



全国计算机等级考试

2007

笔试考试习题集



南开100题

二级 C++ 语言程序设计

全国计算机等级考试命题研究组 编

南开大学出版社

全国计算机等级考试

笔试考试习题集

二级 C⁺⁺ 语言程序设计

全国计算机等级考试命题研究组 编

南开大学出版社

天 津

图书在版编目(CIP)数据

全国计算机等级考试笔试考试习题集. 二级 C++ 语言
程序技术 / 全国计算机等级考试命题研究组编. — 3 版.
天津: 南开大学出版社, 2006. 11

ISBN 7-310-02272-6

I. 全... II. 全... III. ①电子计算机—水平考试
—习题②C 语言—程序设计—水平考试—习题
IV. TP3-44

中国版本图书馆 CIP 数据核字 (2006) 第 110942 号

版权所有 侵权必究

南开大学出版社出版发行

出版人: 肖占鹏

地址: 天津市南开区卫津路 94 号 邮政编码: 300071

营销部电话: (022) 23508339 23500755

营销部传真: (022) 23508542 邮购部电话: (022) 23502200

*

天津市宝坻区第二印刷厂印刷

全国各地新华书店经销

*

2006 年 11 月第 3 版 2006 年 11 月第 3 次印刷

787×1092 毫米 16 开本 11.875 印张 294 千字

定价: 19.00 元

如遇图书印装质量问题, 请与本社营销部联系调换, 电话: (022) 23507125

出版者的话

信息时代,计算机与软件技术日新月异,在国家经济建设和社会发展的过程中,发挥着越来越重要的作用,已经成为不可或缺的关键性因素。国家教育部考试中心自1994年推出“全国计算机等级考试”以来,已经经过了十年,考生近千万人,其中,有350多万考生获得了不同级别的证书。

计算机等级考试需要考察考生的实际操作能力以及理论基础。因此,经全国计算机等级考试委员会专家的论证,以及教育部考试中心有关方面的研究,我社于2002年编写出版了《全国计算机等级考试上机考试习题集》,供考生考前学习使用。该习题集的编写、出版和发行,对考生的帮助很大,自出版以来就一直受到广大考生的欢迎。为配合社会各类人员参加考试,能顺利通过“全国计算机等级考试”,我们组织多年从事辅导计算机等级考试的专家在对近几年的考试深刻分析、研究的基础上,结合上机考试习题集的一些编写经验,并依据教育部考试中心最新考试大纲的要求,编写出这套“全国计算机等级考试笔试考试习题集”。

编写这样一套习题集,是参照上机考试习题集的做法,其内容同实际考试内容接近,使考生能够有的放矢地进行复习,希望考生能顺利通过考试。

南开大学出版社

全国计算机等级考试

笔试题分类精解与应试策略

二级 C 语言程序设计
二级 Visual Basic 语言程序设计
二级 Visual FoxPro 数据库程序设计
三级网络技术

三级信息管理技术
三级数据库技术
三级 PC 技术

冲刺指南

二级 C 语言程序设计
二级 Visual Basic 语言程序设计
二级 Visual FoxPro 数据库程序设计
三级网络技术

三级信息管理技术
三级数据库技术
三级 PC 技术

上机题分类精解与应试策略

一级 MS Office
一级 B
二级 C 语言程序设计
二级 Visual Basic 语言程序设计

二级 Visual FoxPro 数据库程序设计
三级 C 语言程序设计
三级汇编语言程序设计

上机考试习题集

一级
二级 C 语言程序设计
二级 Visual Basic 语言程序设计
二级 Visual FoxPro 数据库程序设计
三级 C 语言程序设计

三级汇编语言程序设计
二级 Access 数据库程序设计
二级 Java 语言程序设计
二级 C++语言程序设计

笔试考试习题集

二级 C 语言程序设计
二级 Visual Basic 语言程序设计
二级 Visual FoxPro 数据库程序设计
二级 Access 数据库程序设计
二级 Java 语言程序设计

二级 C++ 语言程序设计
三级 PC 技术
三级网络技术
三级信息管理技术
三级数据库技术

历届笔试真题详解

二级 C 语言程序设计
二级 Visual Basic 语言程序设计
二级 Visual FoxPro 数据库程序设计
二级 Access 数据库程序设计
二级 Java 语言程序设计

二级 C++ 语言程序设计
三级 PC 技术
三级网络技术
三级信息管理技术
三级数据库技术

历届上机真题详解

一级 MS Office
一级 B
二级 C 语言程序设计
二级 Visual Basic 语言程序设计

二级 Visual FoxPro 数据库程序设计
三级 C 语言程序设计
三级汇编语言程序设计

教育部考试中心笔试试题及答案汇编

二级 Visual Basic 和 Visual FoxPro
二级 C、C++、Java 和 Access

三级 C
三级汇编和四级

目 录

第1套.....	(1)
第2套.....	(9)
第3套.....	(16)
第4套.....	(25)
第5套.....	(33)
第6套.....	(41)
第7套.....	(51)
第8套.....	(60)
第9套.....	(68)
第10套.....	(78)
第11套.....	(86)
第12套.....	(92)
第13套.....	(97)
第14套.....	(103)
第15套.....	(110)
第16套.....	(118)
第17套.....	(126)
第18套.....	(135)
第19套.....	(146)
第20套.....	(158)
附录 参考答案.....	(168)

第 1 套 ***** 每套有选择题和填空题 2 种题型 *****

一、选择题

1. 下面关于对象概念的描述中, () 是错误的。

- A. 对象就是 C 语言中的结构体变量
- B. 对象代表着正在创建的系统中的—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

2. 结构化程序设计所规定的三种基本控制结构是 ()。

- A. 顺序结构、选择结构、循环结构
- B. 输入、输出、处理
- C. for、while、switch
- D. 主程序、子程序、函数

3. 下列程序的运行结果是 ()。

```
#include< iostream. h>
```

```
class A
```

```
{
```

```
  inta;
```

```
  public :
```

```
  A () {a=0; }
```

```
  A (int aa)
```

```
  {
```

```
    a= aa;
```

```
    cout<< a++;
```

```
  }
```

```
};
```

```
void main ()
```

```
{
```

```
  A x, y (2) , z (3) ;
```

```
  cout<< endl;
```

```
}
```

A. 00

B. 23

C. 34

D. 25

4. 已知 fl(int) 是类 A 的公有成员函数, p 是指向成员函数 fl() 的指针, 采用赋值, () 是正确的。

- A. p=fl

- B. P=A: : fl
C. P=A: : fl ()
D. P=fl ()
5. C++语言中的基本数据类型是 ()。
- A. 整型、浮点型、逻辑型、无值型和类类型
B. 整型、字符型、浮点型、无值型和逻辑型
C. 整型、浮点型、字符型、逻辑型和枚举型
D. 整型、浮点型、逻辑型、无值型和指针型
6. 下列叙述中错误的是 ()。
- A. 一个函数中可以有多条 return 语句
B. 调用函数必须在一条独立的语句中完成
C. 函数中通过 return 语句传递函数值
D. 主函数名 main 也可以带有形参
7. 结构化程序设计主要强调的是 ()。
- A. 程序的规模
B. 程序的效率
C. 程序设计语言的先进性
D. 程序易读性
8. 下面关于友元的描述中，错误的是 ()。
- 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
9. C++语言中 while 循环和 do...while 循环的主要区别是 ()。
- A. do...while 的循环体至少无条件执行一次
B. while 的循环控制条件比 do...while 的循环控制条件严格
C. do...while 允许从外部转到循环体内
D. do...while 的循环体不能是复核语句
10. 对类的构造函数和析构函数描述正确的是 ()。
- 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
11. 关于虚函数下面说的正确的是 ()。
- A. 若在重定义虚函数时使用了 virtual，则该重定义函数还是虚函数

- B. 虚函数不能声明为友元
- C. 子类必须重定义父类的虚函数
- D. 虚函数不能是 static 的

12. 关于纯虚函数和抽象类的描述中, () 是错误的。

- A. 纯虚函数是一种特殊的虚函数, 它没有具体的定义
- 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, 该基类的派生类一定不再是抽象类
- D. 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

13. 有以下程序 ()。

```
#include <iostream.h>
```

```
int f(int n)
```

```
{
```

```
if(n==1)return 1;
```

```
else
```

```
return f(n-1)+1;
```

```
}
```

```
void main()
```

```
{
```

```
int i,j=-;
```

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

```
j +=f(i);
```

```
cout<<j<<endl;
```

```
}
```

程序运行后输出结果是 ()。

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

14. 下列关于虚基类的描述中, 错误的是 ()。

- A. 虚基类子对象的初始化由最派生类完成
- B. 虚基类子对象的初始化次数与虚基类下面的派生类个数有关
- C. 设置虚基类的目的是消除二义性
- D. 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对虚基类构造函数的调用

15. 以下哪个基类中的成员函数表示纯虚函数 ()。

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

16. 若定义 cin> str;

当输入

Object Windows Programming!

所得的结果是 str () 。

- A. Object Windows Programming!
- B. Object
- C. Object Windows
- D. Object Windows Programming

17. 一个类的友元函数或友元类能够通过成员操作符访问该类的 ()。
- A. 私有成员
 - B. 保护成员
 - C. 公有成员
 - D. 公用成员、保护成员和私有成员
18. () 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. 虚析构函数
 - B. 虚构造函数
 - C. 纯虚函数
 - D. 静态成员函数
19. 线性表的顺序存储结构和线性表的链式存储结构分别是 ()。
- A. 顺序存取的存储结构、顺序存取的存储结构
 - B. 随机存取的存储结构、顺序存取的存储结构
 - C. 随机存取的存储结构、随机存取的存储结构
 - D. 任意存取的存储结构、任意存取的存储结构
20. 通常的拷贝初始化构造函数的参数是 ()。
- A. 某个对象名
 - B. 某个对象的成员名
 - C. 某个对象的引用名
 - D. 某个对象的指针名
21. 关于虚函数，下列表述正确的是 ()。
- A. 如果重定义虚函数时使用了保留字 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朋友函数
 - D. 派生类必须重新定义基类的虚函数
22. 有如下定义语句: int a[]={1,2,3,4,5};, 则对语句 int *p=a; 正确的描述是 ()。
- A. 语句 int *p=a; 定义不正确
 - B. 语句 int *p=a; 初始化变量 p, 使其指向数组对象 a 的第一个元素
 - C. 语句 int *p=a; 是把 a[0] 的值赋给变量 p
 - D. 语句 int *p=a; 是把 a[1] 的值赋给变量 p
23. 在 int a=10, *p=&a; 语句中, p 的值是 ()。
- A. 10
 - B. 变量 p 的地址

C. 变量 a 的地址

D. 随机数

24. 一个类的友元函数能够访问该类的 ()。

A. 私有成员

B. 保护成员

C. 公有成员

D. 所有成员

25. 如果一个模板声明列出了多个参数, 则每个参数之间必须使用逗号隔开, 每个参数都必须重复使用关键字 ()。

A. const

B. static

C. void

D. class

26. 对虚函数的调用 ()。

A. 一定使用动态联编

B. 必须使用动态联编

C. 一定使用静态联编

D. 不一定使用动态联编

27. 有以下程序

```
#include< iostream. h>
int f ( int n)
{if ( n==1) return 1;
else return f ( n- 1) + 1; }
void main ( )
{int i, j= 0;
for (i=1; i<3; i++) = j+=f (i) ;
cout<<j; }
```

程序运行后的输出结果是 ()。

A. 4

B. 3

C. 2

D. 1

28. 关于 getline () 函数的下列描述中, () 是错误的。

A. 该函数是用来从键盘上读取字符串的

B. 该函数读取的字符串长度是受限制的

C. 该函数读取字符串时, 遇到终止符时便停止

D. 该函数中所使用的终止符只能是换行符

29. C++中, 由类 istream 和类 ostream 而派出的一个类 (), 它提供双向 I/O 操作。

A. iostream

B. fstream

C. strstream

D. istream

30. 下面的描述中, 正确的是 ()。

A. virtual 可以用来声明虚函数

B. 含有纯虚函数的类是不可以用来创建对象的, 因为它是虚基类

C. 即使基类的构造函数没有参数, 派生类也必须建立构造函数

D. 静态数据成员可以通过成员初始化列表来初始化

31. 关于二义性的描述错误的是 ()。
- A. 一个子类的两个父类中都有某个同名成员, 在子类中对该成员访问可能出现二义性
 - B. 解决二义性可以用对成员名的限定法
 - C. 父类和子类中同时出现同名函数, 也可能存在二义性
 - D. 一个子类是从两个父类派生出来的, 而这两个父类只有一个共同的父类, 对该父类成员进行访问时, 可能出现二义性
32. 下列关于运算符重载的描述中, () 是正确的。
- 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
33. 当输入 “Thank you” 时下列程序的输出结果为 ()。
- ```
#include <iostream.h>
#include <iomanip.h>
void main()
{
 char line[80];
 cin.getline(line, sizeof(80), 'k');
 cout<<line;
}
```
- A. Tha
  - B. Thank you
  - C. Thank
  - D. 以上结果都不对
34. 下列关于动态联编的描述中, 错误的是 ( )。
- A. 动态联编是以虚函数为基础的
  - B. 动态联编是在运行时确定所调用的函数代码的
  - C. 动态联编调用函数操作是指向对象的指针或对象引用
  - D. 动态联编是在编译时确定操作函数的
35. 下列关于虚基类的描述, 错误的是 ( )。
- A. 设置虚基类的目的是为了消除二义性
  - B. 虚基类的构造函数在非虚基类之后调用
  - C. 若同一层中包含多个虚基类, 这些虚基类的构造函数按它们说明的次序调用
  - D. 若虚基类由非虚基类派生而来, 则仍然先调用基类构造函数, 再调用派生类的构造函数

## 二、填空题

1. 以下程序的执行结果是\_\_\_\_\_。

```
#include <iostream.h>
```

```

void main ()
{
cout.fill ('*');
cout.width (10);
cout<<123.45<<endl;
cout.width (8);
cout<< 123.45<< endl;
cout.width (4);
cout << 123.45<< endl;
}

```

2. 有如下的程序:

```

#include<iostream>
#include<fstream>
using namespace std;
int main(){
 char s[25]="Programming language";
 ofstream f1("DATA. TXT");
 f1<<"C ++ Programming";
 f1.close();
 ifstream f2("DATA. TXT");
 if(f2.good())f2>>s;
 f2.close();
 cout<<s;
 return 0;
}

```

执行上面的程序将输出\_\_\_\_\_。

3. 大多数 C++ 程序都要包含\_\_\_\_\_头文件。该文件中包含了所有输入/输出流操作所需要的基本信息。

4. 一个类可以从直接或间接的祖先中继承所有属性和方法。采用这个方法提高了软件的\_\_\_\_\_。

5. 若有:

```

int i=1000;
int *ip=&i;
void*VP;
vp=ip;

```

其中, vp=ip 的含义是\_\_\_\_\_。

6. switch 后面括号中的表达式只能是整型、\_\_\_\_\_或枚举型表达式。
7. 头文件\_\_\_\_\_中包含了执行内部格式化所需的信息。
8. 长度为 n 的顺序存储线性表中, 当在任何位置上插入一个元素概率都相等时, 插入一个元素所需移动元素的平均个数为\_\_\_\_\_。
9. 类中数据成员和成员函数的默认类型为\_\_\_\_\_。
10. 下面程序的运行结果是\_\_\_\_\_。
- ```
#include <iostream.h>
void main()
{
    char str[]="SSSWLIA",c;
    int k;
    for(k=2;(c=str[k]!='\0';k++)
    {
        switch(c)
        {
            case 'I': ++k; break;
            case 'l': continue;
            default: cout << c; continue;
        }
        cout << '*';
    }
}
```
11. 数组元素 a[i] 是该数组中的第_____个元素。
12. 请定义一个函数名为 A, 返回值为 int, 没有参数的纯虚函数的定义是_____。
13. 类的具体表现是通过_____定义来操作的。
14. 指针变量所保存的不是一般的数据值, 而是程序中另一个对象的_____。
15. 在一个 C++ 程序中, 每个变量都必须遵循_____的原则。

第 2 套 ***** 每套有选择题和填空题 2 种题型 *****

一、选择题

1. 以下哪个特征不是面向对象思想中的主要特征 ()。
A. 多态 B. 继承 C. 封装 D. 垃圾回收
2. 下面关于虚函数的描述中, 正确的是 ()。
A. 虚函数是一个静态成员函数
B. 虚函数是一个非成员函数
C. 虚函数既可以在函数说明时定义, 也可以函数实现时定义
D. 派生类的虚函数与基类中对应的虚函数具有相同的参数个数和类型
3. 有关构造函数的说法不正确的是 ()。
A. 构造函数名字和类的名字一样
B. 构造函数在说明类变量时自动执行
C. 构造函数无任何函数类型
D. 构造函数有且只有一个
4. 对于常数据成员, 下面描述正确的是 ()。
A. 常数据成员可以不初始化, 并且不能更新
B. 常数据成员必须被初始化, 并且不能更新
C. 常数据成员可以不初始化, 并且可以被更新
D. 常数据成员必须被初始化, 并且可以被更新
5. 下列描述中错误的是 ()。
A. 派生类可以作为基类派生其他的子类
B. 派生类继承基类的所有数据成员
C. 派生类可以有多个基类
D. 派生类不能继承一些函数成员
6. 实现输出为八进制的符号是 ()。
A. dec B. hex C. oct D. setw (8)
7. 派生类的对象对它的基类成员中 () 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

8. 下列说法中错误的是 ()。

- 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. 私有继承时基类中的 `public` 成员在派生类中是 `private` 的
- D. 保护继承时基类中的 `public` 成员在派生类中是 `protected` 的

9. 有如下程序段

```
&include< iostream. h>
void main ( )
{ int a=14, b=15, x;
char c='A';
x= (a&& b) && (c<'B'==;
cout<<x;=
```

执行该程序段后, x 的值为

- A. ture B. false C. 0 D. 1

10. 若有说明: `int s[3][4]={0}`; 则下面叙述正确的 ()。

- A. 只有元素 `s[0][0]` 可得到初值 0
- B. 此说明语句不正确
- C. 数组 `s` 中各个元素都可得到初值, 但其值不一定为 0
- D. 数组 `s` 中每个元素均可得到初值 0

11. 对虚函数的调用 ()。

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

12. 编译时的多态性可以通过使用 () 获得。

- A. 虚函数和指针 B. 重载函数和析构造函数
- C. 虚函数和对象 D. 虚函数和引用

13. 以下程序的结果是 ()。

```
#include<iostream.h>
void main ( )
{ char *p=" abcdefgh", *r;
long *q;
```