



面向对象 程序设计

应试指导

(本科)

- 统览全局 归纳知识要点
- 突出重点 掌握求解思路
- 举一反三 详析典型例题
- 抛砖引玉 提高应试能力

尹为民 陈玉科 编著



► 计算机及应用专业自学考试同步辅导丛书

面向对象程序设计应试指导

(本科)

尹为民 陈玉科 编著

清华大学出版社

北 京

内 容 简 介

本书根据全国高等教育自学考试委员会公布的《面向对象程序设计自学考试大纲》编写,是《面向对象程序设计》课程同步练习与考前复习的指导书。

本书结合自考的特点,根据自考大纲的知识点、要点及层次来组织内容,介绍了面向对象程序设计中的类和对象、构造函数与析构函数、继承和派生类、多态性与虚函数、成员函数、运算符重载及流类库、模板等知识。书中各章高度提炼出教材中的知识体系,按层次结构介绍主要的知识点,对部分难点配以直观的图示,并结合典型例题进行分析与解答,同时设计了大量的习题及其参考答案,供考生练习,检查学习效果。

本书在覆盖全部考核内容的前提下,强化了考核重点与难点内容的讲解与练习,有助于提高实际应试能力。本书不仅能满足参加计算机及应用专业自学考试的复习、训练的需要,也能作为大专院校学习 C++ 语言或面向对象程序设计等相应课程的辅助教材及参考书。

版权所有,盗版必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

面向对象程序设计应试指导(本科)/尹为民,陈玉科编著. —北京:清华大学出版社,2003
(计算机及应用专业自学考试同步辅导丛书)

ISBN 7-302-06657-4

I. 面… II. ①尹…②陈… 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(2003)第 037380 号

出版者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.com.cn>

印刷者:北京市耀华印刷有限公司

发行者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:14.75 字数:352 千字

版 次:2003 年 6 月第 1 版 2003 年 6 月第 1 次印刷

书 号:ISBN 7-302-06657-4/TP·4980

印 数:0001~5000

定 价:19.00 元

丛 书 序

为了适应社会主义现代化建设的需要,我国于 1981 年开始实行高等教育自学考试制度。它是个人自学、社会助学和国家考试相结合的一种教育形式,是高等教育的有机组成部分,其职责就是在高等教育这个水平上倡导自学、鼓励自学、帮助自学、推动自学,为每一位自学者铺就成才之路。20 余年来,高等教育自学考试以其严格的质量和良好的声誉得到了社会的普遍关注,近千万的考生通过自学考试获得了本科、大专和中专学历文凭。

随着计算机技术在我国各个领域的推广和普及,越来越多的行业与单位把操作和应用计算机作为劳动者必须掌握的一种基本技能。许多单位已把掌握一定的计算机知识和应用技能作为干部录用、职务晋升、职称评定、上岗资格的重要依据。故近年来参加计算机及应用专业自学考试的考生越来越多。

计算机行业是一个发展迅猛的行业,技术在不断进步,社会需求也在不断地随之变化,因而自学考试大纲也进行了若干调整,国家教育部考试中心从 2000 年开始,正式执行自学考试新计划,同时施行新编的大纲和教材。虽然新编自学考试教材适合自学,有利于学习者培养实践意识,提升自学能力,但仍无法满足广大应试人员成功通过考试的迫切需要。

为了满足广大自学应考者的学习、复习和应试的要求,北京科海培训中心精心策划了这套“计算机及应用专业自学考试同步辅导丛书”。本套丛书包括:

- 计算机网络与通信应试指导(本科)
- 计算机应用技术应试指导(专科)
- 数据库及其应用应试指导(专科)
- 数据库原理应试指导(本科)
- 计算机网络技术应试指导(专科)
- 数据结构应试指导(本科)
- 数据结构导论应试指导(专科)
- 汇编语言程序设计应试指导(专科)
- 面向对象程序设计应试指导(本科)
- 计算机组成原理应试指导(专科)
- 计算机系统结构应试指导(本科)
- 操作系统概论应试指导(专科)
- 操作系统应试指导(本科)

丛 书 特 点

本套丛书紧扣国家教育部考试中心最新颁布的考试大纲，以指定教材为基础，由长期工作在一线的教学一线的教授、副教授、讲师亲自编写，从结构设计、内容安排到例题、练习题都经过精心设计与整理。丛书具有以下特点：

- 以考试大纲的各项要求和各章的考核知识点为主线，梳理学习要点，归纳知识体系。
- 注重基础、突出重点，以便考生对课程内容建立一个整体的概念。
- 深入浅出，条理清晰，语言通俗易懂。
- 注意对学生解题能力的培养，书中详细分析了大量的例题，并通过大量的针对性练习来强化对考核重点与难点的理解与应用。

编写过程中，严格按照指定教材的章节顺序安排内容。每一章首先列出总体要求、学习重点与难点，让读者做到心中有数，明白学习这一章要达到什么样的目标，什么是难点，什么是重点，特别要注意哪些地方。然后分知识体系、例题分析、练习题与参考答案 3 部分进行介绍。**知识体系**开宗明义，先给出知识体系结构图，让读者从整体上全面把握篇章结构，了解各部分之间的联系，复习起来思路明确、条理清晰；接下来对重点内容进行适当讲解。**例题分析**通过典型例题的分析和解答使学生在掌握基本概念的同时，进一步加深对内容的综合理解和应用。**练习题与参考答案**覆盖全部考核内容，同时加大重点内容的覆盖密度，习题类型与考试要求有关，包括选择题、简答题、改错题、程序分析题、程序填空题。

使 用 说 明

本丛书是与高等教育自学考试指定教材配套使用的同步辅导用书，知识点部分突出强调了考试重点，例题和练习题部分则覆盖了全部考核内容，还包含了指定教材中的部分课后习题。例题和练习题中涉及的个别概念在知识点部分可能未曾提及，所以最好与指定的教材配套使用本丛书。

前 言

“面向对象”不仅仅作为一种思想、一种技术，更作为一种方法论贯穿于软件设计的各个阶段。专家们预测，面向对象的程序设计思想、方法将会主导今后程序设计的发展方向。

本书是全国高等教育自学考试《面向对象程序设计》课程的复习与考试练习的辅导教材，根据自考大纲的知识点、要点及层次，结合自考的特点来组织内容。自学考试的特点是考试题型固定，要求掌握的内容及深度明确，但试题覆盖面广。为了帮助考生更好地掌握教学内容，理解考试大纲的要求及规定的考试题型，特编写了本书。

为了满足自考的需要，本书的章节按自考指定教材《面向对象程序设计》一书的相应结构来安排，内容上融入了课堂教学的素材和实例。每章首先给出教材中的知识体系结构，便于考生把握每章知识的层次关系与整体结构；然后介绍主要的知识点并对部分难理解的内容配以直观的图示，以帮助理解；再针对主要的学习内容给出典型例题分析及解答；最后按自考调整后的题型设计了一定量的练习题并给出其参考答案，供考生练习、检查学习效果。

《面向对象程序设计》是一门新开考不久的课程，其内容涉及到许多新概念及新方法，初学者较难透彻地了解与掌握。有些内容自学难度大，解题时往往感到无从下手。所以，作者借此机会，对几年来的教学实践进行了归纳，将自己对知识的认识、理解、教学经验融入到本书中，并尽可能从考生的角度出发，对难以理解的概念及方法进行了深入浅出的讲解，以求高度概括、易学易懂。希望通过对本书的阅读，能让考生把握本课程的主线，加深对基本概念的理解，掌握求解问题的思路与方法，提高各种题型的分析、解题能力以及应试能力。

学习 C++ 语言的关键是学会面向对象程序设计的思维及方法，这就需要花大力气从面向过程的方法中转变过来，习惯面向对象的编程思想及方法。考生应掌握 C++ 语言的基本结构和主要特征。深刻理解 C++ 语言的对象和类，重点掌握构造函数与析构函数的作用及对象的各种初始化方法。牢固掌握最为重要的有

关继承的知识，不仅要学习继承的特征与方法，还应学会如何应用。虚函数与多态性是本课程的重点和难点，学习时应结合考虑类的封装及继承关系。在自学时要注意与教材同步，参照每章的考核知识点复习，还要注意辅导材料给出的题型变化。希望本书能够成为考生的良师益友。

本书在编写过程中得到了武汉大学计算机学院领导、同行们的指导与大力支持；李春葆教授对书稿进行了认真仔细的审阅，并提出了宝贵的意见和建议；本书在编写过程中参考了许多书籍及文献资料，在此谨向被引用资料的作者和给予帮助的专家及同学表示感谢。

本书旨在帮助考生认识、理解考试内容和题型，做好应试准备，取得更理想的成绩。由于本书涉及的题量较大，在内容上可能会存在不够完整或疏漏之处，恳请广大读者批评指正。

尹为民

2003年1月于武昌

目 录

第 1 章 面向对象及 C++ 基础知识	1
1.1 知识体系	1
1.1.1 知识层次	1
1.1.2 知识点	2
1.2 例题分析	7
1.3 练习题与参考答案	10
1.3.1 单项选择题	10
1.3.2 多项选择题	12
1.3.3 简答题	14
1.3.4 改错题	15
1.3.5 程序分析题	16
第 2 章 类和对象	18
2.1 知识体系	18
2.1.1 知识层次	18
2.1.2 知识点	18
2.2 例题分析	23
2.3 练习题与参考答案	29
2.3.1 单项选择题	29
2.3.2 多项选择题	31
2.3.3 简答题	36
2.3.4 改错题	37
2.3.5 程序分析题	38
第 3 章 构造函数与析构函数	44
3.1 知识体系	44
3.1.1 知识层次	44
3.1.2 知识点	44
3.2 例题分析	47
3.3 练习题与参考答案	56
3.3.1 单项选择题	56
3.3.2 多项选择题	58
3.3.3 简答题	62
3.3.4 程序改错题	64

3.3.5 程序分析题.....	68
第4章 继承和派生类.....	73
4.1 知识体系.....	73
4.1.1 知识层次.....	73
4.1.2 知识点.....	74
4.2 例题分析.....	78
4.3 练习题与参考答案.....	87
4.3.1 单项选择题.....	87
4.3.2 多项选择题.....	90
4.3.3 简答题.....	95
4.3.4 改错题.....	98
4.3.5 程序分析题.....	100
第5章 多态性和虚函数.....	115
5.1 知识体系.....	115
5.1.1 知识层次.....	115
5.1.2 知识点.....	116
5.2 例题分析.....	119
5.3 练习题与参考答案.....	128
5.3.1 单项选择题.....	128
5.3.2 多项选择题.....	131
5.3.3 简答题.....	135
5.3.4 改错题.....	137
5.3.5 程序分析题.....	141
5.3.6 程序填空题.....	146
第6章 进一步使用成员函数.....	147
6.1 知识体系.....	147
6.1.1 知识层次.....	147
6.1.2 知识点.....	147
6.2 例题分析.....	151
6.3 练习题与参考答案.....	156
6.3.1 单项选择题.....	156
6.3.2 多项选择题.....	158
6.3.3 简答题.....	161
6.3.4 程序分析题.....	163
第7章 运算符重载及流类库.....	167
7.1 知识体系.....	167
7.1.1 知识层次.....	167

7.1.2 知识点.....	167
7.2 例题分析.....	170
7.3 练习题与参考答案.....	176
7.3.1 单项选择题.....	176
7.3.2 多项选择题.....	177
7.3.3 简答题.....	179
7.3.4 程序分析题.....	181
第8章 模板.....	185
8.1 知识体系.....	185
8.1.1 知识层次.....	185
8.1.2 知识点.....	185
8.2 例题分析.....	188
8.3 练习题与参考答案.....	193
8.3.1 单项选择题.....	193
8.3.2 多项选择题.....	195
8.3.3 简答题.....	196
8.3.4 改错题.....	199
8.3.5 程序分析题.....	200
第9章 面向对象程序设计基础.....	203
9.1 知识体系.....	203
9.1.1 知识层次.....	203
9.1.2 知识点.....	203
9.2 例题分析.....	206
9.3 练习题与参考答案.....	208
9.3.1 单项选择题.....	208
9.3.2 多项选择题.....	209
9.3.3 简答题.....	210
9.3.4 程序分析题.....	212
9.3.5 程序填空题.....	221
参考文献.....	226

第 1 章 面向对象及 C++ 基础知识

【学习要求】 理解面向对象程序设计的基础知识及 C++ 语言中的新特点与风格。

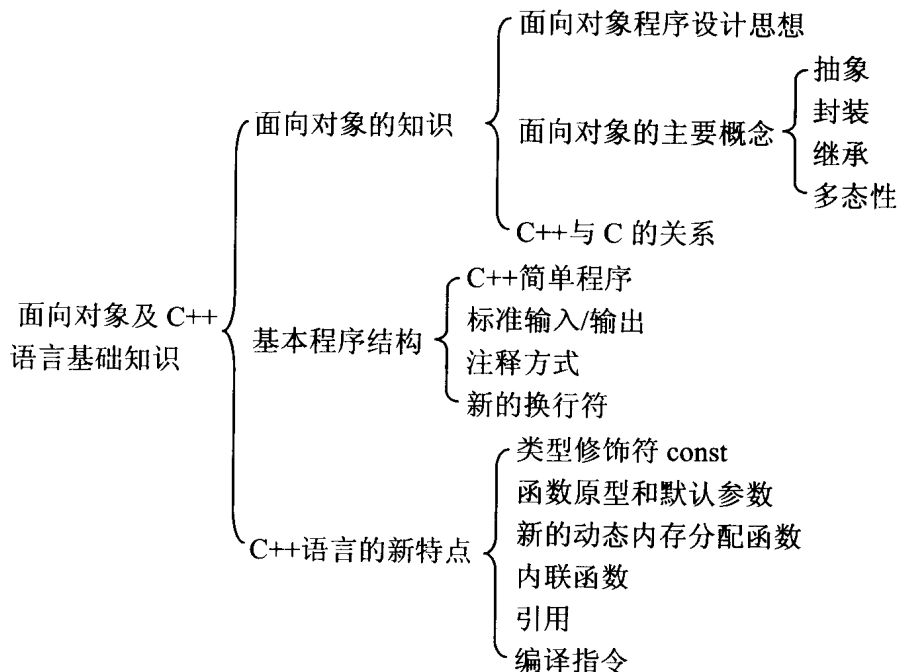
【学习目的】 掌握基本概念及术语，为下一步学习 C++ 的类打下基础。

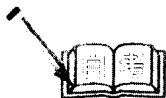
【本章重点】 面向对象的主要概念、基本程序结构、类型修饰符、函数原型、内联函数及引用等新特点。

【本章难点】 引用的概念及使用。

1.1 知识体系

1.1.1 知识层次





1.1.2 知识点

1. 面向对象的知识

(1) 面向对象程序设计思想

面向对象是一种认识世界的方法，也是一种程序设计的方法。面向对象的观点认为，客观世界就是由各种各样的对象组成的。每种对象都有自己的内部状态和运动规律，不同对象间的相互联系和相互作用就构成了各种不同的系统，进而构成整个客观世界。按照这样的思想设计程序，就是面向对象的程序设计。

面向过程的程序设计：采用自顶向下、逐步求精、模块化的方法；模块内部由顺序、分支、循环基本控制结构组成；它基于功能，以过程为中心，将数据与操作分离。

面向对象的程序设计：将数据与操作封装在一起作为对象来处理；以对象为中心进行程序设计；具有抽象、封装、继承和多态性等特征。

(2) 面向对象的主要概念

① 抽象

抽象是指将一组对象的共同特征抽象形成“类”的概念。程序正是由一组抽象的对象类组成的。运行时，不同对象类产生的对象具有不同的行为。

② 封装

封装是将一组数据和这组数据有关的操作代码组装在一起，形成一个能动的实体——对象。封装要求一个对象具有明确的功能及接口，以便和其他对象相互作用。同时，对象的内部是受保护的，只有属于这个对象的操作代码才可以访问该对象的内部数据。封装使得一个对象可以像部件一样用在各种程序中。

③ 继承

继承指一个对象类可以获得另一个对象类的属性和行为。

如图 1.1 所示。学生、教师和商人均可继承人类的属性和行为。如：均可有姓名、年龄、身高、体重等属性，具有会活动、做事、睡眠等行为。

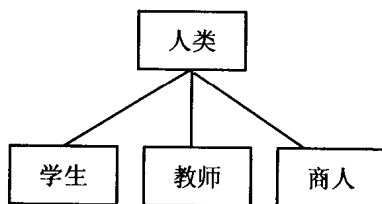


图 1.1 继承示例

利用继承可以大大提高程序共享、重用及扩充的能力。

④ 多态性

不同对象调用相同名称的操作函数时，可导致完全不同行为的现象称为多态性。

利用多态性，可大大提高程序的灵活性及人们解决复杂问题的能力。



(3) C++与 C 的关系

C++语言是 C 语言的扩充；C 语言是 C++的子集；C++与 C 语言兼容。

C++支持面向对象程序设计，它是在标准 C 语言的基础上，引入“面向对象”概念而扩充形成的混合型面向对象语言。

2. 基本程序结构

(1) C++简单程序

C++简单程序是由引入部分、函数首部及函数体构成的。

(2) C++风格的输入/输出

C++可以根据数据的类型，自动使用合适的输出方式。但必须包含头文件“`iostream.h`”。

如果使用的是新式编译器，可以使用 `iostream` 来代替 `iostream.h`。在这种情况下，还应当增加代码行“`using namespace std`”，也就是说，将下面的代码：

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

替换为：

```
#include <iostream>
using namespace std
```

`cout<<表达式`：可将表达式的值送到标准输出设备（显示器）。

`cin>>变量`：可将标准输入设备（键盘）接收的数据存入指定的变量。

(3) 注释方式

从//开始，到行尾结束，都将被计算机当做注释。多行注释仍旧可以使用 C 风格注释，C 注释包括在符号/*和*/之间。

(4) 新的换行符

C++使用 `endl` 换行。可以在一条语句中多次使用换行符。

3. C++语言的新特点

(1) 类型修饰符 `const`

类型修饰符 `const` 用来声明一个常量，必须在声明常量时赋予初始值，其值在程序中只能被读取，不能被改变，并且每当在程序中遇到该常量名，编译器都采用声明时赋予的初始值代替它。

(2) 函数原型和默认参数

函数原型标识一个函数的返回类型、参数的个数和参数的类型（即只有函数首部而无函数体）。当函数调用出现在函数定义之前时，必须先对函数进行原型说明。其一般



形式为：

返回类型 函数名（参数类型说明表）；

默认参数就是不要求程序员设定参数，而由编译器在需要时给该参数赋予预先设定的值。默认参数都是在函数原型中被说明的，且都必须放在参数序列的最后。

（3）新的动态内存分配函数

运算符 `new` 用于进行动态内存分配，形式为：

```
p=new type // 分配一般变量空间
p=new type[数组元素个数] // 分配数组变量空间
```

其中 `p` 是一个指向 `type` 型数据的指针变量，它保存 `new` 分配的内存的首地址。而运算符 `delete` 则释放 `new` 分配的内存。形式一般为：

```
delete p //p 为一般变量
delete [ ] p //p 为数组变量
```

举个例子说明 `new`、`delete` 的使用情况。

```
int *s;
int *p;
s=new int (100);
p=new int [10];
//上述语句结果如图 1.2 所示。
..... //省略使用 s、p 的部分。
delete s;
delete []p;
```

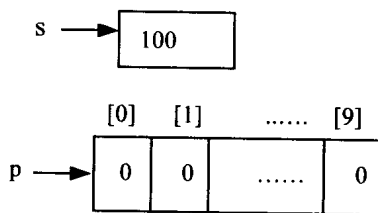


图 1.2 存储空间示意图

（4）内联函数

使用关键字 `inline` 说明的函数称为内联函数。内联函数可以在所有调用该函数的地方对这个函数的函数体进行替换。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，其他函数都可以说明为内联函数。

内联函数必须在程序第一次调用此函数语句之前出现。



(5) 引用

运算符&用于定义一个引用。引用就是给变量起一个别名，使变量的别名和原变量共用一个地址。这样，无论对哪个变量进行修改，都是对同一地址的内容进行修改。因此变量和对它的引用总是具有相同的值。引用的引入方便了对变量的使用，其形式有以下几种：

① 引用变量

变量引用的形式为：

类型 & 变量 = 被引用变量

例如：

```
int i=15;
int & j =i;
j=i+5;
```

3个语句执行的情况分别如图 1.3 (a)、(b)、(c) 所示。

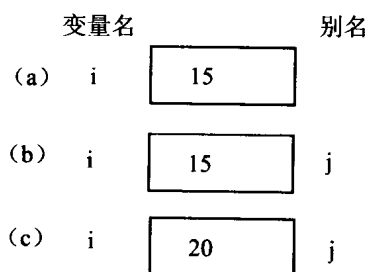


图 1.3 引用示例

② 引用参数

参数引用的形式为：

类型 & 变量

例如：

```
void swap(int &a, int &b);
```

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，通过该方式，在程序中不用指针也能改变实参的值。

③ 引用返回值

引用返回值的形式为：

返回类型 &函数名 (参数表)

```
{
    ..... //函数体语句定义
```



```
return 非局部量;  
}
```

引用的另一个主要用途是用于返回引用的函数。引用是能出现在赋值号左边的表达式，如：函数名(实参表) = 表达式；其目的是使函数可被赋值，以提高效率。



说明

- 为引用提供的初始值必须是一个变量。如果是用 `const` 修饰的一个不可修改的常量，则编译器首先建立一个临时变量，然后将该常量的值置入临时变量中，引用是引用该临时变量，对引用的存取是对该临时变量的存取。
- 如果所说明的引用类型与初始化引用时所使用变量的类型不一致，则编译器同样建立一个和引用类型同类型的临时变量，然后将该变量的值转换成引用类型的值，置入临时变量中并引用。对引用的访问同样是访问该临时变量。
- 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。

(6) 编译指令

所有的编译指令都以 `#` 开始，每条指令单独占用一行，同一行不能有其他编译指令和 C++ 语句（注释例外）。编译指令不是 C++ 的一部分。

① 宏定义

定义宏的形式如下：

```
#define 宏名 (参数表) 字符串
```

功能：用一个指定的标识符（即宏名）来代表一个字符串。若含有参数，还要进行参数替换。

解除宏定义的形式如下：

```
#undef 宏名 1 [, 宏名 2, ...]
```

它的功能是终止宏定义的作用域。



说明

为避免宏定义的不安全因素，对于无参数宏，尽量用 `const` 代替；对于有参数宏，尽量用内联函数代替。

② 嵌入指令

嵌入指令用于文件包含处理。其格式有如下两种：

- `#include "文件名"`
- `#include <文件名>`



③ 条件编译的 3 种形式

(a) #ifdef 标识符

程序段 1

[#else

程序段 2]

#endif

(b) #ifndef 标识符

程序段 1

[#else

程序段 2]

#endif

(c) #if 表达式

程序段 1

[#else

程序段 2]

#endif

此外,关键字 `define` 用于判定一个标识符是否已经被 `#define` 定义。若已被定义,则 `define` (identifier) 为真, 否则为假。

1.2 例题分析

【例 1.1】 阅读、分析下面的程序, 并说明 C++ 简单程序的组成及所用语言的新特点。

```
#include<iostream.h>
void mian( )
{
    float r;
    const float pi=3.14159; // 说明 Pi 为常量, 且初始值为 3.14159。
    cout <<"input r=";
    cin>>r;
    float s=pi * r * r;
    cout<<endl<<"area is="<<s<<endl;
}
```

引入部分

函数首部

【解答】 该程序的功能是利用给出的半径计算并输出圆面积。

C++ 简单程序是由引入部分、函数首部及函数体构成的, 其新特点如下所述:

- 本程序使用了 C++ 风格的输入 / 输出, 即: