

COMPUTER

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

高等学校计算机基础及应用教材

C++ 程序设计基础



电子工业出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY

周霏如 林伟健 编著

高等学校计算机基础及应用教材

C++ 程序设计基础

周霭如 林伟健 编著

电子工业出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书是为高等学校计算机专业、信息类和电类专业的学生开设高级语言程序设计课程而编写的教材。根据高级语言程序设计的基本体系,结合结构化程序设计和面向对象程序设计的基本方法,本书内容包括:C++语言的基本语法,程序流程控制,数据的表示和应用,两种程序模块——函数和类。本书详尽地介绍了C++面向对象的重要特征,包括类、继承、多态和虚函数等内容。本书配有多种形式的习题,还有与本教材配套的实验指导与习题解答。

本书内容循序渐进、难点分散、通俗而不肤浅,概念清晰,例题丰富。本书适合作为理工科高等学校教材,也适用于学习C++语言的读者作为参考书。

未经许可,不得以任何方式复制或抄袭本书之部分或全部内容。

版权所有,侵权必究。

图书在版编目(CIP)数据

C++程序设计基础/周霭如,林伟健编著. —北京:电子工业出版社,2003.8

高等学校计算机基础及应用教材

ISBN 7-5053-9021-X

I. C... II. ①周...②林... III. C语言—程序设计—高等学校—教材 IV. TP312

中国版本图书馆CIP数据核字(2003)第071352号



责任编辑:章海涛

印刷者:北京牛山世兴印刷厂

出版发行:电子工业出版社

北京市海淀区万寿路173信箱 邮编:100036

经 销:各地新华书店

开 本:787×980 1/16 印张:28.25 字数:632千字

印 次:2004年9月第4次印刷

印 数:4000册 定价:33.00元

凡购买电子工业出版社的图书,如有缺损问题,请向购买书店调换。若书店售缺,请与本社发行部联系。联系电话:(010)68279077。质量投诉请发邮件至 zllts@phei.com.cn,盗版侵权举报请发邮件至 dbqq@phei.com.cn。

总 序

当今，很难找出一个不需要应用计算机技术的领域，这意味着几乎所有技术人员都需要学会使用计算机。对计算机了解的深度，不同领域、不同岗位上的工作人员有不同的要求。有些领域要求每个技术人员都对计算机技术有较深入的了解，例如高等学校的电子类专业都开设了大量计算机技术课程。有些领域的大多数工作人员只要对计算机应用有一般性了解就可以了。无论哪一类专业都要开发本领域的专用计算机应用系统，因而需要一大批既懂计算机技术又懂领域业务的技术人才。造就这类人才的途径无非三种：一是设立专门的培养复合人才的专业，由于行业种类、人员层次太多，需求不易量化，难以规划；二是由计算机专业人员深入某个应用领域，学习必要的专业知识并与该领域技术人员合作；三是各非计算机专业领域的技术人员（在学的或在职的）根据工作需要和个人志趣进一步学习有关的计算机技术。由途径二和途径三成长起来的计算机应用人员各有长处，可以互相补充，数量自然会由需求调节，无需行政规划。

编写本系列教材的主要目的是为选择上述第三条途径的人员（首先是高校非计算机专业的学生）提供一套比较系统又比较灵活、实用的学习材料。“系统”是指系列教材的编排从入门开始，循序渐进，涉及的预备知识均包含在教材中。“灵活”是指读者在读完系列教材中的“计算机实用技术”（内容大致相当于教育部提出的计算机基础教育第一层次的要求）、一门“程序设计”语言和“计算机软件与硬件基础”（基本属于第二层次的内容）后，可以根据需要任意选择学习其他课程。“实用”是指根据本系列教材的主要目标，取材注重应用，不追求完备。为了方便教学，每门教材都配有 CAI 课件，有些教材还有配套的习题集或实验指导，供任课教师和读者选用。

从上述关于内容的说明可见，系列教材中的“计算机实用技术”和一门语言的“程序设计”事实上是理工科非计算机专业学生的必修课，其余课程是选修课。

某些非计算机专业对计算机技术训练的要求与计算机应用专业的某些方向的要求交叉，各领域计算机应用系统开发所涉及的技术也有许多相同之处。因此，本系列教材中有一部分也可供计算机专业的相关课程选用。例如，本系列教材中的各种高级语言程序设计的教材，对于选择该语言作为第一门程序设计语言的任何专业都是适用的。又如“计算机组成原理”、“微型计算机接口技术”、“网络技术应用基础”等也可供计算机应用专业选用。

系列教材的作者都是有丰富教学经验的在职一线教师，以计算机系的教师为主，广泛征求相关专业教师的意见并且吸收部分相关专业教师参与编写。尽管经过反复讨论修改，但限于作者水平和其他条件的限制，在总体布局、内容取舍或其他方面一定还存在不足和值得商榷之处，敬请批评指正。

系列教材编委会

2002年11月

前 言

C++ 是优秀的计算机程序设计语言,它的功能相当强大。但学习C++ 比学习 Pascal 和 C 要难得多。我们编写这本书的目的是,为没有任何程序设计基础的理工科大学生提供一本适用教材,以便掌握从理论到实践都要求很高的C++ 语言。

为此,我们根据多年从事计算机程序设计教学的经验,按照学生学习的认知规律,精心构造整本教材体系和叙述方式。原则是循序渐进、难点分散、通俗而不肤浅。

指针是C++ 的重要概念,是操作对象的有力工具。本书没有一般 C 或C++ 教材中专门的“指针”内容的安排。我们从最简单的变量开始,建立对象的名和地址的概念,用对象的不同访问方式贯穿于各章节。从结构化程序设计到面向对象程序设计,采取了比较平滑的过渡。首先,一开始在介绍基本数据类型、程序流程控制、函数等结构化程序设计基本知识时,就非正式地使用“对象”这个术语(从计算机程序的角度,任何占有存储空间的实体都是对象);继而把“结构”和“类”合成一章,掌握结构到类的演变,给出对象的准确定义;进一步,展开介绍面向对象程序的几个基本特性:封装、继承、多态和类属在C++ 中的实现方法。同时,我们在本书的阐述中体现一个思想:没有一种对所有问题都是最好的程序设计方法,对特定问题,选择合适的解决方案是程序员必备的素质。

本书之所以取名为《C++ 程序设计基础》,一是因为它不是一本C++ 语言手册,不可能包罗所有语法规则和特定版本提供的各种功能,二是没有涉及复杂的算法和工程化的面向对象分析设计方法。这两个问题与教材的定位相关。对第一个问题,我们认为学生在掌握了程序设计的基本概念和基本方法之后,可以通过语言平台(如 Visual C++)或者其他资料学习,拓广对语言功能的了解。我们在有关章节中也做了类似的引导,如 STL 标准类库的介绍。那些内容提供给教师选择或学生自学。至于第二个问题,是计算机专业后续课程,例如,数据结构、算法分析、软件工程等课程研究的内容。本书介绍的程序设计方法和使用到的算法都是立足于基本概念和方法,所以通常是简单和小规模的。本课程的教学目标是通过学习C++ 语言,使学生掌握结构化程序设计和面向对象程序设计的基本概念和方法,能够编写一些小型的C++ 程序,具备学习后续课程的程序设计基础和进一步自学的能力。

本书共分 11 章。第 1 章基本数据与表达式,第 2 章程序控制结构,第 3 章函数,第 4 章数组,第 5 章类与对象,第 6 章运算符重载,第 7 章继承,第 8 章虚函数与多态性,第 9 章模板,第 10 章输入输出流,第 11 章异常处理。

本书所有例程在 Visual C++ 6.0 上通过。本书有配套使用的实验指导与习题解答,还制作了本教材的课堂教学用 PowerPoint 幻灯片,以及远程教学用的网络课件(有需要者,请

与华南理工大学电教中心联系)。

本书的第1章、第4章~第11章由周霭如编写,第2章和第3章由林伟健编写。全书习题由林伟健配置。

本书可以作为高等学校计算机专业、信息类、电类专业本科生高级语言程序设计课程教材,也可以作为教师、学生和C++语言爱好者的参考书。

本书编写中,我们参考了大量有关C++书籍和资料,在此对有关作者、译者表示感谢。工作过程中,我们得到华南理工大学教务处和计算机学院领导的大力支持,对此一并感谢。

由于作者的水平所限,成书仓促,难免有错漏之处。敬请各位读者不吝赐教。

编著者

2003年6月

目 录

第 1 章 基本数据与表达式	1
1.1 概述	2
1.1.1 程序设计与程序设计语言	2
1.1.2 一个简单的 C++ 程序	3
1.1.3 程序的编译执行	6
1.2 C++ 的字符集与词汇	7
1.2.1 字符集	8
1.2.2 词汇	8
1.3 C++ 的基本数据类型与存储形式	9
1.3.1 C++ 的数据类型	10
1.3.2 数据存储	10
1.3.3 基本类型	11
1.4 常量与变量	14
1.4.1 常量	14
1.4.2 变量	15
1.5 内存访问	16
1.5.1 名访问	16
1.5.2 地址访问	17
1.5.3 指针变量与间址访问	18
1.5.4 引用	22
1.5.5 const 约束访问	23
1.6 表达式	25
1.6.1 算术表达式	27
1.6.2 关系表达式	31
1.6.3 逻辑表达式	31
1.6.4 赋值表达式	32
1.6.5 条件表达式	34
1.6.6 逗号表达式	35

1.7	数据输入和输出	36
1.7.1	键盘输入	36
1.7.2	屏幕输出	37
1.7.3	表达式语句	38
小结	39
习题 1	39
第 2 章	程序控制结构	45
2.1	选择控制	46
2.1.1	if 语句	46
2.1.2	switch 语句	52
2.2	循环控制	58
2.2.1	while 语句	59
2.2.2	do-while 语句	62
2.2.3	for 语句	67
2.2.4	循环的嵌套	72
2.3	判断表达式的使用	75
2.4	转向语句	78
小结	81
习题 2	81
第 3 章	函数	87
3.1	函数的定义与调用	88
3.1.1	函数定义	88
3.1.2	函数调用	89
3.1.3	函数原型	90
3.2	函数参数的传递	92
3.2.1	传值参数	92
3.2.2	指针参数	97
3.2.3	引用参数	99
3.2.4	函数的返回类型	102
3.3	函数调用机制	106
3.3.1	嵌套调用	106
3.3.2	递归调用	108
3.4	函数指针	113
3.4.1	函数的地址	113

3.4.2	函数指针	115
3.5	内联函数和重载函数	120
3.5.1	内联函数	120
3.5.2	函数重载	121
3.6	变量存储特性与标识符作用域	122
3.6.1	存储特性	123
3.6.2	标识符作用域	124
3.7	多文件程序结构	127
3.7.1	多文件结构	128
3.7.2	预处理指令	129
3.7.3	多文件程序使用全局变量	134
3.8	终止程序执行	135
小结	137
习题 3	138
第 4 章	数组	145
4.1	一维数组	146
4.1.1	一维数组定义与初始化	146
4.1.2	一维数组访问	149
4.2	指针数组	152
4.2.1	指向基本数据类型的指针数组	152
4.2.2	指向数组的指针数组	153
4.2.3	指向函数的指针数组	154
4.3	二维数组	156
4.3.1	二维数组定义与初始化	156
4.3.2	二维数组访问	158
4.4	数组作函数参数	161
4.4.1	向函数传送数组元素	161
4.4.2	数组名作函数参数	163
4.4.3	应用举例	165
4.5	动态存储	170
4.5.1	new 和 delete 操作符	170
4.5.2	动态存储的应用	171
4.6	字符数组与字符串	173
4.6.1	字符串存储	173

4.6.2	字符串的访问	176
4.6.3	字符串处理函数	180
4.7	string 类	184
4.7.1	string 的特性	184
4.7.2	string 的操作	186
小结	191
习题 4	192
第 5 章	类与对象	199
5.1	结构	200
5.1.1	定义结构和访问结构	200
5.1.2	链表	204
5.2	类和对象	214
5.2.1	定义类和对象	215
5.2.2	访问对象成员	218
5.2.3	this 指针	219
5.3	构造函数和析构函数	220
5.3.1	简单构造函数和析构函数	221
5.3.2	带参数的构造函数	222
5.3.3	重载构造函数	226
5.3.4	复制构造函数	227
5.4	静态成员	233
5.4.1	静态数据成员	233
5.4.2	静态成员函数	234
5.5	友员	237
5.5.1	友员函数	237
5.5.2	友员类	239
小结	241
习题 5	241
第 6 章	运算符重载	249
6.1	运算符重载规则	250
6.1.1	重载运算符的限制	250
6.1.2	运算符重载的语法形式	250
6.2	用成员或友员函数重载运算符	253
6.2.1	用成员函数重载算符	254

6.2.2	用友员函数重载	256
6.3	几个典型运算符重载	259
6.3.1	重载 ++ 与 --	259
6.3.2	重载赋值运算符	262
6.3.3	重载运算符[]和()	263
6.3.4	重载流插入运算符和流提取运算符	265
6.4	类类型转换	268
6.4.1	构造函数进行类类型转换	268
6.4.2	类型转换函数	270
小结	274
习题6	274
第7章	继承	277
7.1	类之间的关系	278
7.2	基类和派生类	279
7.2.1	访问控制	280
7.2.2	成员覆盖	287
7.2.3	派生类中的静态成员	287
7.3	基类的初始化	288
7.4	继承的应用实例	291
7.5	多继承	295
7.5.1	多继承的派生类构造和访问	297
7.5.2	虚基类	300
小结	305
习题7	305
第8章	虚函数与多态性	309
8.1	静态联编	310
8.2	类指针的关系	311
8.2.1	基类指针引用派生类对象	312
8.2.2	派生类指针引用基类对象	313
8.3	虚函数和动态联编	316
8.3.1	虚函数和基类指针	316
8.3.2	虚函数的重载特性	320
8.3.3	虚析构函数	322
8.4	纯虚函数和抽象类	324

8.4.1	纯虚函数	324
8.4.2	抽象类	326
8.5	虚函数和多态性的应用	329
8.5.1	一个实例	329
8.5.2	异质链表	336
小结	339
习题 8	340
第 9 章	模板	343
9.1	什么是模板	344
9.2	函数模板	344
9.2.1	模板说明	345
9.2.2	函数模板与模板函数	345
9.2.3	重载函数模板	348
9.3	类模板	350
9.3.1	类模板与模板类	350
9.3.2	类模板作为函数参数	354
9.3.3	在类层次中的类模板	355
9.4	标准模板	357
9.4.1	容器	358
9.4.2	迭代器	363
9.4.3	算法	367
9.5	名空间	372
9.5.1	定义名空间	372
9.5.2	使用名空间	374
小结	376
习题 9	377
第 10 章	输入/输出流	381
10.1	流类和流对象	382
10.1.1	流类库	382
10.1.2	头文件	383
10.2	标准流和流操作	384
10.2.1	标准流	384
10.2.2	输入流操作	385
10.2.3	输出流操作	387

10.2.4	流错误状态	388
10.3	格式控制	390
10.3.1	设置标志字	390
10.3.2	格式控制符	394
10.4	串流	397
10.5	文件处理	399
10.5.1	文件和流	400
10.5.2	打开和关闭文件	401
10.5.3	文本文件	402
10.5.4	二进制文件	407
小结	416
习题 10	416
第 11 章	异常处理	421
11.1	C++ 的异常处理机制	422
11.2	异常处理的实现	423
11.2.1	异常处理程序	423
11.2.2	带异常说明的函数原型	427
11.2.3	再抛出传递	428
11.2.4	创建对象的异常处理	429
小结	430
习题 11	431
附录 A	C++ 关键字	433
附录 B	ASCII 码字符集	434

第 1 章

基本数据与表达式

程序设计语言是人与计算机交流的工具。C++ 由于功能强大、使用灵活，是目前工程中应用比较广泛的一种高级程序设计语言。本章介绍高级程序设计语言的基本概念、C++ 语言的基本语法单位以及表达式运算。

1.1 概述

C++ 语言源于 C 语言。C 语言诞生于 20 世纪 70 年代，最初设计的目的是编写操作系统。因为 C 语言规则简单，不但具有高级语言的数据表示、运算功能，而且可以直接对内存操作，程序运行效率高。基于以上优点，C 语言很快成为世界流行的程序设计语言。

然而，人们要求计算机解决的问题越来越多。C 语言在处理大问题、复杂问题时表现出来的弱点也越来越明显，如缺乏数据类型检查机制、代码重用性差等。

20 世纪 80 年代，美国 AT&T 贝尔实验室对 C 语言扩充改版，成为 C++。C++ 保持了 C 原有的高效、简洁的特点，强化了数据的类型检查和语句的结构性，增加了面向对象程序设计的支持。由于 C++ 的灵活性、良好的继承性和前瞻性，许多软件公司都为 C++ 设计编译系统，提供不同级别的应用类库以及方便实用的开发环境，使之得到广泛应用。

1.1.1 程序设计与程序设计语言

在人类社会生活中，“语言”是人与人之间用来表达意思、交流思想的工具，是由语音、词汇和语法构成的一定系统。人类的思维、感情相当丰富，所以语言系统非常复杂。甚至同一个词、句子，在不同的环境、以不同的语气表达，都可能解释成完全不同的意思。

“程序设计语言”是人与计算机交流的工具，是由字、词和语法规则构成的指令系统。一种高级程序设计语言往往有一百几十条词汇、若干条规则。

高级语言提供了常用的数据描述和对数据操作的规则描述。这些规则是“脱机”的，程序员只需要专注于问题的求解，不必关心机器内部结构和实现。我们说的“程序设计语言”通常是指高级语言。

计算机对问题的求解方式通常可以用数学模型抽象。随着社会科学的发展，人们要求计算机处理的问题越来越复杂，计算机工作者不断寻求简捷可靠的软件开发方法。从过程化、结构化到近代出现的面向对象程序设计，体现了程序设计理论、方法的不断发展。

用高级语言编写的程序称为“源程序”。计算机不能直接识别源程序，必须翻译成二进制代码才能在机器上运行。一旦编译成功，目标程序就可以反复高速执行。

程序设计是根据特定的问题，使用某种程序设计语言，设计出计算机执行的指令序列。程序设计是一项创造性的工作，根据任务主要完成两方面工作：

(1) 数据描述

数据描述是把被处理的信息描述成计算机可以接受的数据形式，如整数、实数、字符、数组等。

信息可以用人工或自动化装置进行记录、解释和处理。使用计算机进行信息处理时，

这些信息必须转换成可以被机器识别的“数据”，如数字、文字、图形、声音等。不管什么数据，计算机都以二进制形式存储和处理。数据是信息的载体，信息依靠数据来表达。

有一些数据，程序设计语言可以直接用“数据类型”描述，如数值、字符。还有一些数据，一般的程序设计语言没有提供直接定义，但可以作为外部文件使用。例如，C++可以在界面或程序代码中使用图形、声音文件，语言本身也提供丰富的多媒体数据处理方法。

(2) 数据的处理

数据处理是指对数据进行输入、输出、整理、计算、存储、维护等一系列的活动。数据处理的目的是为了提取所需的数据成分，获得我们有用的资料。

在结构化程序设计中，数据的描述和处理是分离的。利用面向对象方法，程序把数据和处理进行封装。按照人们习惯的思维模式和软件重用原则，对象还具有继承、多态等特性。每种程序设计方法都有自己的一套理论框架，以及相应的设计、分析、建模方法，但都有各自的优缺点。采用什么方法设计程序，应该依据问题的性质、规模和特点来选择。世界上没有一种能解决所有问题的最优方法。

学习C++语言，不但要掌握一种实用的计算机软件设计工具，而且要通过该课程的学习，掌握计算机程序设计语言的基本语法规则，掌握结构化程序设计和面向对象程序设计的基本方法，为进一步学习和应用打下良好基础。

1.1.2 一个简单的C++程序

问题：输入圆的半径，求圆的周长和面积。

【例 1-1】方法一：用结构化方法编程。

数据描述：半径、周长、面积均用实型数表示

数据处理：

- 输入半径 r 。
- 计算周长 $= 2 * \pi * r$ 。
- 计算面积 $= \pi * r^2$ 。
- 输出半径、周长、面积。

我们可以编写程序如下：

```
#include<iostream.h>
void main()
{ float r, girth, area;           //声明数据
  const float pi = 3.1415;
  cout << "Please input radius: \n";
  cin >> r;                       //输入半径
```