

## (一)

4. 下列关于C++关键字的说法中正确的是 (D)

- A) 关键字是用户为程序中各种需要命名的“元素”所起的名字。
- B) 关键字是对程序中的数据进行操作的一类单词。
- C) 关键字是在程序中起分割内容和界定范围作用的一类单词。
- D) **关键字**是C++中**预先定义**并实现一定功能的一类单词。

分析：关键字的概念，选D。

6. 下面关于对象概念的描述中 (A) 是错误的。

- A) 对象就是C语言中的结构变量 X
- B) 对象代表正在创建的系统中的**一个实体**
- C) 对象是一个状态和操作（或方法）的封装体
- D) 对象之间的信息传递是通过消息进行的

分析：A 对象是类的实例，不是结构体变量。D对象之间的消息传递通过消息进行？

8. 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 X

分析：同前选D错误。

11. 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** X
- D) C++包含C语言的全部语法特征

分析：选C。

17. 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) ;

分析：选C。注意复合语句结束标记是 } **语句块**

21. 下列不正确的选项是 (D)

- A) C++语言是一种面向对象的程序设计语言，它支持面向对象思想中的**3个主要特征** ✓
- B) 标点符号是在程序中起分割内容和界定范围作用的一类单词
- C) iostream是一个标准的头文件，定义了一些输入输出流对象
- D) 类与类之间不可以进行通信和联络

分析：注意A:主要三个特性是继承、封装、多态。抽象并非主要故A正确。

类和类之间显然可以通信，故D错误。选D

24. 下列正确的选项是 (A)

- A) 继承是创建一个具有别的类的属性和行为的新类的能力
- B) C语言支持**面向对象的程序设计** X

C)空白符是一种可以被编译的符号 ~~X~~

D)标识符不宜过长，一般设定为16个字符 ~~X~~ 无限制

分析：标识符就是我们定义的变量、常量、类对象名字等等。此题选A

29.下列引用的定义中 ( ) 是错误的。

A)int i; int &j=i;

B)int i; int &j; i=i; ~~X~~

C)float i; float &j=i;

D)char d; char &k=d;

分析：在C++中，常量对象和引用 不能被赋值，只能被初始化。故B错误，选B

34.对定义重载函数的下列要求中 ( ) 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 ~~X~~

分析：特别注意函数返回值不能区分函数，注意等价说法：函数的类型。故D

35.下面函数调用语句中实参的个数是 A

xxx((v1,v2),(v3,v4,v5),v6);

A)3

B)4

C)5

D)6

分析：此题易误选D。对于函数xxx来说，他只有3个实参。(v1,v2)返回值作为实参，只有1个，其余同理，故A

37.若要定义一个只允许本源文件中所有函数使用的全局变量，则该变量需要使用的存储类别是 D

A)extern

B)register

C)auto

D)static

分析：extern 是不同文件中使用，错误。选D

38.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 ~~X~~

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 ~~X~~

分析：此题易误选B。默认参数只能在函数 声明中 设定一次，函数实现 定义

变不可声明。如果无函数声明，才可再函数定义中声明。选C

41.在函数声明时，下列 ( ) 项是不必要的。

A)函数的类型

B)函数参数类型

C)函数的名字

D)返回值表达式

分析：D选项说的是 {} 里内容，这是可以省略的。选D

42.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 反了

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

47.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
```

分析: fun中返回的是 p与s的地址之差, 是一个int类型。另外注意: 传一个字符常量进去, 而函数要求参数是char\*

。此题选B

50. 下列 ( D ) 的调用方式是引用调用。

- A)形参是指针, 实参是地址值      B)形参和实参都是变量  
C)形参是数组, 实参是变量      D)形参是引用, 实参是变量

分析: A 这是传地址, C中如果变量是数组也是 传地址≠传引用。选D

53. 在传值调用中, 要求 ( C )

- A)形参和实参类型任意, 个数相等      B)实参和形参类型都 完全一致, 个数相等  
C)实参和形参对应的类型 一致, 个数相等      D)实参和形参对应的类型一致个数任意

分析: 并非要求完全一致, 参数类型是double, 也可传int进入。故选C

*隐性转换*

54. 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

分析: 定义为 寄存器变量可以加快存取。故A

56. 采用 函数重载 的目的在于 ( D )

- 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

分析: 函数重载就是为了使用方便, 故D。

58. 下面程序的输出结果是 ( D )

```
#include<iostream.h>
void fun(int i);
int main()
{
```

```

int n=2;
fun(n);
return 0;
}
void fun(int i)
{
    static int x=1;
    int y=1;
    if(i>0)
    {
        ++x; 2 3
        ++y; 2 2
        cout<<x<<' '<<y<<' ';
        fun(i- 1);
    }
}

```

静态变量,只初始化一次

A) 语法错误, 不能输出正确结果    B) 2 2 2 2    C) 2 2 3 3    D) 2 2 3 2

分析: 这题是递归调用, 上一层的static变量不会消亡或重新定义。故D

59.? 下面程序的输出结果为 (B)

```

#include<iostream.h>
void prt(int *x,int *y,int *z)
{
    cout<<+ + *x<<","<<+ + *y<<","<<*(z++)<<endl;/ / 11 21 40 12 22 41
}
int a=10;
int c=20;
int b=40;
void main()
{
    prt(&a,&b,&c);
    prt(&a,&b,&c);
}

```

↑  
后加.指向C.后加

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

64. 下面程序的执行结果是 (B)

```

#include<iostream.h>

```

```

void main()
{
    void swap();
    extern int a,b; // (1)
    a=3;
    b=10;
    swap(); // 那么问题来了，这个swap交换外面定义的ab影响得到这里的ab吗？
    cout<<"a="<<a<<" ,b="<<endl;
}
int a,b; //(2)
void swap()
{
    int temp;
    temp=a;
    a=b;
    b=temp;
}

```

A) a=3,b=10 B)a=10,b=3 C)a=temp,b=temp D)语法错误，不能输出正确结果

分析：实际输出的a和b已经交换了。因为extern关键字表明ab(1)是在别处声明的，后面给ab赋值实际相当于是给 (2)

处赋值，最后输出也是输出(2)处的ab值。故选B

68.在C++语言中，局部变量的隐含存储类型是 ( A )

A)auto B)static C)extern D)无存储类型

分析：应该是auto类型，选A。

1. ?有关类的说法不正确的是 BD

- 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成员。✗
- C) 在类中, 如果不做特别说明, 所指的数据均为私有类型。✓
- D) 在类中, 如果不做特别说明, 所指的成员函数均为公有类型。✗

分析: 此题应该BD均不正确。B友元函数也可以访问; D默认数据和成员函数都是private。

特别注意A: 类、结构体、枚举、联合都是自定义数据类型。

4. 有关析构函数的说法不正确的是 B

- A) 析构函数有且仅有一个 ✓
- B) 析构函数和构造函数一样可以有形参 ✗
- C) 析构函数的功能是用来释放一个对象
- D) 析构函数无任何函数类型 ✓

分析: 析构函数常考特点: 1.析构函数无形参 2.析构函数唯一 3.析构函数功能是释放对象

故, 此题选B。

5. ? 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]

分析: 答案选C, 但是C中缺少 int类型形参? void (X::\*pa) (int)

7. 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[]

分析: 此前有分析: B中, p加上()表示p和\*先结合, 则p为指针, 指向一个有4个元素的X对象数组。A中[]优先级更高和

p结合, 表示p是数组, 有4个X\*变量。D中必须指定大小。故选B。

9. 已知f1(int)是类X的公有成员函数, p是指向成员f1()的指针, 采用它赋值 C 是正确的.

- A) p=f1 B) p=&X::f1 C) p=X::f1 D) p=f1()

分析: 此题易错选B。首先要使用函数要加(), 排除D; 按题意要限定类名, 排除A; B为注意, f1本身就是一个地址 (函数地

址入口), 故直接C形式即可, 选C。

13. 下列静态数据成员的特性中, D 是错误的.

- A) 说明静态数据成员时前边要加关键字static来修饰
- B) 静态数据成员在类体外进行初始化
- C) 引用静态数据成员时, 要在静态数据成员名前加<类名>和作用域运算符
- D) 静态数据成员不是所有对象所共有的

## 也不能用构造函数初始化静态成员

分析: B: static成员只能在类外初始化, 但static

const

(≠ static) 可在类内; C: 也可以对象访问, 但C表述无误。D明显错误, 是所有对象共有的, 故选D。

14. 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 × 概念来说还是选a比较好

分析: 易错选B。template <class T> class A{...}: 类模板在使用前一定要先实例化, 把类模板类型参数确定了,

类模板变成了一个具体的类, 再由类产生一个对象。故此题选A。

16. 关于常数据成员的说法不正确的是 D。

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

static const 可以在定义时初始化

分析: 常数据成员在类中定义时不能被直接赋值! 除非是静态数据成员! 故选D。

19. 下列 D 不是构造函数的特征

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) 构造函数必须指定类型说明 ×

分析: 注意构造函数并没有返回类型, 故选D。

22. 关于下面程序段说法正确的是\_\_\_\_\_。

```
class X
{
private: int n;
public: 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&x)也可以不使用引用参数而仍然是拷贝构造函数 ×

D) X(X&x)不能访问对象的私有成员 ×

分析: A: 显然正确。C: 拷贝构造函数参数一定要是引用

, 防止拷贝构造函数无限递归! D: X是成员函数, 它的参数[本类]对象引用可以访问[本类]的私有成员。

23. 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



分析: this指针是C++实现的一种封装机制!!

## 26. 类模板template<class

T>classX{.....};其中, 友元函数f()成为该类模板实例化的每个模板类的友元, 则其说明应为\_\_\_\_\_.

- A) friend void f(); 用于全局对象的访问
- B) friend void f(X<T>&);
- C) friend void A::f();
- D) friend void C<T>::f(X<T>&);

分析: 不要犹豫, 就是选A!

27. 运算符—>\*的功能是 \_\_\_\_\_

- A) 用来表示指向对象指针对指向类成员指针的操作。
- B) 用来表示对象类成员的操作。
- C) 用来表示指向对象指针对类成员的操作。
- D) 用来表示对象指向类成员指针的操作。

分析: 选A, 有点绕口, 但实际就是这么一回事儿。

28. 下列说明中

const char \* ptr;

ptr 应该是 C

- A) 指向字符的指针      B) 指向字符的常量指针
- C) 指向字符串常量的指针      D) 指向字符串的常量指针

分析: 选C。右→左结合: \*ptr表示ptr是一个指针; char \*ptr表示ptr指向char类型的指针; const char\* ptr

则表示指向字符串常量的指针。

同理: char \* const ptr; 表示ptr是一个常量指针指向字符串 (等同于D表达)

29. 下面关于友元的描述中, 错误的是 D

- A) 友元函数可以访问该类的私有数据成员。✓
- B) 一个类的友元类中的成员函数都是这个类的友元函数。✓
- C) 友元可以提高程序的运行效率。
- D) 类与类之间的友元关系可以继承。✗

分析: 易知选D。注意B, 友元类所有成员函数都是该类的友元函数!

## 30. 一个 \_\_\_\_\_

允许为类定义一种模式, 使得类中的某些数据成员、某些成员函数的参数和返回值可以取任意数据类型。

- A) 函数模板      B) 模板函数
- C) 类模板      D) 模板类

分析: 注意区分类模板和模板类。类模板实例化之后就是模板类。此题选C

31. 有关构造函数说法不正确的是 D

- A) 构造函数的名字和类的名字一样。
- B) 在创建对象时，系统自动调用构造函数
- C) 构造函数无任何 函数类型
- D) 构造函数有且只有一个 X

分析:D错误, 选D。注意C函数类型的说法, 就是返回类型的意思。

34. 下列关于构造函数的描述中,错误的是 C

- A) 构造函数可以设置默认参数 ✓
- B) 构造函数在说明类变量时自动执行 ✓
- C) 构造函数可以对静态数据成员进行初始化 X
- D) 构造函数可以重载 ✓

分析: 静态数据成员, 只能在类外类初始化! static const 可以在类内。

35. 对于类 A, 语句 `void (A::*P)(void);` 表明 A

- A) P是一个指向类成员函数的指针
- B) P是类A的一个成员
- C) P是类A的一个对象
- D) P是一个指向类对象的指针

分析: 此题易认为只是一个函数指针, 它还特地指向了是 类A成员函数的指针, 故A。

36. 运算符 `delete` 删除一个动态对象时 C

- A) 首先为该动态对象调用构造函数,再释放其占用的内存
- B) 首先释放该动态对象占用的内存,再为其调用构造函数
- C) 首先为该动态对象调用析构函数,再释放其占用的内存
- D) 首先释放动态对象占用的内存,再为其调用析构函数

分析: 考察delete的用法, 先析构 ---> 释放内存。故选C

37. 在类的定义形式中, 数据成员、成员函数和 A 组成了 类定义体。

- A) 成员的访问控制信息
- B) 公有消息
- C) 私有消息
- D) 保护消息

分析: 成员控制信息+数据成员+函数成员 == 类定义体, 故A。

41. ( A ) 只能访问静态成员变量, 静态成员函数和类以外的函数和数据不能访问类中的非静态成员变量。  
(类外函数可以访问类中静态?类名限定?)

- A) 静态函数
- B) 虚函数
- C) 构造函数
- D) 析构函数

分析: 静态函数只能访问静态成员。此题选A

42. 下面说法正确的是 B

- A) 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 X
- B) 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
- C) 类的内联函数 必须在类体内定义 (定义不是声明) X
- D) 类的内联函数 必须在类体外关键字inline定义 X

分析：在类中内联函数定义：1.隐式内联：在类中直接定义函数具体实现，编译器会自动认为是内联。2.显式内联：

在1.类中声明原型2.类外具体实现。在1-2任意前面声明一个inline即可，也可都声明。故CD错误。

内联函数是在编译时将函数代码插到调用地方，而不用调用函数，减少开销，故B。

43. 下列关于成员访问权限的描述中，不正确的是 ( B )

- A) 公有数据成员和公有成员函数都可以被类对象直接处理
- B) 类的私有数据成员只能被公有成员函数处理。
- C) 保护成员在派生类中可以被访问，而私有成员不可以 V
- D) 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

分析：易知B错误。友元函数、类中私有成员函数等等都可以访问私有数据成员。而且！B中说的公有成员函数只能是

本类的，派生类公有成员函数也无法访问！

44. 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

分析：这题说的是只是类内吗？如果 `int a; int main() { int a; }` 如何访问全局变量a呢？

其实很简单直接就是： `::a = 5;`，代表访问全局变量a赋值5。此题选B。

45. 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

分析：缺省析构函数 == 不自己定义析构函数，是默认析构。默认析构函数体为空，选C。

48. 关于delete运算符的下列描述中， ( C ) 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 X 只能用一次
- D) 指针名前只用一对方括号符，不管所删除数组的维数 V

分析：1.delete必须用于new分配空间返回指针 2.空指针也可（指没有赋值）3.无论多维都是delete[] A

就完事儿。C明显错误，故选C。

49. 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 X
- B) 成员函数可以重载 V
- C) 成员函数可以设置参数的默认值 V
- D) 成员函数可以是静态的 V

分析：成员函数在类中实现了才是内联。否则没有关键字inline不是内联。故A

51. 由于数据隐藏的需要，静态数据成员通常被说明为 ( A )

- A) 私有的 B) 公有的 C) 保护的 D) 不可访问的

分析：此题易错选B，静态数据成员通常声明是私有。

53? . 关于new运算符的下列描述中，( D ) 是错误的。

- A) 它可以用来动态创建对象和对象数组 ✓
- B) 使用它创建对象或对象数组,可以使用运算符DELETE删除 ✓
- C) 使用它创建对象时要调用构造函数 ✓
- D) 使用它调用对象数组时不允许指定初始值

分析: B正确。Cnew指针本质就是调用构造函数初始化。D: 如A \*arr = new A[5];此时没有指定初始值。又如

`int* arr = new int[5]();` 此时指定了初始值0, 故D。

54. 对于常成员函数, 下列描述正确的是 ( )

- A) 常成员函数只能修改常数据成员 X
- B) 常成员函数只能修改一般数据成员 X
- C) 常成员函数不能修改任何数据成员 ✓
- D) 成员函数只能通过常对象调用 X

分析: 此题易误选A。常成员函数: 1. 只能调用常成员函数, 常数据成员 2. 不能修改任何成员包括常成员。故此题选C

55. 友元的作用是 ( )

- A) 提高程序是运行效率 A
- B) 加强类的封装性 X
- D) 实现数据的隐藏性 X
- D) 增加成员函数的种类 X

分析: 选A。友元函数是破坏封装和隐藏故BC错误, D乱入。

57. 下面说法中正确的是 ( )

- A) 当声明的内容不同时, 声明同一个名字的两个类是允许的
- B) 不完全声明的类也可以实例化 X
- C) 声明同一个名字的两个类时, 要求声明的内容也相同
- D) 声明同一个名字的两个类是错误的 ✓

分析: 注意抽象类不能被实例化, 只能被继承! 且类无法重载, 故D正确。

59. 下列关于对象数组的描述中, 错误的是 ( )

- A) 对象数组的下标是从0开始的 ✓
- B) 对象数组的数组名是一个常量指针
- C) 对象数组的每个元素是同一个类的对象
- D) 对象数组只能赋初值, 而不能在定义后赋值 X

分析: D显然错误。 `int* A = new int[5] { 1, 2 ... } / int A[5]` 初始化后也可以赋值的。故D

61. 已知: `print()` 函数是一个类的常成员函数, 它无返回值, 下列表示中, 正确的是 A

- A) `void print() const`
- B) `const void print()`
- C) `void const print()`
- D) `void print(const)`

分析: 常成员函数的定义选A。

62. 下面是类MyClass的定义, 对定义中各语句描述正确的是 ( )

```

class MyClass
{
private:
    int X,y,Z;
public;
void MyClass(int a){X=a;}      file:// ①
int f(int a,int b)              file:// ②
{
X=a;
Y=b;}
int f(int a,int b,int c=0)      file:// ③
{X=a;
Y=b;
Z=c;}
Static void g() {X=10;}         file:// ④

```

- A) 语句①是类MyClass的构造函数定义 **X**
- B) 语句②和语句③实现类成员函数的重载
- C) 语句④实现对类成员变量X的更新操作 **X**
- D) 语句①②③和④都不正确

分析：此题interesting! A : 构造函数没有返回值

**二义性  
重载混乱**

BC注意大小写错误

D

静态成员函数不能访问非静态成员(非静态成员都没有分配内存)。故此题选D。

63. ?友元访问类的对象的成员时使用 ( **D** )

- A) 类的成员名                      B) this指针指向成员名
- C) “类名： 成员名”的形式      D) “对象名.成员名”的形式

分析：题意描述不清楚。答案选D。

66. 静态成员遵循类的其他成员所遵循的访问限制，除了 **B**

- A) 静态成员函数                      B) 静态数据成员初始化
- C) 私有静态数据成员                  D) 公有静态成员函数

分析：AD说的是静态成员函数。B正确静态初始化在类外，故选B。

69. 关于 类中数据成员的生存期的说法正确的是 ( **D** )

- A) 与对象的生存期无关              B) 比对象的生存期长
- C) 比对象的生存期短                  D) 由对象的生存期决定

分析：选D。类中数据成员，类对象消亡也会一起消亡，否则存在。

70. ?类example的析构函数为 ( **C** )

A) example ();    B) example ()    C) ~example ();    D) ~example ()

分析：这题选D??不带分号?

72.对于类模板anytemp,执行语句anytemp<int>iobject(23,32);后( C )

- A) 产生的模板类为 anytemp
- B) 产生的模板类为 iobject
- C) 产生的模板类为 anytemp<int>
- D) 类模板实例化对象

分析：产生类模板anytemp-->产生模板类anytemp<int>-->产生对象iobject, 故C。

76. 有类的作用域，下面说法中不正确的是( B )

- A) 说明类时所使用的一对花括号形成所谓的类作用域 ✓
- B) 类作用域不包含类汇总成员函数的作用域 ✗
- C) 类作用域中说明的标识符只在类中可见 ✓
- D) 在可能出现两义性的情况下,必须使用作用域限定符"::" ✓

分析：类作用域即指类所使用{ }形成，里面成员（声明的变量/函数名 ==

标识符）只在类中是可见的，而在类外不可见，如果发生屏蔽现象要使用:: 访问。故A、C正确。B错误，类作用

域{ }肯定包含类成员函数作用域{ }，故选B。

80.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); // 1
    example arr[2]={1,2}; // 2
    x=10;
    x=example(15); // 3
}
```

- A) 3            B) 4            C) 7            D) 8

分析：1、2构造函数初始化一共有1+2 = 3；3处复制构造 创临时对象1个，故4个选B。

✗ 赋值运算符重载  
(三)

2. 下面关于纯虚函数和抽象类的描述中, B是错误的。

- A) 纯虚函数是一种特殊的虚函数, 它没有具体意义
- B) 一个基类中说明有纯虚函数, 该基类的派生类一定不再是抽象类 ✗ 如果没实现, 就是
- C) 抽象类只能作为基类来使用, 其纯虚函数的定义由派生类给出

D) 抽象类是指具有纯虚函数的类 ✓

分析：纯虚函数：1.没有实现部分，其定义由派生给出。

派生类继承后不实现纯虚函数就是抽象类

2.含纯虚函数的基类一定是抽象类，但派生类只有实现纯虚才是抽象类。且抽象类是无法实例

化的。故选B.

4. ? 对虚函数的调用\_\_\_\_\_。

- A) 一定使用动态联编      B) 一定使用静态联编  
C) 必须使用动态联编      D) 不一定使用动态联编

分析：此题选D。虚函数有动态和静态连编两种：虚函数的动态调用，是指动态联编用指针或引用确定绑定；静态连编如

直接对象.虚函数名，则静态。故选D

5. 若类A和类B的定义如下：

```
class A
{
    int i,j;
    public:
    void get();
};

class B:A
{
    int k;
    public:
    void make();
};

void B:: make()
{
    k=i*j;
}
```

公有派生. 公有成员与保护成员不变. 私有成员可直接访问  
私有派生. 公有成员与保护成员变为私有. 私有成员不可直访  
保护派生. 公有成员与保护成员变为保护. 私有成员不可直访

则上述定义中，D是非法的表达式。

私有继承私有成员在派生类中不能访问

- A) void get();    B) int k;    C) void make();    D) k=i\*j;

分析：加粗部分继承方式未指定是private继承，D访问i和j是非法的。故选D

7. 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C。

- 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

分析：背名词解释，选C。

8. A是一个在基类中说明虚函数，它在该基类中没有定义，但要求任何派生类中的可访问性的规定是相同的。

- A) 纯虚函数    B) 虚析构函数    C) 虚构造函数    D) 静态成员函数

分析：题中要求虚函数D首先排除。C：不会有虚构造，构造之前对象都不存在，如何绑定。

B:虚析构函数可以有定义。故此题A是最佳也是附和题意选项，选A。

11. 派生类的对象对它的基类成员中A是可以访问的。

- A) 公有继承的公有成员    B) 公有继承的私有成员

- C) 公有继承的保护成员 ?? D) 私有继承的公有成员

分析：建议背熟那三个名词解释。公有继承下，基类的成员访问属性不会被改变；派生类中只能访问基类的公有成员（

其他继承还可访问保护成员）；派生类对象（其他继承也是：派生对象只能访问基类（继承后）

和派生类成员公有）只能访问基类和派生类的公有成员。故C不正确，基类protect在public继承也是protect

无法访问，故选A。

12. 关于函数的描述中，C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 X

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 X

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 X

分析：选C。注意基类的虚函数，派生类对应函数没有必要声明为虚函数，B错。

13. 对基类和派生类的关系描述中，B是错误的。

A) 派生类是基类的具体化

B) 派生类是基类的子集 反了

C) 派生类是基类定义的延续

D) 派生类是基类的组合

分析：A：基类一般是若干个派生类的抽象，派生类是其具体化，描述正确。B派生类不是基类的子集，派生类还有基类没

有的部分（区分：在公有继承下派生类是基类的子类型

）。C描述正确。D派生类通常还有由继承多个基类，此表述无误。故此题选B

14. 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(不是这么调用就是非动态)

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 X

分析：此题选D,应该是运行时确定的。

15. ? 派生类的构造函数的成员初始化列表中不能包含C。

A) 基类的构造函数 ✓ B) 派生类中子对象的初始化 ✓

C) 基类的子对象初始化 X D) 派生类中一般数据成员的初始化 ✓

分析：基类的子对象会在基类的构造函数中进行。派生类会为基类构造提供参数，故C。

派生类初始化列表：1.基类成员，给基类构造提供参数2.派生类成员3.派生类对象。

17. 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

分析：选B。二义性分为：1.同名二义性，发生在基类之间和基类和派生类之间

。2.路径二义性：祖宗类p有成员函数say，f1 f2继承p，s同时继承f1 f2，s.say不知道调用哪个fsay。

18. ? 在下面程序中，编译时出现的错误是C。

组合对象



```

class A      (1)
{
    Public:   (2)
    A()
    {
        func(); (3)
    }
    virtual void func()=0; (4)
};
A) (1)   B) (2)   C) (3)   D) (4)

```

分析：纯虚函数不能被调用??? 构造函数中不能调用，成员函数可以！选C

19. 关于继承二义性的描述中，错误的是C。

- A) 一个派生类的基类中都有某个同名成员，在派生类中对这个成员的访问可能出现二义性
- B) 解决二义性的最常用的方法是对成员名的限定法 ✓
- C) 基类和派生类同时出现的同名函数，也存在二义性问题 ✗
- D) 一个派生类是从两个基类派生出来的，而这两个基类又有一个共同的基类，对该基类成员进行访问时，可能出现二义性

分析：选C。前面又说虚基类是为了消除二义性？对。当基类和派生类出现同名函数时，有virtual实现重写，没有二义性

；当无virtual时出现覆盖基类，无二义性。

21. 下列对派生类的描述中，错误的是 ( D )
- A. 派生类的缺省继承方式是PRIVATE ✓
  - B. 派生类至少有一个基类 ✓
  - C. 一个派生类可以作为另一个派生类的基类 ✓
  - D. 派生类只继承了基类的公有成员和保护成员

分析：D错误。派生类继承了私有成员只是无法访问

22. 编译时的多态性通常使用 ( C ) 获得。

- A. 继承
- B. 虚函数
- C. 重载函数
- D. 析构函数

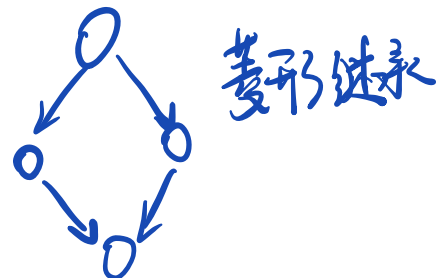
分析：注意看清题目，说的是编译时故选C。

25. 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

分析：虚基类使一个基类被多次继承只有一份保存，且只需继承时声明，故D。

27. 带有虚基类的多层派生类构造函数的成员初始化列表中都要列出虚基类构造函数，这样将对虚基类的子对象初始化 (

- D
- A. 与虚基类下面的派生类个数有关
  - B. 多次
  - C. 二次
  - D. 一次



分析：虽然都要列出虚基类构造函数，但是虚基类对象只会被初始化一次。

29? . 下列关于子类型的描述中，错误的是

- A. 子类型关系是可逆的
- B. 公有派生类的对象可以初始化基类的引用
- C. 只有在公有继承下，派生类是基类的子类型
- D. 子类型关系是可传递的

分析：基类派生类不可逆。B正确，多态性。C：公有继承下，派生类对象可以作为基类对象处理，此时是基类子类型

，正确，记住。此题选A

30. 若要强制C++对虚函数的调用使用静态联编，则在调用中对该函数使用

- A. 成员名限定 B. 指针 C. 引用 D. VIRTUAL关键字

分析：选A。引用特定对象调用虚函数就是静态连编。

31? . 模板和继承的关系是 ( )

- A. 模板反映的是不同类型的相同操作；而同一类系各类处理数据很多情况下是同一类型的
- B. 继承类较模板类有更多的动态特征
- C. 模板类系的成员较继承类系的成员稳定。
- D. 模板和继承的方法都可以派生出一个关系，因而本质上它们是相同的。

分析：选A。B: 不该更动态吗？C: 模板不稳定吗？继承还可能改变访问方式。

32. 下面 ( ) 的叙述不符合赋值兼容规则。

- A. 派生类的对象可以赋值给基类的对象
- B. 基类的对象可以赋值给派生类的对象
- C. 派生类的对象可以初始化基类的对象
- D. 派生类对象的地址可以赋值给指向基类的指针

分析：选B。

34. 对于下列程序段，没有二义性的表达式是 ( )

```
class A{public:int f();};
class B{public:int g();int f();};
class C:public A,public B{ public : int g( ) ; h( ) ; };
```

C obj;

- A. obj.f( )
- B. obj.A::g( )
- C. obj.B::f( )
- D. obj.B::h( )

分析：A: 不确定访问AB哪个f(); B、D: 访问错误。C: 正确。故选C。

36? . 派生类的对象与基类的对象的关系是 ( )

- A. 属于前者而不属于后者
- B. 属于前者，不一定属于后者
- C. 不属于前者则一定属于后者
- D. 属于前者则一定属于后者

分析：此题什么意思？答案选D。

37. 对于下面程序段，表达式正确的是 ( )

```

class Base {public:int b;};
class Base1:public base{ };
class Base2:public base{ };
class Derived:public base1,public base2{public:int fun ( );};
Derived d;

```

没有直接继承 Base

- A. d. Base1::b ✓  
 B. d. Base ::b ✗ = 歧义  
 C. d.b ✗ = 歧义 要用虚基类  
 D. d. Base :: fun() ✗

分析：C显然错误：不知道是访问base1还是base2的b；D错误，base没有fun；此题易误选B，派生类没有继承Base！故

选A

应该是二义性问题

40. 不能说明为虚函数的是 ( B )

- A.析构函数 B.构造函数 C.类的成员函数 D.以上都不对

分析：虚函数没虚构造和静态 友元、普通函数，可以被声明为：虚析构、成员函数故选B。

41.不同的对象可以调用相同名称的函数，并可导致完全不同的行为的现象称为 ( A )

- A.多态性 B.抽象 C.继承 D.封装

分析：真题填空。选A

43. 关于虚函数，下列表述正确的是 ( AB C )

A. ?如果在重定义虚函数时使用virtual，则该重定义函数仍然是虚函数

B.虚函数不得声明为静态函数

C.虚函数不得声明为另一个类的友元函数 ✓

D. 派生类必须重新定义基类的虚函数 ✗

分析：同40易知BC错误。D：派生类不必重新定义虚函数，哪怕纯虚也不一定。A：

44. 下列程序段中具有相同值的是 ( A )

```

class Base {int b;};
class Base1:virtual public base{int b1;};
class Base2:virtual public base{int b2;};
class derived:public base 1,public base 2{int b3;};
derived d;

```

- A. d.b与d.base1::b B. d.base1::b与 d.base1::b1  
 C. d.b与 d. b3 D. d.base2::b与 d.base2::b2

分析：Base被虚继承，b只有一份，故A正确，选A不会有二义性？

虚基类不会有

46. 关于纯虚函数，下列表述正确的是 ( B )

A. 纯虚函数是未给出实现版本（即无函数体定义）的虚函数

B. 纯虚函数的声明总是以“=0”结束 ✓

C. 派生类必须实现基类的纯虚函数

D. 含有纯虚函数的类一定是派生类

分析：易错选A。虚函数根本没有函数体{}，故A错误，选B。

50. 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

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

分析：答案选A？可是公有继承，保护成员也不能被访问啊？应该选D？

54. 对于类定义

class A{

public:

virtual void func1(){}  
T

void func2(){}  
F

};

class B:public A{

PUBLIC:

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

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

};

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，A::func2()不是虚函数

D. B::func1()不是虚函数，A::func2()是虚函数

分析：易错选B。基类声明是虚函数，派生类中则也是虚函数且可不写virtual (反之不成立)

56. 下列程序段中，错误的语句是 ( C )

class A{

public: virtual void f()=0; (1)

void g(){f();} (2)

A(){f();} (3)

}

A. (1) B. (2) C. (3) D. (1) 和 (2)

分析：纯虚函数构造函数中不能被调用！成员函数可以，故选C。

57. 下面程序段中，说法正确的是 ( d )

class location{

```

public :
int GetX();
};
class rectangle:private location {
public:
int GetW();
};

```

A.类rectangle的私有派生类是location

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

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

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

分析：又错选C。私有继承，派生类中还是可以访问基类公有成员和保护成员；但对象只能访问公有。故选D

59.可以用p.a的形式访问派生类对象p 的基类成员a其中a是

A.私有继承的公有成员

B.公有继承的私有成员

C.公有继承的保护成员

D.公有继承的公有成员

分析：同前，对象能访问基类成员，则定是公有。只有public继承才能访问基类公有故D。

## (四)

1. 下列运算符中，A运算符在C++中不能重载。

A) ? : B) + C) - D) <=

分析：三目不能重载。选A

2. 在C++中，打开一个文件就是将这个文件与一个

myfile ; myfile.open(" filename" );

A) 流 B) 类 C) 结构 D) 对象

分析：A

3. 下列关于运算符重载的描述中，D是正确的。

A) 运算符重载可以改变操作数的个数 X B) 运算符重载可以改变优先级 X

- C) 运算符重载可以改变结合性 X      D) 运算符重载不可以改变语法结构

分析:运算符重载: 结合性、优先级、操作符目数、语法结构都是不能改变的。故D

4. 若定义 cin>>str; 当输入

Microsoft Visual Studio 6.0!

所得的结果是str= B

- A) Microsoft Visual Studio 6.0!    B) Microsoft  
C) Microsoft Visual                  D) Microsoft Visual Studio 6.0

分析: 选B。会以空白为分隔符

5. 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)

分析:易错选D。题目说了是友元, 故A。 成员函数

6. 进行文件操作时需要包含 B 头文件。

- A) iostream    B) fstream    C) stdio    D) stdlib

分析: B。

7. 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 A。

- A) ios::in    B) ios::out    C) ios::trunk    D) ios::binary

分析: A。

8. 使用如 setw() 的操作符对数据进行格式输出时, 应包含 D 文件。

- A) iostream    B) fstream    C) stdio    D) iomanip

分析: 格式输出要包含iomanip(控制), 故D。

manipulate