

3导 自考 3导丛书
导学·助学·备考

最新版

核心学案

自学考试新教材·计算机及应用专业(二)

面向对象程序设计

教材依据/经济科学出版社《面向对象程序设计》 刘振安/主编
组编/全国高等教育自学考试命题研究组



同步辅导同步过关



指定教材核心浓缩



预测试卷历年真题



航空工业出版社

应
对
自
考
课
程
大
规
模
修
订
后
新
教
材
内
容



导数·导学·导考

高等教育自学考试3导丛书

面向对象程序设计

教材依据 / 经济科学出版社《面向对象程序设计》 主编 / 刘振安
编 / 全国高等教育自学考试命题研究组

应对自考课程大规模修订后新教材内容

自学考试 **新教材**

核心学案



航空工业出版社

图书在版编目(CIP)数据

面向对象程序设计/自学考试命题研究组,《面向对象程序设计》编委会编. —北京:航空工业出版社, 2005. 6

(自学考试新教材核心学案. 计算机及应用专业. 第2辑)

ISBN 7-80183-608-1

I. 面... II. ①自... ②面... 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(2005)第 056049 号

面向对象程序设计

Mianxiang Duixiang Chengxu Sheji

航空工业出版社出版发行

(北京市安定门外小关东里 14 号 100029)

发行部电话:010-84926529 010-64978486

北京市通县华龙印刷厂印刷

全国各地新华书店经售

2005 年 6 月第 1 版

2005 年 6 月第 1 次印刷

开本: 850 × 1168 1/32

印张: 55

字数: 2030 千字

(全 10 册) 定价: 140.00 元

简介



张立勇 一个普通的农民孩子，清华大学打工8年，一直坚持刻苦自学，不仅80分以上通过四级、六级考试，托福考试630分，而且获得了北京大学本科文凭。2004年10月共青团中央向张立勇颁发了“中国青年学习成才奖”，他被誉为共青团中央树立的全国十大杰出学习青年之一。

张立勇的事迹被中央电视台“东方之子”“面对面”“新闻会客厅”等多个栏目采访报道，被北京电视台、中国教育电视台等电视媒体，新浪网、雅虎网等网络媒体，《人民日报》《中国青年报》《大学生》等报纸杂志，共100多家媒体采访报道，在社会上引起很大反响。被众多青年学子视为学习的榜样。

“**因**为我选择了这样一条自己的人生道路，所以我没有机会像大多数的学子那样，经历从学校到学校，顺利地接受高等教育的过程。我只能通过自学来圆我的大学梦。”

“**我**常常想，上帝会厚爱每一个人的，它会用不同的方式对你所付出的艰辛和努力给予补偿。但是，上帝只钟爱那些自助的人。如果你不努力，你不拼搏，所有的机会都会和你失之交臂。如果在这十年之中，我放弃了对人生理想和人生价值的追求，那么，当这一切机遇到来的时候，我又怎么可能把握住呢？”

“**大**家觉得我是一个榜样，但我个人并不这么想。社会把我放到这样的位置，充当这样的角色，能够影响一些人，这是最让我自豪的。”

----- 张立勇



★ 编委会 ★

导教 导学 导考



编委主任：程 琨 魏 莹



编委名单：（按姓氏笔画排列）

万 鹏 刘 斌 刘海飞 刘 涛

闫树茂 宋玉珍 张 沁 张远盛

肖 果 邰桂英 崔海燕 程 琨

董金波 董 蕾 蒋 怡 魏 莹



★ 前言 ★

导学·导考·导学



“其实人的智力相差并不悬殊,可毅力的差距却使每个人拥有各自不同的前途。尤其是对于参加自考的人来说,毅力是非常重要的,当然还需要有得当的学习方法。”

“有很多人抱怨自考难以通过,然而正是这种严格的管理制度保证了自考毕业生的质量,使自考生获得了社会的认可和一致的好评。”

——一名从自考获得本科学历后又考上硕士生直到博士生的成功者的自述

参加自学考试,除了需要具备以上成功者所提到的毅力和方法外,还应该了解自考的每门课程都采用我们通常所说的“过关”考试——只要通过课程的一次性考试,就可拿到课程的学分,通过某专业要求课程的全部考试,也就会顺利获得这个专业的自考毕业证。然而,一分之差也会导致参考课程过关失败,有些考生难免多次重考才能修完规定课程。因此,在本书的编写过程中,编委们反复研讨自学考试的特点,努力寻求帮助自考生的有效途径。本书是多位学者、专家,历时数年的产物,具有以下优点。



掌握核心内容,了解命题动态,注重知识系统化

了解命题精神,是自学考试的核心,是达到专业标准的关键。自学考试的课程命题以课程自学考试大纲为依据,以最新指定教材为范围。本书紧紧贴住每一门课程的考试大纲和指定教材,用【考纲要求提示】、【知识结构图示】、【核心内容速记】、【同步精华题解】、【典型例题解析】等多个栏目解剖教材内容,是一套脉络清晰的速成讲义,可以使考生在厚厚的教材中抓住重点,对教材的系统学习有极强的指导作用。同时,对于临考考生,它又可以成为离开教材仍能独立使用的贴身笔记。《核心学案》摒弃了一些辅导书的题海战术,引导考生重视教材的学习。那么怎样去自学才能弄懂教材并将厚书读“薄”呢?抓住重点才是关键。《核心学案》用清晰的思路,帮助考生将教材知识系统化,使考生在答卷时知识系统、逻辑清晰、胸有成竹。



依据权威资料,重视最新信息,紧跟时代脉搏

参加高等教育自学考试的考生,常常会感到市面上的辅导资料甚至教材都有



滞后性。全国高教自考办也认可这一事实,并采取了一些有效措施,比如在发布考试大纲和指定教材的基础上又组编了《全国高等教育自学考试活页丛书》等补充学习材料,并明文规定增补内容纳入统一命题范围,要占卷面5~10分。同时高教自考办还加快了教材的修订频率。面对这种情况,原有的一些辅导资料的严重滞后和内容缺陷也是必然的。本套《核心学案》则高度重视这一现象,在依据考试大纲和指定教材时,选用高教自考办的最新修订本(2004年起自考课程已在做大规模修订),并将活页丛书等内容融会贯通其中,有的科目还特意增加了【最新内容补充】以引起考生重视。另外,本套书还吸收了许多自考强化班的授课精华,目的是帮助考生了解最新考试动态。我们还将开通网上自考辅导随时更新有关内容和提供特色售后服务,欢迎点击 www.study-book.com.cn。

三

做到讲练结合,力求精讲精练,提高辅导命中率

本套书配有【同步精华题解】和综合演练习题,是在对考纲、教材归纳总结后选编的一些经典同步练习题。这些练习题的题型与考试题型完全一致,使考生能够迅速掌握答题方法与同步要点。另外,本书的编者还依据各科内容,遴选考点,在对历年实考真题做详细分析的基础上精编了《命题预测试卷》。这些试卷不仅题型题量完全与真考试卷保持一致,而且力求覆盖考试大纲的各科重点。考生如果在学习《核心学案》的基础上再认真研习《命题预测试卷》,既可熟悉题型、了解试卷难易度,又可将其作为自测、练习之用,找出差距,查漏补缺。因此,在《核心学案》的首印首发优惠活动中,为了帮助考生用好的学习方法提高应试过关率,我们特意将《命题预测试卷》作为《核心学案》的赠品送给每个考生。这样,本书即成为真正具有命中率的辅导用书。

总之,面对数千万的自考考生,我们是抱着高度的责任感来完成这项使命的。我们的目的是:减轻考生的学习负担;我们口号是:用最短的时间使考生自考过关!因为工作量的巨大和考期的压力,也许我们遗留了某些不足,欢迎读者批评指正。来函可致: reader@study-book.com.cn,我们将高度重视,以求完善。

编者



第一章 面向对象及C++ 基础知识

考纲要求提示	(1)
知识结构图示	(1)
核心内容速记	(1)
同步精华题解	(7)
历年真题回放	(14)



第二章 类和对象

考纲要求提示	(15)
知识结构图示	(15)
核心内容速记	(15)
同步精华题解	(19)
历年真题回放	(29)



第三章 构造函数与析构函数

考纲要求提示	(30)
知识结构图示	(30)
核心内容速记	(30)
同步精华题解	(33)
历年真题回放	(44)



第四章 继承和派生类

考纲要求提示	(46)
知识结构图示	(46)
核心内容速记	(46)
同步精华题解	(50)
历年真题回放	(59)



第五章 多态性和虚函数

考纲要求提示	(62)
知识结构图示	(62)
核心内容速记	(62)
同步精华题解	(65)
历年真题回放	(76)



第六章 进一步使用成员函数

考纲要求提示	(78)
知识结构图示	(78)
核心内容速记	(78)
同步精华题解	(81)
历年真题回放	(89)



第七章 运算符重载及流类库

考纲要求提示	(91)
知识结构图示	(91)
核心内容速记	(91)
同步精华题解	(95)
历年真题回放	(100)



第八章 模板

考纲要求提示	(102)
知识结构图示	(102)
核心内容速记	(102)
同步精华题解	(105)
历年真题回放	(110)



第九章 进一步掌握面向对象程序设计

考纲要求提示	(112)
知识结构图示	(112)
核心内容速记	(112)
同步精华题解	(116)



综合演练题	(118)
-------------	-------



综合演练题参考答案	(127)
-----------------	-------



第一章 面向对象及C++ 基础知识

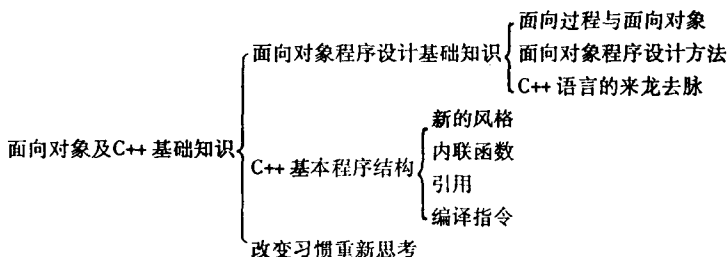


考纲要求提示

理解面向对象程序设计的思想及C++ 语言中的新思想,为学习C++ 的类打下基础。



知识结构图示



核心内容速记

一、面向对象程序设计基础知识

(一) 面向过程与面向对象

现代计算机是面向算法的自动机,算法通过程序告诉计算机;构成程序的符号系统是语言,它是描述算法的编程工具。计算机语言(程序设计语言)的发展过程就是其功能不断完善、描述问题的方法更加接近人类思维过程的过程。

C 语言是美国贝尔(Bell)实验室开发成功的。它的表达式简洁,具有丰富的运算符和良好的控制结构与数据结构。目前,其应用领域已不限于系统软件的开发,而成为当前最流行的程序设计语言之一。

所谓“面向过程”,就是不必了解计算机的内部逻辑,而是把精力主要集中在解题算法的逻辑和过程的描述上,使用过程语言编写程序,通过程序把解决问题的执行步骤告诉计算机。

C++ 则是在标准 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。面向对象方法的产生,是计算机科学发展的要求。和传统的程序设计方法相比,面向对象的程序设计具有抽象、封装和多态性等特征。

首先,面向对象是一种认识世界的方法,其次,面向对象也是一种程序设计方法。面向对象的观点认为,客观世界是由各种各样的实体,也就是对象组成的。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,并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。

(二) 面向对象程序设计方法

面向对象的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。

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,以便提供对程序进行模块化的一种程序设计方法。这些模块可以被用做样板,在需要时再建立其副本。

1. 抽象

面向对象鼓励程序员以抽象的观点看待程序,即程序是由一组抽象的对象组成的。另一方面,又可以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。

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本质特征,而不是具体细节或具体实现。面向对象鼓励程序员以抽象的观点看待程序,也就是说,程序是一组抽象的对象——类组成的。程序员从一组对象为起点,抽取公共的行为将其放入到一个类中,这基本上是抽象分类的观点,不同类的对象具有不同的行为。

类的概念来自于人们认识自然、认识社会的过程。在这一过程中,人们主要使用两种方法:由特殊到一般的归纳法和由一般到特殊的演绎法。在归纳的过程中,把一个个具体的事物的共同特征抽取出来,形成一个一般的概念,这就是“归类”;在演绎的过程中,又把同类的事物,根据不同的特征分成不同的小类,这又是“分类”。对于一个具体的类,它有许多具体的个体,这些个体叫做“对象”。

2. 封装

所谓数据封装,就是将一组数据和与这组数据有关的操作集合组装在一起,形成一个能动的实体,也就是对象。在这种情况下,用户是不可以直接操作数据的,他必须通过和数据相关的操作来访问数据。换句话说,数据封装就是给数据提供了与外界联系的标准接口,无论是谁,只有通过这些接口,使用规范的方式,才能访问这些数据。

封装要求一个对象应具备明确的功能,并具有接口以便和其他对象相互作用。同时,对象的内部实现(代码和数据)是受保护的,外界不能访问它们,只有局部于对象的代码才可以访问该对象的内部数据。对象内部数据结构的不可访问性称为数据隐藏。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。

数据封装是软件工程发展的必然产物。它一方面使得程序员在设计程序时可以专注于自己的对象,另一方面也切断了不同模块之间数据的非法使用,减少了出错的可能性。

3. 多态性

不同的对象可以调用相同名称的函数,并可导致完全不同的行为的现象称为多态性。利用多态性,程序中只需进行一般形式的函数调用,函数的实现细节留给接受函数调用的对象。

(三) C++ 语言的来龙去脉

C++ 语言是 C 语言的扩充。1980 年,贝尔实验室的 Bjarne Stroustrup 博士及其同事开始对 C 语言进行改进和扩充,最初被称为“带类的 C”,1983 年才取名为 C++,以后又经过不断完善和发展,成为目前的 C++ 语言。一方面,它将 C 语言作为它的子集,使它能与 C 语言兼容,这就使许多 C 语言代码不经修改就可以为 C++ 语言所用,用 C 语言编写的众多的库函数和实用软件可以用于 C++ 语言中。另一方面, C++ 语言支持面向对象的程序设计,这是对 C 语言的最重要改进。目前, C++ 语言已被应用于程序设计的众多领域,实践证明,它尤其适用于中型和大型的程序开发项目。从开发时间、费用到形成的软件的可靠性、可重用性、可扩充性、可维护性等方面都显示出 C++ 语言的优越性。

二、C++ 基本程序结构

(一) 新的风格

1. 新的输入和输出风格

C++ 语言的输入/输出,是通过流来实现的。C++ 是自带输入和输出的,并且可以根据数据的类型,自动使用合适的输出方式。

“cout <<”的功能是把后面的内容送到标准输出设备(这里是显示器);“cin >>”是把标准输入设备(键盘)接收到的数据,存入后面的变量。如果把“<<”和“>>”看做表示方向的符号,我们就会发现,数据确实是在按照一定的方向流动,这就是“流”这个名称的由来。

2. 灵活的注释方式

C++ 提供了一种新的注释方式:从“//”开始,直到行尾,都将被计算机当做注释。

另外,C风格的多行注释在C++中也仍然可以使用。一般情况下,多行注释仍旧使用“/*……*/”,而短的注释则较多使用行注释方式“//”。

3. 告别宏定义

在C语言中,宏定义是一个重要内容。无参数的宏作为常量,而带参数的宏则可以提供比函数调用更高的效率。在C++中,由于const修饰符和内联函数的引入,无论是带参数的宏还是不带参数的宏,都失去了存在的必要。

const修饰符的使用很简单。事实上,对基本数据类型的变量,一旦加上const修饰符,编译器就将其视为一个常量,不再为它分配内存,并且每当在程序中遇到它时,都用在说明时所给出的初始值取代它。使用const可以使编译器对处理内容有更多的了解,从而允许对其进行类型检查,同时还能避免对常量的不必要的内存分配,并可改善程序的可读性。

const还可以修饰指针变量。以char类型为例,有以下三种情况:

(1) const int * p 这表明p是一个指针,它只能指向一个被const修饰的int类型的变量,即只能指向一个整型常量。

(2) char * const p 这表明指针p本身是常量,即p指向一个固定的char类型的地址,而p的内容却是可以修改的。

(3) const char * const p = "abcd" 这里指针和它所指的内容都是常量,必须对指针进行初始化,且对任何一个进行修改都是错误的。

4. 使用函数原型和缺省参数

在C++中,函数原型是一个很重要的概念。任何函数,如果缺少了函数原型,C++都将无法编译。函数原型使得C++能够提供更强的类型检查,将函数调用表达式中可能存在的问题发现在编译阶段。

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。C++编译器从一个函数定义中抽取该函数的函数原型。程序员也可在程序中使用函数说明语句来说明一个函数的原型。

函数说明语句一般形式为:

类型 函数名(参数类型说明列表);

其中“列表”是用逗号隔开的一个类型说明,其个数和指定的类型必须和函数定义中的一致,例如:

```
int sum (int,int);
```

在函数说明中也可以给出参数名,例如:

```
int sum (int first,int second);
```

在函数调用时,引进了一种新类型的参数:缺省参数。缺省参数就是不要程序员设定该参数,而由编译器在需要时给该参数赋予预先设定的值。当程序员需要传递一个与预先设定不同的值时,必须显式地指明。缺省参数是

在函数原型中说明的。

5. 使用新的动态内存分配函数

在C语言中,动态内存分配是通过系统函数 `malloc()`、`free()` 和运算符 `sizeof` 来实现的。在C++中,上述用法已经很少使用了,取代它们的是C++的 `new` 和 `delete` 两个运算符。

(1) 运算符 `new` 用于进行内存分配,它的使用形式为:

```
p = new type;
```

其中 `type` 是一个数据类型名, `p` 是指向该类型的指针。用 `new` 建立数组类型变量的一般形式为:

```
p = new type[size];
```

`new` 的最大优点是不必进行类型转换。

(2) 运算符 `delete` 释放 `new` 分配的内存。它的一般使用形式为:

```
delete p;
```

其中 `p` 必须是一个指针,指向 `new` 分配的内存的首址。

6. 使用新的换行符

在语句

```
cout << "Hello," << name << "!" << endl;
```

中,C++使用 `endl` 换行。这个换行符号可以用在其他位置。例如:

```
cout << endl << "How are you?" << endl << endl;
```

上述语句在“`How are you?`”的前后各产生一个空行。

(二) 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,其他函数都可以说明为内联函数。

(三) 引用

引用的引入是C++对C语言的一项重大改进。所谓引用,就是给变量起一个别名,换句话说,是使新变量和原变量共用一个地址。这样,无论对哪个变量进行修改,其实都是对同一地址的内容进行修改。因而原变量和新变量,规范地说是变量和它的引用总是具有相同的值。

C++是通过引用运算符 `&` 来定义一个引用的。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。

(四) 编译指令

1. 嵌入指令

嵌入指令 `#include` 指示编译器将一个源文件嵌入到带有 `#include` 指令的源文件中该指令所在的位置处。

角括号或双引号中的文件名可含有路径信息,例如:

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

注意:由于编译指令不是C++的一部分,因此,在这里表示反斜杠时只使用一个反斜杠。如果在C++程序中表示上述文件名,则必须使用双反斜杠。

例如:

```
char fname[ ] = " << user << prog. h";
```

2. 宏定义

#define 指令定义一个标识符及串,在源程序中每次遇到该标识符时,编译器均用定义的串代替之。该标识符称为宏名,而将替换过程称之为宏替换。#define 指令用以进行宏定义,其一般形式为:

```
#define 宏名 替换正文
```

“宏名”必须是一个有效的C++标识符,“替换正文”可为任意字符组成的字符序列。“宏名”和“替换正文”之间至少有一空白相隔。C++程序员习惯将宏名写成大写字体。注意,宏定义由新行结束,而不以分号结束。如果给出了分号,则它也被视做替换正文的一部分。

#define 可以定义带有参数的宏。带参数的宏很像函数,每当遇到这样的宏时,编译将程序中的实际参数代替宏中有关的参数。

当替换正文要书写在多行上时,除最后一行之外,每行的行尾要加上一个反斜线,表示宏定义继续到下一行。

宏定义有许多不安全因素,所以,对需要使用无参数宏的场合,应该尽量使用 const 代替其定义。对使用有参数宏的场合,可以用内联函数达到同样的效果。

3. 条件编译指令

条件编译指令是#if, #else, #elif 和 #endif, 它们构成类似于C++的 if 选择结构,其中 endif 表示一条指令结束。

编译指令#if 用于控制编译器源程序的某部分有选择地进行编译。该部分从#if 开始,到#endif结束。如果#if 后的常量表达值为真,则编译这部分,否则就不编译该部分,这时,这部分代码相当于被从源文件中删除掉。

编译指令#else 在#if 测试失效的情况下建立另外一种选择。为了防止#else分支被编译,程序无法被用户使用,可以使用编译指令#error 输出出错信息。#error 使用的形式为:

```
#error 出错信息
```

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

三、改变习惯重新思考

与C语言完全兼容是C++最大的优势,这可以使人们不必放弃原有的C程序而直接转向面向对象的程序设计。

(一)通过重新编译已有C程序了解C++

由于C++的编译器比C语言的编译器更加严格,因此重新编译时会发现许多不一致性和一些小的错误。通过增加类型说明,严格函数说明以及在程序中直接增加#include头文件等手段来使该函数说明更加确切,就可以很好地解决这些问题。

(二)通过重新设计C程序的核心部分掌握C++的新特性

通过对一个熟悉的C语言程序,加进C++的新特性,可以使人们将软件的核心部分转向C++,尽快熟悉C++的新特性。

(三)将所有外部变量的说明放到头文件中

应该养成将所有extern说明放到头文件中去的习惯。extern说明对于规模较大的软件项目来说是非常关键的,因为编译器需要依赖它们。extern说明是人们和编译器之间达成的一种协议。

(四)减少使用预处理

C++内联子过程都是程序语句的一部分,并不属于预处理的范围。所有函数都应该用内联函数来代替。

(五)重视函数类型

重视函数类型问题构成了避免错误发生的第一道防线。通过努力,产生错误的机会就会大大减少。

同步精华题解

一、单项选择题

1. 在C++中,函数原型能够标识_____。()
 - A. 函数的返回类型
 - B. 函数参数的个数
 - C. 函数参数类型
 - D. 以上都正确
2. 下列说法错误的是_____。()
 - A. 在面向对象程序设计中,只有局部于对象的代码才可以访问存储于这个对象中的数据
 - B. 在面向对象程序设计中,对象之间只能通过函数调用相互通信
 - C. 在面向对象程序设计中,内存块的结构可被用做样板产生对象的更多拷贝
 - D. 在面向对象程序设计中,对象易受未知的外部其他的事件影响,而