

面向对象程序设计

高等教育自学考试同步辅导 / 同步训练

全国高等教育自学考试指定教材辅导用书

蔡振法 / 主编

计算机及应用专业 (独立本科段)

F E B

煤炭工业出版社



全国高等教育自学考试指定教材辅导用书

高等教育自学考试同步辅导/同步训练

计算机及应用专业(独立本科段)

面向对象程序设计

主 编 蔡振法

副主编 谢明岗

崔孝凤

于长锐

煤炭工业出版社

图书在版编目 (CIP) 数据

面向对象程序设计/蔡振法主编. —北京: 煤炭工业出版社, 2002

(高等教育自学考试同步辅导/同步训练)

全国高等教育自学考试指定教材辅导用书

ISBN 7-5020-2158-2

I. 面… II. 蔡… III. 面向对象语言—程序设计
—高等教育—自学考试—自学参考资料 IV. TP312

中国版本图书馆 CIP 数据核字 (2002) 第 026519 号

全国高等教育自学考试指定教材辅导用书
高等教育自学考试同步辅导/同步训练

面向对象程序设计

蔡振法 主编

责任编辑: 王铁根

*

煤炭工业出版社 出版

(北京市朝阳区芍药居 35 号 100029)

山东济宁市火炬书刊印务中心 印刷

新华书店北京发行所 发行

*

开本 $880 \times 1230 \text{mm}^{1/32}$ 印张 $9^{1/4}$

字数 259 千字 印数 1—10,000

2002 年 5 月第 1 版 2002 年 5 月第 1 次印刷

社内编号 4929 定价 16.00 元

版权所有 违者必究

本书如有缺页、倒页、脱页等质量问题, 本社负责调换

说 明

为了满足高等教育自学考试社会助学和适应考试的需要，我们组织编写了这本《面向对象程序设计》（计算机及应用专业——独立本科段）同步辅导/同步训练辅导用书。

编写依据：

（1）严格遵照全国高等教育自学考试指导委员会颁布的《面向对象程序设计自学考试大纲》的命题原则和命题范围。

（2）全国高等教育自学考试指导委员会组编的指定教材《面向对象程序设计》（刘振安主编，经济科学出版社）为依据编写的。

本书特点：

本书以自学考试大纲规定的考核知识点及能力层次为线索，按照大纲和指定教材的顺序章节进行辅导，涵盖了大纲所要求的全部考核内容。本书以章为单位，将自学考试中每一章节可能出现的考核知识点给予相应的提示，并配有大量的同步练习，同时给出参考答案，最后附模拟试题两套。

本书的每章均由“内容提示”、“同步练习”和“参考答案”三部分组成。“内容提示”列出了每章的重点内容，然后以“同步练习”的形式加以巩固，为考生系统地复习教材内容提供了全面的配套练习。为便于考生查阅答案，每章均附有“参考答案”。书后所附两套模拟试题的目的在于帮助考生检验所学知识，掌握考题类型，提高应试能力，顺利通过自学考试。

本书可供高等教育自学考试集体组织学习或个人自学使用，也

可供相关专业人士参加其他考试使用。在编写过程中，虽然努力耕耘，但由于时间仓促，加之编者水平有限，错误之处在所难免，敬请广大读者、学者同仁批评指正。

本书在编写过程中，得到了山东省自学考试委员会张新元同志的大力支持与帮助，在此深表谢意。

编 者

2002年4月

目 录

第一章 面向对象及 C++ 基础知识	(1)
内容提示	(1)
同步练习	(3)
参考答案	(14)
第二章 类和对象	(23)
内容提示	(23)
同步练习	(26)
参考答案	(38)
第三章 构造函数与析构函数	(52)
内容提示	(52)
同步练习	(54)
参考答案	(73)
第四章 继承和派生类	(85)
内容提示	(85)
同步练习	(88)
参考答案	(107)
第五章 多态性和虚函数	(123)
内容提示	(123)
同步练习	(125)
参考答案	(141)
第六章 进一步使用成员函数	(156)
内容提示	(156)
同步练习	(158)

参考答案	(177)
第七章 运算符重载及流类库	(193)
内容提示	(193)
同步练习	(196)
参考答案	(203)
第八章 模板	(215)
内容提示	(215)
同步练习	(218)
参考答案	(229)
第九章 进一步掌握面向对象程序设计	(241)
内容提示	(241)
同步练习	(244)
参考答案	(256)
模拟试题(一)	(267)
参考答案	(278)
模拟试题(二)	(281)
参考答案	(287)

第一章 面向对象及 C++ 基础知识

内 容 提 示

本章主要介绍面向对象设计的基础知识和 C++ 语言对 C 语言的基本改进部分。学习本章需重点掌握以下知识点:

一、面向对象程序设计基础知识

1. 面向对象程序设计方法:面向对象的程序设计是通过为数据和代码建立分块的内存区域来提供对程序进行模块化的一种程序设计方法,这些模块可以被用作样板在需要时建立其拷贝。面向对象程序设计方法要求语言必须具备抽象、封装、继承和多态性这几个关键要素。

2. C++ 与 C 语言的关系:C 语言虽然有着简洁、高效和接近汇编语言等一系列的突出优点,但其自身也存在着类型检查机制较弱、几乎没有支持代码重用的语言结构、不适合于开发大型程序等局限。为此,1980 年贝尔实验室的 Bjarne Stroustrup 博士及其同事开始对 C 进行改进和扩充,形成最初的“带类的 C”,1983 年取名为 C++。C++ 语言是 C 语言的扩充,保持与 C 兼容。

二、C++ 基本程序结构

1. 输入输出:“cout <<”:把待输出的内容送到标准输出设备(显示器);“cin >>”:把从标准输入设备(键盘)接收到的数据存入其后的变量。使用 C++ 风格的输入输出,在程序中必须包含头文件“iostream.h”。

2. 注释方式:C++ 可以使用两种注释方法:从“//”开始,直到行

尾都将被计算机当做注释;多行注释仍旧使用“/*……*/”。

3. **const 修饰符**:用类型修饰符 `const` 声明的变量只能被读取,该变量必须在声明时被定义,并且它的值在程序中不能被改变。对基本数据类型的 `const` 变量,编译器将其视为一个常量,不再为它分配内存,并且每当在程序中遇到该变量的名字时,都用在说明它时所给出的初始值取代它。

4. **函数原型和缺省参数**:函数原型标识一个函数的返回类型,同时也标识该函数参数的个数和类型。函数原型的重要作用是可以使编译器检查一个函数调用表达式中可能存在的问题;在函数调用时,C++ 引进了一种新类型的参数:缺省参数。缺省参数就是不求程序员设定参数,而由编译器在需要时给该参数赋予预先设定的值。缺省参数都是在函数原型中说明的,且都必须放在参数序列的最后。

5. **动态内存分配**:在程序运行时为程序分配内存的一种方法。运算符 `new` 用于进行内存分配, `delete` 用于释放 `new` 分配的内存。

6. **内联函数**:使用关键字 `inline` 说明内联函数。在 C++ 中,除具有循环语句、`switch` 语句的函数不能说明为内联函数外,其它函数都可以说明为内联函数。

7. **引用**:引用运算符“&”用于定义一个引用。引用不是变量,而是给变量起一个别名,在程序中对引用的存取都是对它所引用的变量进行的。不能说明引用的引用,也不能说明元素类型为引用数组或指向引用的指针,但可以说明对指针的引用。C++ 提供引用,主要用途之一是建立函数参数的引用传递方式,另一个主要用途是用于返回引用的函数。

8. **编译指令**:C++ 所有的编译指令都以“#”开始,共包括嵌入指令、宏定义、条件编译指令三种。

三、改变习惯重新思考

改变习惯重新思考,要记住以下几点:

1. 用 C++ 来编译 C 程序,了解 C++ 的严格检查机制。
2. 重新设计 C 程序,掌握 C++ 的新特性。

3. 外部变量说明放在头文件中(培养好习惯)。
4. 少用预处理(# define)而用内联子过程(inline)和 const(定义常量)。
5. 要重视函数类型。任何函数都要定义其类型,而且是严格的。

同步练习

一、名词解释题

1. 面向过程
2. 面向对象程序设计
3. 抽象
4. 封装
5. 多态性
6. 动态内存分配
7. 引用

二、填空题

1. C++ 的注释符号有两种,一种是沿用 C 语言的_____符号,一种是新引入的_____符号。
2. 将一组对象的共同特征进一步抽象出来,从而形成“_____”的概念。
3. 面向对象的程序设计方法要求语言必须具备_____,_____,_____和_____这几个关键要素。
4. C++ 最初被称为“_____”。
5. 使用 C++ 风格的输入输出,在程序中必须包含头文件“_____”。
6. 当一个函数调用出现在函数定义之前时,必须先对函数进行_____。
7. 运算符_____用于进行动态内存分配,运算符_____用于释放动态分配的内存。
8. 使用关键字_____说明的函数称内联函数。
9. C++ 使用运算符_____来定义一个引用,对引用的存取都是对它所引用的_____的存取。

10. C++ 所有的编译指令都以_____开始,共包括_____,
_____,_____ 三种。

三、单项选择题

1. C++ 对 C 语言做了很多改进,下列描述中使得 C 语言发生了质变,即从面向过程变成面向对象的是()。

- A. 增加了一些新的运算符
- B. 允许函数重载,并允许设置默认参数
- C. 规定函数说明必须用原型
- D. 引进了类和对象的概念

2. 关于 C++ 与 C 语言的关系的描述中,错误的是()。

- A. C 语言是 C++ 的一个子集
- B. C++ 与 C 语言是兼容的
- C. C++ 对 C 语言进行了一些改进
- D. C++ 和 C 语言都是面向对象的

3. 面向对象的程序设计将数据与()放在一起,做为一个相互依存、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

4. 面向对象系统的()是一种信息隐藏技术,目的在于将对象的使用者与设计者分开,使用者不必知道对象行为实现的细节,只需用设计者提供的协议命令对象去做即可。

- A. 多态性
- B. 封装性
- C. 继承性
- D. 复用性

5. 下面哪项不是面向对象系统所包含的要素?()

- A. 重载
- B. 对象
- C. 类
- D. 继承

6. 下面关于类概念的描述中,错误的是()。

- 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 语言中的结构类型

7. 下面关于对象概念的描述中错误的是()。

- A. 对象就是 C 语言中的结构变量
 - B. 对象代表着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
8. 下列说法错误的是()。
- A. 抽象是一种从一般的观点看待事物的方法,它要求程序员集中于事物的具体细节或具体实现
 - B. 将一组对象的共同特征进一步抽象出来,从而形成“类”的概念
 - C. 程序是一组抽象的对象——类组成的
 - D. 不同类的对象具有不同的行为
9. 下面说法中不正确的是()。
- A. 封装把一个数据结构同操作数据的函数衔接在一起构成一具有类类型的对象
 - B. 在 C++ 中,封装是借助于函数来达到的
 - C. 封装要求一个对象应具备明确的功能和一个接口
 - D. 对象的内部数据结构不可访问,这称为数据隐藏
10. 下列说法错误的是()。
- A. 封装是将一组数据和这组数据有关的集合组装在一起,形成一个能动的实体
 - B. 封装要求一个对象应具备明确的功能
 - C. 封装和其他对象相互作用时并不需要一个确定的接口
 - D. 封装使得一个对象可以像一个部件一样用在各种程序中,而不用担心对象的功能受到影响
11. 下面说法正确的是()。
- 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

12. 一个函数功能不太复杂,但要求被频繁调用,应选用()。
- A. 内联函数 B. 重载函数 C. 递归函数 D. 嵌套函数
13. 可以使编译器显示报错信息,并中止对程序的编译的编译指令是()。
- A. # ifdef B. # endif C. # error D. # define
14. 下面的语句编译运行后,正确的结果是()。
- ```
define one 1
define two one + one
define three two * 5
i = two * three;
```
- A. i = 20 B. i = 11 C. i = 7 D. i = 10
15. 下面的类定义中,哪个更有意义? ( )
- A. class A1 { int x; };
- B. class A2 { int x, y; };
- C. class A3 { int x; public A3( int n) { x=n; } };
- D. class A4 { int x; A4( int n) { x=n; } };

#### 四、多项选择题

1. C++ 头文件约定使用扩展名( )。
- A. .cpp B. .c C. .h D. .hpp
2. 下列语句格式或说法正确的是( )。
- A. void main(){};
- B. cout >> "This is the start of something wonderful!";
- C. cin >> cc;
- D. # include <iostream.h>;
- E. C 只能使用/\* .... \*/作为注释符,C++ 只能使用//作为注释符
3. 下面语句完全正确的是( )。
- A. void fun(int w, int x = 1, int y = 1; int z = 1);
- B. void fun(int w = 1, int x = 2, int y = 3, int z);
- C. void fun(int w = 1, int x, int y = 2, int z);

D. void fun(int w = 1, int x, int y, int z);

4. 下面说法正确的是( )。
- A. 新行符“\n”和操纵算子 endl 作用一样
  - B. 任何变量都必须在使用前进行说明
  - C. cin >> a 将键盘中输入的一个数据或字符, 送到变量 a 中保存起来
  - D. 标准输入设备一般指键盘, 标准输出设备一般指屏幕
5. C 语言的特点是( )。
- A. C 是结构化的程序设计语言
  - B. C 的类型检查机制相对较强
  - C. C 可以部分地取代汇编语言
  - D. C 具有较高的可移植性
6. C++ 的特点是( )。
- A. C++ 具有简洁、高效和接近汇编语言的特点
  - B. C++ 对 C 的重要扩充是支持面向对象的程序设计
  - C. C++ 尤其适用于中等和大型的程序开发项目
  - D. C++ 本身几乎没有支持代码重用的语言结构
7. 下面说法正确的是( )。
- A. 每条编译指令单独占用一行, 同一行不能有其他编译指令和 C++ 语句(注释例外)
  - B. 编译指令是 C++ 指令系统的一部分
  - C. 要嵌入程序员自己定义的头文件, 必须使用编译指令:  
#include "文件名"
  - D. 要嵌入程序员自己定义的头文件, 必须使用编译指令:  
#include <文件名>
8. 下面说法正确的是( )。
- A. 所有的函数都可以说明为内联函数
  - B. 具有循环语句、switch 语句的函数不能说明为内联函数
  - C. 使用内联函数, 可以加快程序执行的速度, 但会增加程序代码的大小
  - D. 使用内联函数, 可以减小程序代码大小, 但使程序执行的

### 速度减慢

9. 下面说法正确的是( )。
- A. 对象是计算机内存中的一块区域,它可以存放代码和数据
  - B. 对象实际是功能相对独立的一段程序
  - C. 各个对象间的数据可以共享是对象的一大优点
  - D. 在面向对象的程序中,对象之间只能通过函数调用相互通信
10. 下面说法正确的是( )。
- A. 将数据和与其相关的操作方法置于同一个函数内,即为数据封装
  - B. 一个对象通过继承可以获得另一个对象的特性
  - C. 面向对象要求程序员集中于事物的本质特征,用抽象的观点看待程序
  - D. 同一函数为不同的对象接受时,产生的行为是一样的,这称为一致性

### 五、问答题

1. C++ 语言与 C 语言的本质差别是什么?
2. 面向过程的程序设计方法与面向对象的程序设计方法,在对待数据和函数关系上有什么不同?
3. 如果为引用提供的初始值是一个常量或是一个使用 const 修饰的一个不可修改的变量,编译器将如何处理? 如果所说明的引用类型与初始化引用时所使用的变量的类型不一致,又将如何?
4. 从学习 C 语言转向学习 C++ 语言,要改变习惯重新思考问题,应如何做起?

### 六、改错题

1. 改正下面程序中的错误,使其完成两个数的求和计算,并能正确编译执行。

```
include <stdio.h>
```

```

main()
{ int a,b,c;
 cout << "Enter two numbers:";
 cin << a << b;
 c = sum(a,b);
 cout << "sum is:" << c << endl;
}

void sum(int a, int b)
{ return a + b; }

```

2. 不能说明引用的引用,不能说明元素类型为引用数组或指向引用的指针,也不能说明对指针的引用。

3. `char * const p` 表明 `p` 是一个指针,它只能指向一个字符串常量。

4. 用类型修饰符 `const` 声明的变量,可以在声明后对其赋初值。

5. 使用指令 `# include` 嵌入一个文件时,必须可以在指令中包含有路径信息,例如:

```
include < \ user \ prog. h >
```

6. 下列的程序定义了一个交换两个浮点数的函数 `swap`,并在主函数中调用这个函数进行两个浮点数的交换。请指出错误,并改正之。

```

void swap(float &a, float &b);
void main()
{ float a = 1, b = 2;
 swap(&a, &b);
}

void swap(float &a, float &b)
{ float temp = a;
 a = b;
 b = temp;
}

```

## 七、程序分析题

1. 分析下面的程序 prog1 的输出结果。

```
//prog1
include < iostream.h >
void main()
{ int num = 500;
 int& ref = num;
 cout << ref;
 ref = ref + 100;
 cout << " " << num;
 num = num + 50;
 cout << " " << ref;
}
```

2. 分析下面的程序 prog2 的输出结果。

```
//prog2
include < iostream.h >
void main()
{ int num = 500;
 int& ref = num;
 cout << ref << endl;
 ref = ref + 100;
 cout << ref << endl;
 cout << num << endl;
}
```

3. 写出下面程序的输出结果。

```
include < iostream.h >
int &f(int &i)
{ i += 15; return i; }
void main()
{ int k = 0;
```