

\*\*\*求C++再爱我一次\*\*\*

1713848 张家玮

## 单选

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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,也可以是非成员函数  
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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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( ), 其他方面都应与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

等级: 1 难度: 1 区分度: 1

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, 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

等级: 1 难度: 1 区分度: 1

知识点: C++部分/多态性/抽象类的概念

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

知识点: C++部分/多态性/抽象类的工作方式

84、假定要类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

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

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

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

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

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

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: operator+有两个参数, operator\*有两个参数
- B: operator+有两个参数, operator\*有一个参数
- C: operator+有一个参数, operator\*有两个参数
- D: operator+有一个参数, operator\*有一个参数

答案: C

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

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" $\text{a} += \text{b} - \text{c}$ "被C++编译器解释为( )。

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

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

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

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

答案: A

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

知识点: C++部分/运算符重载/运算符重载规则

94、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

等级: 1 难度: 1 区分度: 1  
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: cin
- B: cout
- C: cerr
- D: clog

答案: C

等级: 1 难度: 1 区分度: 1  
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

等级: 1 难度: 1 区分度: 1  
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:  $\text{cin.read}()$
- B:  $\text{cin.get}()$
- C:  $\text{cin.put}()$
- D:  $\text{cin.getline}()$

答案: B

等级: 1 难度: 1 区分度: 1  
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:  $\text{cout} << \text{put}(\text{k});$
- B:  $\text{cout} << 'k';$
- C:  $\text{cout.put}(\text{k});$
- D:  $\text{cout.put}('k');$

答案: C

等级: 1 难度: 1 区分度: 1  
知识点: C++部分/输入输出流/输入输出流的成员函数

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

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

答案: C

等级: 1 难度: 1 区分度: 1  
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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

知识点: C++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

等级: 1 难度: 1 区分度: 1

知识点: C++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;
- C: fstream fout;
- D: fstream fout; fout("c:\\test.txt");

答案: D

等级: 1 难度: 1 区分度: 1

知识点: C++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

等级: 1 难度: 1 区分度: 1

知识点: C++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

等级: 1 难度: 1 区分度: 1

知识点: C++部分/文件/文本文件的顺序读写

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

等级: 1 难度: 1 区分度: 1

知识点: C++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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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 f1(T1 a,b)
{
    return (a>b) ? (a) : (b);
}
```

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

答案: B

等级: 1 难度: 2 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 1 区分度: 1

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

等级: 1 难度: 3 区分度: 1

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

等级: 1 难度: 1 区分度: 1

知识点: C++部分/文件/二进制文件的顺序读写

## 所有是非题

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

等级: 1 难度: 1 区分度: 1

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  
等级：1 难度：1 区分度：1  
知识点：CPP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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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  
等级：1 难度：1 区分度：1  
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函数。

答案： 错误  
等级：1 难度：1 区分度：1  
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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

知识点：CPP部分/多态性/虚函数的工作方式

87、析构函数中调用虚函数采用静态联编。

答案： 正确

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP部分/模板/类模板定义

113、用类模板定义对象时，可以省类型实参

答案： 错误

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP部分/模板/类模板定义

115、类模板描述了一组类

答案： 正确

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

知识点：CPP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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

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

等级：1 难度：1 区分度：1

知识点：CPP部分/运算符重载/特殊运算符重载