# C++试题库

## 选择题

* 下列关于对象初始化的叙述中，正确的是

A 定义对象的时候不能对对象进行初始化

B 定义对象之后可以显式地调用构造函数进行初始化

C 定义对象时将自动调用构造函数进行初始化

D 在一个类中必须显式地定义构造函数实现初始化

**答案：C.**

* 下列关于基类和派生类关系的叙述中，正确的是

A 每个类最多只能有一个直接基类

B 派生类中的成员可以访问基类中的任何成员

C 基类的构造函数必须在派生类的构造函数体中调用

D 派生类除了继承基类的成员，还可以定义新的成员

**答案：D.**

* 下列关于赋值运算符“=”重载的叙述中，正确的是

A 赋值运算符只能作为类的成员函数重载

B 默认的赋值运算符实现了“深层复制”功能

C 重载的赋值运算符函数有两个本类对象作为形参

D 如果己经定义了复制（拷贝） 构造函数，就不能重载赋值运算符

**答案：A**

* 下列定义语句中，错误的是

A int px\*;

B char \*acp[10];

C char (\*pac)[10];

D int (\*p)();

**答案：A.**

* 若MyClass为一个类，执行“MyClass a[4]，\*p[5]；”语句时会自动调用该类构造函数的次数是

A 2 B 5 C 4 D 9

**答案：C.**

* 有如下语句序列：

char str[10]；

cin>>str；

当从键盘输入“I lovethis game”时，str中的字符串是

A"I love this game" B "I love thi" C"I love" D "I"

**答案：D**

* 有定义语句：int \*p[4];以下选项中与此语句等价的是

Aint p[4]; B int \*\*p; C int \*（p[4]）; D int （\*p）[4];

**答案：C**

* 下列说法正确的是

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

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

D 类的内联函数必须在类体外通过加关键字inline定义

**答案：B**

* 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 派生类的对象可以赋给基类的对象

B 派生类的对象可以初始化基类的引用

C 派生类的对象可以直接访问基类中的成员

D 派生类的对象的地址可以赋给指向基类的指针

**答案：C**

* 下列关于多态性的描述，错误的是

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绑定

**答案：C**

* 下面关于模板的描述，错误的是

A 函数模板和类模板的参数可以是任意的数据类型

B 类模板不能直接使用，必须先实例化为相应的模板类．然后定义了模板类的对象

C 函数模板不能直接使用，需要实例化为模板函数后才能使用

D 类模板的成员函数都是模板函数

**答案：A**

* 下面对静态数据成员的描述中，正确的是

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 静态数据成员不受private控制符的作用

D 静态数据成员可以直接用类名调用

**答案：D**

* 抽象基类是指

A 嵌套类 B 派生类　 C含有纯虚函数 D 多继承类

**答案：C**

* 下面有关重载函数的说法中正确的是

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

**答案：C**

* 如下关于编译链接的说法错误的是

A)编译优化会使得编译速度变慢

B) 预编译头文件可以优化程序的性能

C) 静态链接会使得可执行文件偏大

D)动态链接库会使进程启动速度偏慢

**答案：B**

* 已知程序如下：

int s(int n)

{

return (n<=0) ? 0 : s(n-1) +n;

}

void main()

{

cout<< s(1);

}

程序运行时使用栈来保存调用过程的信息，自栈底到栈顶保存的信息一次对应的是

A．main()->S(1)->S(0)               B．S(0)->S(1)->main()

C．main()->S(0)->S(1)               D．S(1)->S(0)->main()

**答案：A**

* 下列程序的输出是：

#define add(a,b) a+b

int main(){

printf("%d\n"，5\*add(3,4));

return 0;}

A．23 B．35 C．16 D．19

**答案：D**

* 对于派生类的构造函数，在定义对象时构造函数的执行顺序为？

1：成员对象的构造函数

2：基类的构造函数

3：派生类本身的构造函数

A．123 B．231 C．321 D．213

**答案：D**

* 递归函数最终会结束，那么这个函数一定？

A 使用了局部变量

B 有一个分支不调用自身

C 使用了全局变量或者使用了一个或多个参数

D 没有循环调用

**答案：B**

* 编译过程中，语法分析器的任务是

A 分析单词是怎样构成的

B 分析单词串是如何构成语言和说明的

C 分析语句和说明是如何构成程序的

D 分析程序的结构

**答案：B**

## 填空题

1. int func(x){

int countx = 0;

while(x){

countx ++;

x = x&(x-1);

}

return countx;

}

假定x = 9999。 函数的运行结果为：**8**

有如下类定义：

class A1 {

public:

char **a**;

int **b**;

static int **c**;

};

则sizeof(A1)的输出结果为：8

有初始序列为1 8 6 2 5 4 7 3，分别写出按照以下算法的第一趟排序结果：

冒泡排序：

快速排序：

以下为Windows NT下的32位C++程序，请计算sizeof的值。

  char str[] = “Hello”

char \*p = str  int n = 10;

请计算

sizeof (str ) = \_\_\_\_5\_\_\_\_\_ sizeof ( p ) = \_\_\_\_4\_\_\_ sizeof ( n ) = \_\_\_\_4\_\_\_\_

void Func ( char str[100]) { //请计算 sizeof( str ) =\_\_\_4\_\_\_\_\_\_ }

void \*p = malloc( 100 ); //请计算 sizeof ( p ) =\_\_\_\_\_4\_\_\_\_

## 程序设计题

## 简答题