说明了虚函数后，派生类中其对应的函数一定要说明为虚函数

C 虚函数是一个成员函数

D 虚函数是一个static类型的成员函数

265:

this指针是C++实现(B)的一种机制.

A 抽象

B 封装

C 继承

D 重载

266:

要禁止修改指针p本身，又要禁止修改p所指向的数据，这样的指针应定义为（D）

A const char \*p=“ABCD”；

B char \*const p=“ABCD”；

C char const \*p=“ABCD”；

D const char \* const p=“ABCD”；

