

# C++期末考试选择题题库100道&&C++期末判断题的易错知识点复习

今天备考C++，看到了一些好的复习资料，整合一起给大家分享一下

## 选择题

1. 对于常数据成员，下面描述正确的是【 B 】  
A. 常数据成员必须被初始化，并且不能被修改  
B. 常数据成员可以不初始化，并且不能被修改  
C. 常数据成员可以不初始化，并且可以被修改  
D. 常数据成员必须被初始化，并且可以被修改
2. 在C++中，使用流进行输入输出，其中用于定义文件输入流对象的类是【 B 】  
A. ios B. ifstream C. ofstream D. istream
3. 下列关于异常的叙述错误的是【 D 】  
A. 只要是编程者认为是异常的都可当异常抛出  
B. 硬件故障也可当异常抛出  
C. 运行错属于异常  
D. 编译错属于异常，可以抛出
4. 在面向对象的程序设计中，基类可以为其派生类提供一组公共的行为，而各个派生类可以为这组公共行为提供不同的实现，从而使得派生类的对象在执行这些公共行为时会有不同的表现，这种特性称为面向对象的【 D 】  
A. 封装性 B. 消息性 C. 继承性 D. 多态性
5. 关于域的描述中，错误的是【 C 】  
A. 一个类是一个独立的域  
B. 派生类域被自动嵌套在基类类域中  
C. 域操作符::可以被重载  
D. 不同域中可以定义同名变量和函数
6. 下列对变量的引用中错误的是【 C 】  
A. int a; int &p = a;  
B. char a; char \*p = &a;  
C. int a; int &p; p = a;  
D. float a; float p=a;
7. 下面【 A 】的叙述不符合赋值兼容规则。  
A. 基类的对象可以赋值派生类的对象  
B. 派生类指针可以指向基类对象  
C. 派生类的对象可以初始化基类的对象  
D. 派生类的对象的地址可以赋值给指向基类的指针
8. 构造函数不具备的特征的是【 D 】  
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. 构造函数必须指定返回类型

9. 关于虚函数的描述中，正确的是【 D 】

- A. 虚函数 是一个static类型的成员函数
- B. 虚函数是一个友元函数
- C. 基类中说明了虚函数后，派生类中与其对应的函数必须再次说明为虚函数
- D. 派生类的虚函数与基类的虚函数具有相同的参数个数和类型

10. 下列关于静态数据成员的说法，不正确的是【 C 】

- A. 类中定义的公用静态数据成员，可以通过类的对象来访问
- B. 类中定义的所有静态数据成员，都必须在类外初始化
- C. 静态数据成员不是所有对象所共用的
- D. 普通的成员函数可以直接访问类中的静态数据成员

11. 下列关于运算符重载的描述中，正确的是【 D 】

- A. 运算符重载可以改变操作数的个数
- B. 运算符重载可以改变优先级
- C. 运算符重载可以改变结合性
- D. 运算符重载不可以改变语法结构

12. 有关内联函数的描述，正确的是【 A 】

- A. 内联函数可以提高程序效率
- B. 内联函数的调用过程与普通函数一样
- C. 只要用 inline 关键字声明的函数都是内联函数
- D. 全局函数不能声明为内联函数

13. 下列操作符中不能被重载的是【 D 】

- A. ->
- B. \*
- C. %
- D. ?:

14. 如果一个函数的声明是int fun(char\*)，则下列函数声明中，错误地重载了该函数的是【 B 】

- A. int fun (char\*, int)
- B. void fun (char\*)
- C. int fun (int)
- D. void fun (char\*, int)

15. 下面是关于派生类构造函数的执行顺序的叙述：①执行派生类的构造函数函数体中的内容。②按照基类被继承时声明的顺序（从左向右）执行基类的构造函数。③按照内嵌成员对象在类中声明的顺序执行它们的构造函数。上述执行步骤的正确排序是【 D 】

- A. ①②③ B. ②①③ C. ③②① D. ②③①

16. 设A是一个有不带参数的构造函数的类，fun是一个如下定义的函数：Afun(A&a){return a;}则在执行程序段{Aa; fun(a); }时，对A类的：①构造函数，②析构函数，③拷贝构造函数的调用顺序是【 B 】

- A. ①③③②②② B. ①③②②
- C. ①②③② D. ①②③③②②

17. 在公有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 】

- 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

18. 所谓数据封装就是将一组数据和与这组数据有关操作组装在一起，形成一个实体，这实体也就是【 C 】

A. 函数体 B. 对象 C. 类 D. 数据块

19. 关于运算符重载，下列表述中正确的是【 B 】

A. 可以通过运算符重载来创建C++中原来没有的运算符

B. 在类型转换函数的定义中不需要声明返回类型

C. 运算符函数的返回类型不能声明为基本数据类型

D. C++已有的任何运算符都可以重载

20. 下列关于静态数据成员的特性的叙述中，错误的是【 C 】

A. 说明静态数据成员时前边要加修饰符static

B. 静态数据成员要在类体外进行初始化

C. 静态数据成员不是所有对象所共有的

D. 初始化静态数据成员时，要在其名称前加类名和作用域运算符

21. 为了使类中的某个成员不能被类的对象通过成员操作符访问，则不能把该成员的访问权限定义为【 AD 】。

A.public B.protected C.private D.static

22. 在用关键字class定义的类中，以下叙述正确的是【 A 】

A. 在类中，不作特别说明的数据成员均为私有类型

B. 在类中，不作特别说明的数据成员均为公有类型

C. 类成员的定义必须是成员变量定义在前，成员函数定义在后

D. 类的成员定义必须放在类定义体内部

24. 下列情况中，不会调用复制构造函数的是【 C 】

A. 函数的返回值是类的对象，函数执行返回调用时

B. 函数的形参是类的对象，调用函数进行形参和实参相结合

C. 将类的一个对象赋值给该类的另一个对象时

D. 用一个对象去初始化同一类的另一个新对象时

25. 下列关于抽象类描述不正确的是【 D 】

A. 抽象类不能定义对象

B. 其派生类没有定义纯虚函数，则仍是抽象类。”

C. 其派生类定义所有纯虚函数，则是具体类

D. 不能定义指向抽象类的指针变量

26. 下列关于常成员和常对象的叙述中，正确的是【 A 】

A. 通过常对象只能调用其常成员函数

B. 常成员函数只能由常对象调用

C. 通过常成员函数可以修改常成员数据的值

D. 常成员函数可以通过类名来调用

27. 关于成员函数特征的下列叙述中，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. 成员函数可以设置缺省参数值

D. 成员函数可以是静态的

28. 下面关于友元函数和友元类的叙述中，错误的是【 C 】

A. 如果在类A中把类B的某个函数声明为自己的友元函数，那么在类A的定义之前一定要有类B的完整定义

B. 如果在类A中把类B声明为自己的友元类，那么在类A的定义之前可以仅有类B的前向声明，而不必有类B的完整定义

C. 如果将类B声明为类A的友元，那么类A也就成为类B的友元

D. 如果将类B声明为类A的友元，那么类B中的函数可以访问类A的全部成员

29. 程序中重载了C++流运算符<<，它是【 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. 用于输入操作的非成员函数

D. 用于输出操作的非成员函数

30. 下列字符常量的写法中错误的是【 C 】

A. '\105' B. '\*' C. "" D. '\n'

31. 下列关于静态数据成员的描述中，正确的是【 A 】

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

D. 静态数据成员不能通过类的对象调用

32. 设A是一个类，且语句“A\*p=newA[10];”是能够正确执行的，则下列关于该语句的执行的叙述最恰当的是【 D 】

A. 类A有不带参数的构造函数，且构造函数被调用10次

B. 类A有拷贝构造函数，且拷贝构造函数被调用10次

C. 类A有这样的构造函数，其每个参数都有默认的形参值，且该构造函数被调用10次

D. A和C皆有可能。

33. 如果派生类中有指针类型的数据成员，该数据成员指向的内存空间需要动态申请，并且需要在构造函数中初始化该数据成员，在析构函数中回收该成员可能指向的内存空间。此时，如果存在通过基类的指针删除派生类的对象的情况，则需要【 B 】

A. 将基类的构造函数声明为虚函数

B. 将基类的析构函数声明为虚函数

C. 将派生类的构造函数声明为虚函数

D. 将派生类的析构函数声明为纯虚函数

34. 在类 object 中，以下定义会出现编译错误的是【 C 】

A. static object a;

B. object a;

C. static object \*pa;

D. object\* pa;

35. 运算符">"被友元重载，则表达式"obj1>obj2"被编译器解释为【 A 】

A. `operator >` (obj1,obj2)

B. > (obj1,obj2)

C. obj2.operator > (obj1)

D. obj1.operator > (obj2)

36. 关于友元，下列说法错误的是【 A 】

- A. 如果类A是类B的友元，则类B也是类A的友元
- B. 如果函数fun()被说明为类A的友元，那么在fun()中可以访问类A的私有成员
- C. 友元关系不能被继承
- D. 如果类A是类B的友元，那么类A的所有成员函数都是类B的友元

37. 面向对象系统具有【 A 】、继承性和多态性三种基本特性。

- A. 封装性
- B. 消息性
- C. 维护性
- D. 动态性

38. 从基类 public 派生一个类，基类 protected 成员成为派生类【 B 】成员。

- A. public
- B. protected
- C. private
- D. virtual

39. 下列关于虚函数的描述中，正确的是【 C 】

- A. 虚函数可以是一个static类型的成员函数
- B. 虚函数可以是非成员函数
- C. 基类中说明了虚函数后，派生类中其对应的函数可以不必说明为虚函数
- D. 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

40. this指针存在的目的是【 C 】

- A. 保证基类共有成员在子类中可以被访问
- B. 保证基类保护成员在子类中可以被访问
- C. 保证每个对象拥有自己的数据成员，但共享处理这些数据成员的代码
- D. 保证基类私有成员在子类中可以被访问

41. 以下成员函数表示纯虚函数的是【 C 】

- A. virtual int vf (int);
- B. void vf (int) = 0;
- C. virtual void vf () = 0;
- D. virtual void vf (int) {}

42. 对于一组功能相同但处理数据类型不同的函数可以通过【 B 】抽象为同一段代码。

- A. 类 模板
- B. 函数模板
- C. 函数重载
- D. 虚函数

43. C++标准库中的类、结构以及相关函数等被定义在名字空间【 D 】中。

- A. cplus

- B. news
- C. std
- D. namespace

44. 已知在一个类体中包含如下函数原型: `Volume operator-(Volume)const;`,

下列关于这个函数的叙述中, 错误的是【 B 】

- A. 这是运算符-的重载运算符函数
- B. 这个函数所重载的运算符是一个一元运算符
- C. 这是一个成员函数
- D. 这个函数不改变类的任何数据成员的值

45. 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);`

46. 下面四个选项中, 【 A 】是用来声明虚函数的。

- A. `virtual` B. `public`
- C. `include` D. `using namespace`

47. 实现运行时的多态性要使用【 D 】

- A. 重载函数 B. 构造函数
- C. 析构函数 D. 虚函数

48. 要实现动态联编, 必须通过【 A 】调用虚函数

- A. 对象指针 B. 成员名限定
- C. 对象名 D. 派生类名

49. 以下【 C 】成员函数表示纯虚函数

- A. `virtual int vf(int);` B. `void vf(int)=0;`
- C. `virtual void vf()=0;` D. `virtual void vf(int)`

50. 在表达式 `x+yz` 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。则 `operator+` 有\_\_\_个参数, `operator*` 有\_\_\_参数。【 C 】

- A. 2、2 B. 2、1 C. 1、2 D. 1、1

51. 一个类的友元函数或友元类可以访问该类的【 C 】

- A. 私有成员 B. 保护成员
- C. 所有成员 D. 公有成员

52. 对C++语言和C语言的兼容性, 其中描述正确的是【 B 】

- A. C兼容C++
- B. C++兼容C
- C. C++不兼容C
- D. C++部分兼容C

53. 关于公有继承, 下列说法错误的是【 D 】

- A. 基类的公有成员和保护成员被继承后作为派生类的公有成员和保护成员
- B. 派生类的其他成员可以直接访问基类的公有成员和保护成员
- C. 派生类成员和对象都无法访问基类的私有成员
- D. 派生类的对象可以访问基类的私有成员

54. 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. 使用它创建的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. 使用它创建数组时必须指定初始值

55. 面向对象程序设计数据与【 A 】放在一起，作为一个相互依存、不可分割的整体来处理。

- A. 对数据的操作 B. 信息
- C. 数据隐藏 D. 数据抽象

56. 在类中说明的成员可以使用关键字的是【 A 】

- A. public B. extern C. cpu D. register

57. 下列关于C++语言类的描述中错误的是【 C 】

- A. 类用于描述事物的属性和对事物的操作
- B. 类与类之间通过封装而具有明确的独立性
- C. 类与类之间必须是平等的关系，而不能组成层次结构
- D. 类与类之间必须是平等的关系，而不能组成层次结

58. 下面关于类中概念的描述中错误的是【 D 】

- 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. 类是创建对象的模板
- D. 类就是C语言中的结构体类型

59. 作用域运算符的功能是【 C 】

- A. 标识作用域的级别
- B. 指出其大小
- C. 标识某个成员是属于哪一个类
- D. 给出的范围

60. 下列关于模板的叙述中， 错误的是【 B 】

- A. 在一定条件下函数模板的实参可以省略
- B. 类模板不能有数据成员
- C. 在模板声明中用< >括起来的部分是模板的形参表
- D. 模板声明中的第一个符号总是关键字template

61. 下列关于C++函数的叙述中， 正确的是【 D 】

- A. 每个函数都必须有参数
- B. 每个函数都必须返回一个值
- C. 函数在被调用之前可以先声明
- D. 函数可以自己调用自己

62. 下列带缺省值参数的函数说明中， 正确的说明是【 A 】

- A. int Fun(int x, int y=2, int z=3);
- B. int Fun(int x=1, int y, int z);

C. int Fun(int x=1, int y, int z=3);

D. int Fun(int x, int y=2, int z);

63. 假定A为一个类，f()为该类公有的函数成员，a1为该类的一个对象，则访问a1对象中函数成员f()的格式为( B )。

A. a1.f B. a1.f() C. a1->f() D. (a1) .f()

64. 下列关于运算符重载的叙述中，正确的是【 B 】

A. 通过运算符重载，可以定义新的运算符

B. 有的运算符只能作为成员函数重载

C. 若重载运算符+，则相应的运算符函数名是+

D. 重载二元运算符时，必须声明两个形参

65. 假定AA为一个类，a()为该类公有的函数成员，x为该类的一个对象，则访问x对象中函数成员a()的格式为【 B 】

A. x.a B. x.a() C. x->a D. x->a()

66. 下列关于构造函数的特点，其中描述错误的是【 B 】。

A. 构造函数是一种成员函数，它具有一般成员函数的特点

B. 构造函数必须指明其类型

C. 构造函数的名称与其类名相同

D. 一个类中可以定义一个或多个构造函数

67. 假设Class Y: public X，即类Y是类X的派生类，则定义一个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. X,Y; Y,X B. X,Y; X,Y

C. Y,X; X,Y D. Y,X; Y,X

68. 对使用new申请的存储空间，释放时必须使用【 C 】

A. free B. release C. delete D. malloc

69. 下列叙述错误的是【 D 】。

A. catch (...) 语句可捕获所有类型的异常

B. 一个try语句可以有多个catch语句

C. catch (...) 语句可以放在catch语句组的中间

D. 程序中try语句与catch语句是一个整体，缺一不可

70. 所谓的数据封装就是将数据和对这些数据相关的操作组装在一起，形成一个实体，这个实体我可以称为【 A 】

A. 类 B. 函数 C. 对象 D. 数据块

71. 有关类和结构体的关系，错误的是【 B 】

A. 结构体一般只有数据成员，通常不会加入函数方法

B. 类一般只有函数方法，通常不会包含数据成员

C. 类的成员默认都是私有的

D. 结构体的成员默认都是公开的

72. 类的构造函数被自动调用执行的情况是在定义类的【 C 】

A. 成员函数时 B. 数据成员时

C. 对象时 D. 友元函数时

73. 关于this指针使用说法正确的是【 A 】



- A. 保证每个对象拥有自己的数据成员，但共享处理这些数据的代码
- B. 保证基类私有成员在子类中可以被访问。
- C. 保证基类保护成员在子类中可以被访问。
- D. 保证基类公有成员在子类中可以被访问。

74. 下面对静态数据成员的描述中，正确的是【 C 】

- 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调用
- C. 静态数据成员可以直接用类名调用
- D. 静态数据成员不能受private控制符的作用

75. 下面对友元函数的描述中正确的是【 C 】

- 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. 在类中必须用friend声明友元函数
- D. 友元函数不能访问类的私有成员

76. 如果类A被说明成类B的友元，则【 D 】

- A. 类B也一定是类A的友元
- B. 类B的成员即类A的成员
- C. 类A的成员即类B的成员
- D. 类A的成员函数可以访问类B的成员

77. 有关多态性说法不正确的是【 C 】

- A. C++语言的多态性分为编译时的多态性和运行时的多态性
- B. 编译时的多态性可通过函数重载实现
- C. 运行时的多态性可通过模板和虚函数实现
- D. 实现运行时多态性的机制称为动态多态性

78. 对基类和派生类的关系描述中，错误的是【 B 】

- A. 派生类是基类的具体化
- B. 基类继承了派生类的属性
- C. 派生类是基类定义的延续
- D. 派生类是基类的特殊化

79. 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. 一个对象调用相同名称的函数
- D. 一个对象调用不同名称的对象

80. 下列打开文件的表达式中，错误的是【 A 】

- A. `cout.open("C:\\vc\\abc.txt",ios::binary);`
- B. `ifstream ifile ("C:\\vc\\abc.txt");`
- C. `fstream iofile;iofile.open("abc.txt",ios::ate);`
- D. `ofstream ofile; ofile.open("C:\\vc\\abc.txt",ios::binary);`

81. 以下有关析构函数的叙述，不正确的是【 A 】

- A. 析构函数和构造函数一样可以有形参
- B. 析构函数名前必须冠有符号“~”
- C. 析构函数不允许用返回值
- D. 在一个类只能定义一个析构函数

82. 对类的构造函数和析构函数描述正确的是【 B 】

- A. 构造函数不能重载，析构函数可以重载
- B. 构造函数可以重载，析构函数不能重载
- 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

83. 下列描述中，成员函数中包括【 B 】的类肯定是抽象类。

- A. `virtual void draw();`
- B. `virtual void draw() = 0;`
- C. `virtual void draw(){}`
- D. `void draw() = 0;`

84. 下列关于C++函数的叙述中，正确的是【 C 】

- A. 每个函数至少要具有一个参数
- B. 每个函数都必须返回一个值
- C. 函数在被调用之前必须先声明
- D. 函数不能自己调用自己

85. 下列关于类和对象的叙述中，错误的是【 A 】

- A. 一个类只能有一个对象
- B. 对象是类的具体实例
- C. 类是对某一类对象的抽象
- D. 类和对象的关系是一种数据类型与变量的关系

86. 假定 A 为一个类，则执行 `A a1;` 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

87. 通常，拷贝构造函数的形参是【 D 】

- A. 本类对象名 B. 本类对象的指针名
- C. 本类对象的成员名 D. 本类对象的引用名

88. 下列不是类的成员函数的是【 B 】

- A. 构造函数 B. 友元函数
- C. 析构函数 D. 拷贝构造函数

89. 下面描述中，表达错误的是【 A 】

- A. 公用继承时基类中的private成员在派生类中仍是private的
- B. 公用继承时基类中的public成员在派生类中仍是public的
- C. 公用继承时基类中的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

90. 带默认参数的函数原型声明若有: void fun(int a,int b=7,char z="");下面函数调用中, 不合法的是【 C 】

A. fun(1); B. fun(1,2);

C. fun( ); D. fun(1,2,'a');

91. cout是I/O流库预定义的【 B 】

A. 类 B. 对象 C. 包含文件 D. 常量

92. C++中解决命名冲突的机制是【 D 】

A. 虚基类 B. 虚函数 C. 函数重载 D. 名称空间

93. 下面关于命名空间的说法, 错误的是【 A 】

A. 一个命名空间对应多个命名作用域

B. 一个命名空间中的标识符命名作用域相同

C. 一个命名空间中可以集合很多不同的标识符

D. 命名空间的引入让程序员可以在不同的模块中使用相同名字表示不同事物

94. 关键字【 C 】说明对象或变量初始化后不会被修改。

A. static B. public C. const D. inline

95. 下面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. 重载函数形参个数必须不同

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

96. 对基类和派生类的关系描述中, 错误的是【 B 】

A. 派生类是基类的具体化

B. 基类继承了派生类的属性

C. 派生类是基类定义的延续

D. 派生类是基类的特殊化

97. 下列关于多继承二义性的描述中, 错误的是【 B 】

A. 一个派生类的两个基类中都有某个同名成员, 派生类中这个成员的访问可能出现二义性。

B. 基类和派生类中同时出现的同名函数, 也存在二义性问题。

C. 解决二义性的最常用的方法是对成员名的限定法

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。

98. 下列有关C++类的说法中, 不正确的是【 D 】

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或类的友元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, 所有成员的访问权限均为私有的

D. 在类中, 如果不做特别说明, 所有成员的访问权限均为公用的

99. C++ 中对于类中定义的成员,其默认的访问权限为【 C 】

A. public B. protected C. private D. static

100. 下列关于类的访问权限的描述中,错误的是【 D 】

- A. 说明为公有的成员可以被程序中的任何代码访问
  - B. 说明为私有的成员只能被类的成员和说明为友元类的成员函数访问
  - C. 说明为保护的成员,除了能被本身的成员函数和说明为友元类的成员函数访问外,该类的派生类的成员也可以访问
  - D. 类的所有成员都可以被程序中的任何代码访问
- 

## 判断题

- 1. 构造函数、析构函数能被继承或取地址 (×)
- 2. 构造函数可以重载, 但析构函数只有一个 (√)
- 3. 返回值类型为引用类型的函数可以返回局部变量 (×)
- 4. 静态成员函数没有this指针 (√)
- 5. 静态成员函数可以直接访问非静态成员变量 (×)

//由于静态成员函数没有this指针, 所以不能直接访问非静态成员变量, 必须通过向函数传递当前对象作为参数才可以访问当前对象的成员变量

- 6. const类型的变量声明时可以不用初始化 (×)
- 7. const类型的对象只能访问const类型的成员函数, 构造函数、析构函数除外 (√)
- 8. const类型的对象如果作为实参, 只能传递给const类型的形参 (√)
- 9. volatile类型的对象只能访问volatile类型的成员函数, 构造函数、析构函数除外 (√)
- 10. 复合类调用构造函数时, 成员对象的构造函数调用顺序取决于复合类的 **构造函数初始化列表** (×)

//成员对象的构造函数调用顺序取决于其在复合类的声明顺序

- 11. 作为友元的类或类的函数, 必须先声明后使用 (√)
- 12. 友元关系可以继承 (×)

//友元关系不继承、不传递、不对称

- 13. 重载函数允许只有返回值类型不同 (×)

//重载函数可以在参数个数、参数类型、是否为const类型上有不同, 但如果只有返回值类型不同, 则调用函数时有歧义

- 14. 可以重载C++中不存在的运算符 (×)
- 15. 运算符重载可以改变优先级 (×)
- 16. 运算符重载不改变操作数个数 (√)
- 17. 不能重载的运算符有 ? : :: . # ## .\* -> \* \*

- 18. 返回值为临时对象的重载运算符有 后缀++ 后缀-- + - \* / %

- 19. 返回值为引用类型的重载运算符有 前缀++ 前缀-- = [] () << >>

- 20. 必须以友元方式重载的运算符有 << >>

- 21. 必须以成员函数方式重载的运算符有 = [] () ->

- 22. 派生类对象可用在基类对象使用的任何场合, 如派生类对象可以赋值给基类对象、基类指针或基类引用, 但此时这一基类的对象、指针或引用只能访问基类成员

- 23. 复合类或派生类的定义前需要包含成员对象所在类或基类的定义

- 24. 派生类构造函数的执行顺序: 基类的构造函数 (都为或都不为虚基类时按继承时声明的顺序) -> 内嵌成员对象的构造函数 (按成员对象在类中声明的顺序) -> 自身类的构造函数

- 25. 派生类对象在调用其基类的构造函数时, 若其有多个在同一层次上的基类, 则虚基类的构造先于非虚基类的构造

- 26. 派生类和基类的虚函数原型可以不一致 (×)

//如果仅是同名, 则只是普通重载; 如果仅是返回值类型不同, 则程序会出错

- 27. 虚函数可以为static类型 (×)

//静态函数、内联函数、构造函数不可以是虚函数

28. 在普通函数调用虚函数时，实际调用的是this指针指向的对象所在类的虚函数 (√)

29. 在基类的构造函数或析构函数中调用虚函数，调用的是基类的虚函数 (√)

30. 虚特性不可以继承 (×)

31. 实现多态的两种方式：重载、虚函数

32. 纯虚函数是空函数 (×)

33. 函数调用的匹配原则：

① 寻找完全匹配的函数

② 函数模板实例化后匹配

③ 重载函数参数隐式类型转化后匹配

④ 在上述某一步找到两个及以上匹配的函数或三步都找不到匹配的函数时，则程序出错

34. 析构函数不可以主动调用 (×)

//主动调用析构函数后，程序最终还会再次调用析构函数

35. 友元函数的定义必须在类外 (×)

//友元函数的定义可以在类内或类外

36. 静态成员函数的定义在类外时，需要加上static关键字 (×)

37. 在类中定义的函数是内联函数 (√)

//类中的内联函数实现有两种方式，一种是在类中进行定义；另一种是在类外进行定义，函数定义前加上incline关键字。注意第二种方式的函数定义需要放在头文件

38. 定义引用类型的变量时必须初始化 (√)

39. 在调用派生类的拷贝构造函数时，其基类的构造函数不会被调用 (×)

40.在异常处理机制中，throw的变量一定会被catch (×)

---

原文链接：[https://blog.csdn.net/m0\\_63947712/article/details/128321829](https://blog.csdn.net/m0_63947712/article/details/128321829)

原文链接：[https://blog.csdn.net/m0\\_63947712/article/details/128323913](https://blog.csdn.net/m0_63947712/article/details/128323913)