

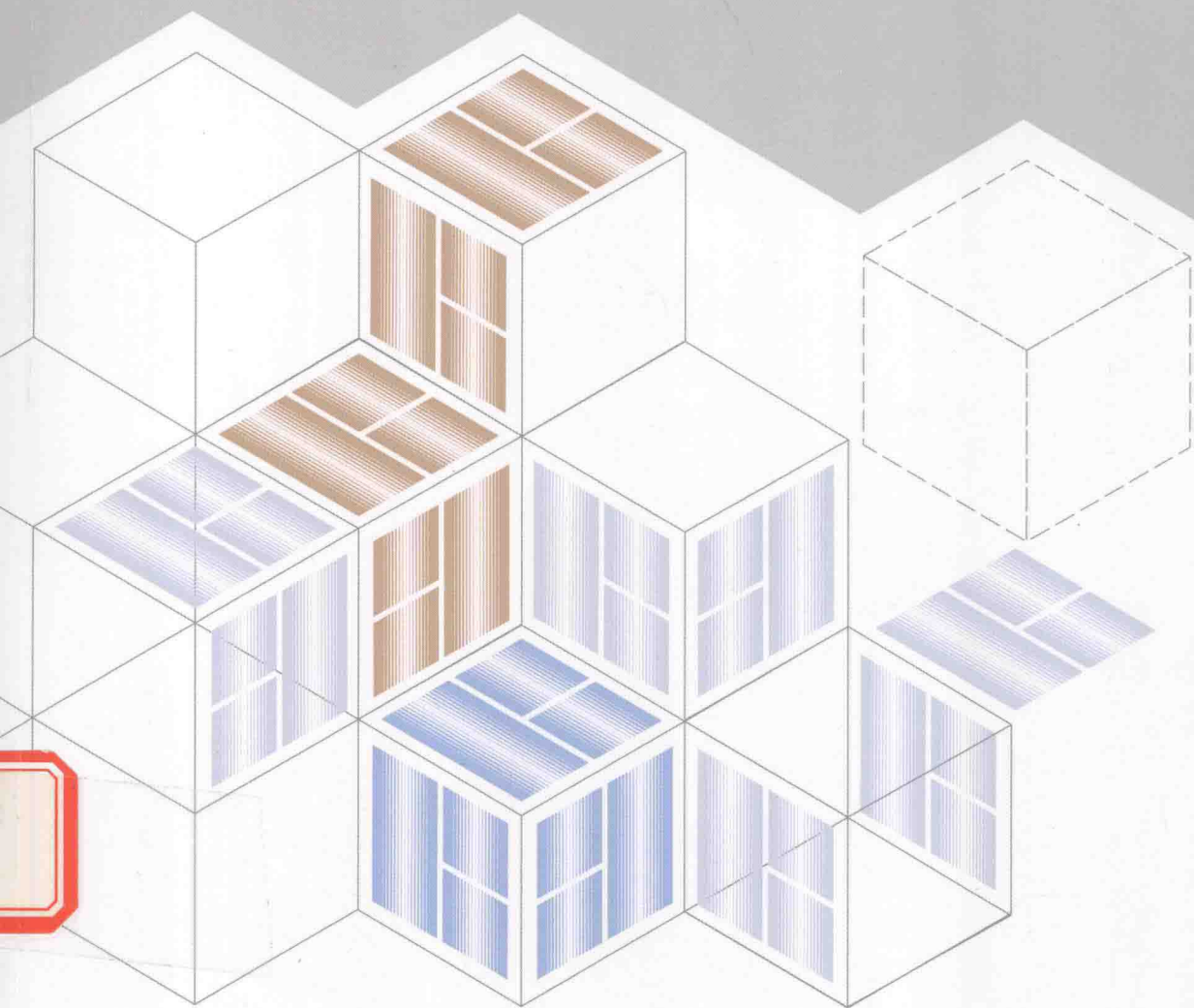


合肥学院模块化教学改革系列教材

C语言 程序设计

C Programming

项响琴 主编



中国科学技术大学出版社

合肥学院模块化教学改革系列教材

C语言 程序设计

C Programming

项响琴 主编



中国科学技术大学出版社

内 容 简 介

本书是一本以应用为根本,以能力培养为主线,以素质教育为目标的实用型C语言程序设计模块化教材。全书体例创新性地由五大模块(共10章)组成,内容包括:数据的输入输出、计算处理、软件结构、结构化数据处理和工程实训。

本书以一个关联案例“学生成绩管理系统”的开发贯穿前四大模块。每章通过对该系统中若干功能的分析,由浅入深,逐步提出问题,引出知识点,并设置相关的例题阐释知识点的运用,另配有若干实训项目来强化知识点应用、算法设计与编程技巧,并给出关联案例的代码实现和常见错误分析。各章都配有自主学习任务和习题,培养读者的自主学习能力、查阅资料和归纳总结能力。本书在工程实训模块中精心设置了传统DOS界面、图形界面、软硬件结合三个层面的系统开发实例,充分体现其培养应用能力的特色。

本书适合作为应用型高等学校各专业的C语言程序设计课程教材,也适合作为软件开发人员、计算机爱好者及各类自学人员的参考用书。

图书在版编目(CIP)数据

C语言程序设计/项响琴主编. —合肥:中国科学技术大学出版社,2016.8
(合肥学院模块化教学改革系列教材)
ISBN 978-7-312-03840-2

I. C… II. 项… III. C语言—程序设计—教材 IV. TP312

中国版本图书馆CIP数据核字(2015)第186829号

出版 中国科学技术大学出版社
安徽省合肥市金寨路96号,230026
<http://press.ustc.edu.cn>
印刷 安徽省瑞隆印务有限公司
发行 中国科学技术大学出版社
经销 全国新华书店
开本 787 mm×1092 mm 1/16
印张 20.5
字数 524 千
版次 2016年8月第1版
印次 2016年8月第1次印刷
定价 40.00 元

总 序

课程是高校应用型人才培养的核心,教材是高校课程教学的主要载体,承载着人才培养的教学内容,而教学内容的选择关乎人才培养的质量。编写优秀的教材是应用型人才培养过程中的重要环节。一直以来,我国普通高校教材所承载的教学内容多以学科知识发展的内在逻辑为标准,与课程相对应的知识在学科范围内不断地生长分化。高校教材的编排是按照学科发展的知识并因循其发展逻辑进行的,再由教师依序系统地教给学生。

当我们转变观念——大学的学习应以学生为中心,那我们势必会关注“学生通过大学阶段的学习能够做什么”,我们势必会考虑“哪些能力是学生通过学习应该获得的”,而不是“哪些内容是教师要讲授的”,高校教材承载的教学内容及其构成形式随即发生了变化,突破学科知识体系定势,对原有知识按照学生的需求和应获得的能力进行重构,才能符合应用型人才培养的目标。合肥学院借鉴了德国经验,实施的一系列教育教学改革,特别是课程改革都是以学生的“学”为中心的,围绕课程改革在教材建设方面也做了一些积极的探索。

合肥学院与德国应用科学大学有近30年的合作历史。1985年,安徽省人民政府和德国下萨克森州政府签署了“按照德国应用科学大学办学模式,共建一所示范性应用型本科院校”的协议,合肥学院(原合肥联合大学)成为德方在中国最早重点援建的两所示范性应用科学大学之一。目前,我校是中德在应用型高等教育领域里合作交流规模最大、合作程度最深的高校。在长期合作的过程中,我校借鉴了德国应用科学大学的经验,将德国经验本土化,为我国的应用型人才培养模式改革做出了积极的贡献。在前期工作的基础上,我校深入研究欧洲,特别是德国在高等教育领域的改革和发展状况,结合博洛尼亚进程中的课程改革理念,根据我国国情和高等教育的实际,开展模块化课程改革。我们通过校企深度合作,通过大量的行业、企业调研,了解社会、行业、企业对人才的需求以及专业对应的岗位群,岗位所需要的知识、能力、素质,在此基础上制订人才培养方案和选择确定教学内容,并及时实行动态调整,吸收最新的行业前沿知识,解决人才培养和社会需求适应度不高的问题。2014年,合肥学院“突破学科定势,打造模块化课程,重构能力导向的应用型人才培养教学体系”获得了国家教学成果一等奖。

为了配合模块化课程改革,合肥学院积极组织模块化系列教材的编写工作。以实施模块化教学改革的专业为单位,教材在内容设计上突出应用型人才能力

的培养。即将出版的这套丛书此前作为讲义,已在我校试用多年,并经过多次修改。教材明确定位于应用型人才的培养目标,其内容体现了模块化课程改革的成果,具有以下主要特点:

(1) 适合应用型人才培养。改“知识输入导向”为“知识输出导向”,改“哪些内容是教师要讲授的”为“哪些能力是学生通过学习应该获得的”,根据应用型人才的培养目标,突破学科知识体系定势,对原有知识、能力、要素进行重构,以期符合应用型人才培养目标。

(2) 强化学生能力培养。模块化系列教材坚持以能力为导向,改“知识逻辑体系”为“技术逻辑体系”,优化和整合课程内容,降低教学内容的重复性;专业课注重理论联系实际,重视实践教学和学生能力培养。

(3) 有利于学生个性化学习。模块化系列教材所属的模块具有灵活性和可拆分性的特点,学生可以根据自己的兴趣、爱好以及需要,选择不同模块进行学习。

(4) 有利于资源共享。在模块化教学体系中,要建立“模块池”,模块池是所有模块的集合地,可以供应用型本科高校选修学习,模块化教材很好地反映了这一点。模块化系列教材是我校模块化课程改革思想的体现,出版的目的一是与同行共同探索应用型本科高校课程、教材的改革,努力实现资源共享。

(5) 突出学生的“学”。模块化系列教材既有课程体系改革,也有教学方法、考试方法改革,还有学分计算方法改革。其中,学分计算方法采用欧洲的“work-load”(即“学习负荷”,学生必须投入 28 小时学习,并通过考核才可获得 1 学分),这既包括对教师授课量的考核,又包括对学生自主学习量的考核,在关注教师“教”的同时,更加关注学生的“学”,促进了“教”和“学”的统一。

围绕着模块化教学改革进行的教材建设,是我校十几年来教育教学改革大胆实践的成果,广大教师为此付出了很多的心血。在模块化系列教材付梓之时,我要感谢参与编写教材以及参与改革的全体老师,感谢他们在教材编写和学校教学改革中的付出与贡献!同时感谢中国科学技术大学出版社为系列教材的出版提供了服务和平台!希望更多的老师能参与到教材编写中,更好地展现我校的教学改革成果。

应用型人才培养的课程改革任重而道远,模块化系列教材的出版,是我们深化课程改革迈出的又一步,由于编者水平有限,书中还存在不足,希望专家、学者和同行们多提意见,提高教材的质量,以飨莘莘学子!

是为序。

合肥学院党委书记 蔡敬民
2016 年 7 月 28 日于合肥学院

前 言

教育部《工程教育专业认证标准(试行)》规定,高校毕业生必须达到知识、能力与素质的基本要求。“C 语言程序设计”作为计算机学科的基础课程,所涉及的能力培养目标具体包括:① 扎实掌握 C 语言基本语法的能力;② 面向过程的结构化程序设计能力;③ 数据、流程的表达能力;④ C 标准库函数的查找和应用能力;⑤ 算法设计与程序调试能力;⑥ 多文件多模块的构建能力。

“C 语言程序设计”是计算机学科的核心课程之一,其前导课程有“计算机专业导论”,后续课程有“数据结构与算法”“Arduino 程序开发”“单片机原理与应用”等。由此可见,C 语言在计算机学科中占有重要地位。本书编写的指导原则是:以知识点模块化构建为出发点,以应用为根本,以能力培养为主线,以素质教育为目标。本书紧扣程序设计三大基本动作——数据的输入、输出和计算处理,结合编者的模块化教学改革成果,分为数据的输入输出、计算处理、软件结构、结构化数据处理和工程实训五大模块,以一个关联案例——“学生成绩管理系统”的开发贯穿前四个模块,通过对该系统各功能的分析、设计与实现,由浅入深,逐步完成 C 语言程序设计学习者知识学习、能力和素质培养的过程。

工程实训模块的设置是本书“应用”特色的体现,力图培养读者综合应用、软件开发和自主学习的能力,让读者从实训中体会 C 语言无处不在、经久不衰的魅力。实训项目给出了 C 语言的三个典型应用层面:学生成绩管理系统开发、图形界面的贪吃蛇游戏系统开发和 SOS 信号发射装置系统开发,在保证不同层次读者都能理解的同时,引导读者享受学习 C 语言的无穷乐趣,让读者在自主、实践、应用和协作创新诸方面得到综合性的锻炼。

在内容设置上,本书结合“专业认证”要求,给出每个模块的知识简介、学习目标和能力及素质培养的总体要求。每个模块由若干章节构成。有别于传统教材“先理论再实践”的编写套路,本书力图做到“学以致用,实用为主”,各章以“关联案例分析—问题提出—知识点介绍—例题讲解—实训项目的设计—关联案例代码实现—常见错误分析”为主线,将 C 语言能解决什么问题、需要哪些知识、如何解决问题这一过程脉络清晰地呈现在读者面前。同时,每一章都配有自主学习内容与任务(以“★”号标出),引导读者运用现代信息技术查阅资料和归纳总结,培养读者的自主学习能力。

全书例题、实训项目内容的选取兼顾趣味性和实用性。习题以巩固基本知识点、检测自主学习内容、强化编程思维、培养规范编程能力、加强综合应用能力

为目的,从模仿到创造,遵循程序设计语言学习者的科学认知规律。

在语言组织上,本书注重文字的易读性,语言简明,案例生动,算法流程图表达清晰,程序代码规范统一,程序结果配有截图演示和细致分析,由浅入深、分层次地介绍知识模块。全书少讲空道理,多用实范例,力图激发读者对编程的兴趣。

本书的特色如下:

- 在体例上,将传统章节知识内容进行模块化构建,更有利于学习者能力和素质的培养。
- 以能力培养为主线,每章以关联案例的分析引出知识点,再通过知识点的学习,完成一定量的实训项目及关联案例代码的编写。
- 以应用为根本,在工程实训模块中精心设置了传统 DOS 界面、图形界面、软硬件结合三个层面的系统开发示例,充分突出 C 语言编程特色的同时,为后续课程的学习打下良好基础。
- 创新性地将自主学习内容与任务列为“标准配置”,引导读者运用现代信息技术查阅资料和归纳总结,培养读者自主学习、终身学习的良好习惯。
- 全书编写的指导思想是:不囿于 C 语言课程本身,而是以 C 语言在整个计算机学科本科教育中的“有机课程”作用为根本,注意与其他课程的衔接与拓展。

本书由项响琴主编,王昆仑主审。第 1 模块由项响琴、檀明、李国斌编写,第 2 模块由章义刚、李国斌编写,第 3 模块由华珊珊、李祎、徐静编写,第 4 模块由周艳玲、项响琴编写,第 5 模块由项响琴、何立新、高玲玲编写。

“书无完书”,因编者水平有限,本书不足之处在所难免。热忱欢迎广大读者给作者发送邮件(xxq_03@163.com),对本书提出意见和建议,我们会在再版时予以修订。

编 者

2016 年 4 月

目 录

总序	(i)
前言	(iii)
导读	(1)

第1模块 数据的输入输出

第1章 基本输入输出操作	(4)
1.1 格式化输入输出操作	(5)
1.1.1 格式化输出操作	(6)
1.1.2 数值型数据类型	(9)
1.1.3 格式化输入操作	(12)
1.2 字符输入输出操作	(14)
1.2.1 字符型数据类型	(14)
1.2.2 字符型数据输入输出操作	(15)
1.3 特殊常量*	(18)
1.4 实训项目	(19)
1.5 通常易犯的错误	(22)
1.6 自主学习任务	(23)
习题1	(23)
第2章 文件读写操作	(25)
2.1 地址(Address)与指针(Pointer)	(26)
2.1.1 变量与地址	(26)
2.1.2 指针变量	(28)
2.2 文件的打开与关闭	(32)
2.2.1 文件打开	(33)
2.2.2 文件关闭	(34)
2.3 文件读写操作	(34)
2.3.1 文件格式化读写操作	(34)
2.3.2 文件字符读写操作	(37)
2.4 文件的其他操作	(38)
2.4.1 文件结束判断操作	(38)
2.4.2 文件随机读写操作*	(39)

2.5 实训项目	(41)
2.6 通常易犯的错误	(45)
2.7 自主学习任务	(46)
习题 2	(46)

第2模块 计算处理

第3章 运算符及表达式	(48)
3.1 赋值运算符及类型转换	(49)
3.1.1 赋值运算符	(49)
3.1.2 类型转换	(49)
3.2 算术运算符	(52)
3.2.1 常用算术运算符	(52)
3.2.2 自增自减运算符	(54)
3.3 关系运算符	(55)
3.4 逻辑运算符	(57)
3.5 条件运算符*	(58)
3.6 逗号运算符*	(59)
3.7 运算符优先级和结合方向	(60)
3.8 实训项目	(60)
3.9 通常易犯的错误	(64)
3.10 自主学习任务	(64)
习题 3	(65)

第3模块 软件结构

第4章 程序结构	(68)
4.1 算法*	(68)
4.2 顺序结构	(69)
4.3 选择结构	(70)
4.3.1 单分支选择结构	(70)
4.3.2 双分支选择结构	(73)
4.3.3 多分支选择结构	(76)
4.4 循环结构	(85)
4.4.1 for 语句	(86)
4.4.2 while 语句	(89)
4.4.3 do-while 语句	(91)
4.4.4 其他语句	(92)
4.4.5 循环规律的发现	(96)
4.5 实训项目	(99)

4.6 通常易犯的错误	(106)
4.7 自主学习任务	(107)
习题 4	(108)
第 5 章 文件结构	(112)
5.1 结构化程序设计(Modular Programming)	(112)
5.2 函数(Function)	(114)
5.2.1 函数定义	(115)
5.2.2 函数调用	(117)
5.2.3 函数的特殊调用	(123)
5.3 多文件结构	(128)
5.4 变量的作用域	(132)
5.4.1 局部变量	(132)
5.4.2 全局变量	(133)
5.4.3 静态变量*	(134)
5.4.4 外部变量*	(135)
5.5 实训项目	(137)
5.6 通常易犯的错误	(143)
5.7 自主学习任务	(144)
习题 5	(145)

第 4 模块 结构化数据处理

第 6 章 数组(Array)	(149)
6.1 一维数组	(150)
6.1.1 一维数值型数组	(150)
6.1.2 一维字符型数组(字符串)	(154)
6.1.3 指向一维数组的指针	(158)
6.1.4 一维数组与函数	(162)
6.1.5 一维数组相关算法	(166)
6.2 二维数组	(171)
6.2.1 二维数值型数组	(171)
6.2.2 二维字符型数组	(174)
6.2.3 指向二维数组的指针*	(176)
6.3 实训项目	(178)
6.4 通常易犯的错误	(187)
6.5 自主学习任务	(188)
习题 6	(189)
第 7 章 结构体	(193)
7.1 结构体类型	(194)

7.2 结构体变量	(196)
7.2.1 结构体变量定义	(196)
7.2.2 结构体变量初始化	(197)
7.3 结构体数组	(201)
7.3.1 静态初始化	(202)
7.3.2 动态初始化	(202)
7.4 结构体与函数	(205)
7.5 文件块读写操作	(208)
7.6 链表	(209)
7.6.1 动态数据结构	(209)
7.6.2 单向链表创建与遍历	(209)
7.6.3 单向链表插入与删除	(213)
7.7 枚举类型*	(218)
7.8 实训项目	(220)
7.9 通常易犯的错误	(228)
7.10 自主学习任务	(229)
习题 7	(230)

第5模块 工程实训

第8章 学生成绩管理系统开发	(233)
8.1 系统任务要求	(233)
8.2 系统功能介绍	(234)
8.2.1 学生对象描述	(234)
8.2.2 预处理工作	(234)
8.3 系统功能实现	(235)
8.3.1 系统主函数的实现	(235)
8.3.2 其他功能函数的实现	(236)
8.4 自主学习任务	(253)
第9章 图形化界面的“贪吃蛇”游戏系统开发	(254)
9.1 系统任务要求	(254)
9.2 系统相关实例分析	(255)
9.3 系统的设计与实现*	(264)
9.3.1 流程图的设计	(264)
9.3.2 预处理工作	(266)
9.3.3 系统主函数	(267)
9.3.4 其他功能函数的实现	(268)
习题 9	(285)

第 10 章 SOS 信号发射装置系统开发	(287)
10.1 系统任务要求	(287)
10.2 硬件系统设计	(287)
10.3 环境搭建及使用说明	(288)
10.4 编程框架及库函数介绍	(289)
10.5 SOS 信号发射装置系统开发*	(297)
10.5.1 密电码介绍	(297)
10.5.2 系统流程图	(297)
10.5.3 主要算法及关键代码	(299)
10.5.4 常见问题	(300)
习题 10	(301)
附录 1 C 关键字	(302)
附录 2 常用字符 ASCII 码	(303)
附录 3 常用的 C 语言库函数	(305)
附录 4 C 运算符的优先级与结合性	(309)
附录 5 硬件基础知识	(310)
参考文献	(312)

导 读

“C 语言程序设计”是一门实践性很强的课程。对 C 语言的学习,读者不仅要掌握其语法理论知识,更应重视知识的应用,完成一定量的编程任务,在实践中掌握程序设计语言的学习方法,培养知识的综合应用能力。这样,在后续学习中就能够以 C 语言为根基,举一反三,轻松学习其他程序设计语言及相关专业课程了。

本书编写的指导思想是“学以致用”。开篇部分不再介绍有关程序设计语言的发展史与 C 语言的应用等,有关这方面的内容要求读者自行学习,同时完成相关自主学习任务,目的是培养读者运用网络与图书馆等现有资源的自主学习能力。

本书结合编者模块化教学改革成果,将“学生成绩管理系统”的开发贯穿始终,分为数据的输入输出、计算处理、软件结构、结构化数据处理和工程实训五大模块。各模块以应用为根本,以能力培养为主线,知识点由浅入深地嵌入到案例中。例如,“数据的输入输出”模块,介绍数据的基本输入输出操作和文件读写操作,最后将这些知识应用到“学生成绩管理系统”中,完成该系统中信息输入输出的功能;“计算处理”模块,介绍运算符及其表达式,运用算术运算符、关系运算符等,实现该系统中数据(成绩)的计算处理功能;最后的“工程实训”模块,将 C 语言的基础知识进行综合应用,实现该系统全部功能模块的功能,达到本书的学习目标。读者通过对本书几个模块的学习,可以培养理论知识的综合应用能力以及动手参与开发软件系统来解决实际问题的能力。

对于实践性很强的 C 语言课程,“纸上谈兵”的学习方法是行不通的。读者首先要准备一台 PC 机并安装 C 语言运行环境,本书所有程序都是在 VC++ 6.0 环境下编译、运行的。

有了安装 C 语言运行环境的 PC 机,就可以进入本书的学习阶段了。前四大模块是 C 语言的基础知识及简单应用,建议分四步进行:

第一步,读懂本书中的程序,学习相关知识点,并在机器上运行程序,以加深对知识点的记忆。这一步学习切忌流于“指法练习”。在输入本书例题的过程中,要有四个方面的收获:一是理论知识的掌握;二是编程思路的理解;三是代码书写风格的掌握;四是运行结果的分析。

第二步,尝试改写书中的程序,加深对知识的理解和掌握。这一步在学习中非常重要,也往往是初学者不愿意做的一步。初学者多有一种“求新”的心态,不愿再去研究别人写过的程序。其实不然,读者只有在深刻理解例题编写思路的基础上有所突破和创新,才能改写出更高效的程序。

第三步,编写程序,体验对知识的应用。完成课后习题,独立完成代码的编写,体会代码编写的规范性,这可以说是学习的初级目标。

第四步,调试程序。完成了程序的编写,还只是处在代码的编辑状态。对于编程人员来说,完成一段程序,重点要掌握程序的“调试”方法,这才是重要的检验环节。水平再高的程序员编写程序,都难免会出现这样那样的错误,初学者与高手的差距在于:初学者常常是左看右看上看下看,就是找不出问题所在,而高手可以通过调试过程,改正一个个错误,让程序臻于完美。

程序的错误有两种:

一种是编译错误,基本上都属于语法错误,在 VC 中出现这类错误时,编译器会给出提示信息,修改起来相对比较轻松。良好的调试习惯是:从第一条提示信息开始排查,因为第一条错误信息最能反映错误的位置和类型,所以调试程序时务必从第一条错误信息开始。另外,在修改后应立即重新编译程序,编译后如果还有错误,再由前向后逐条修改,每修改一处错误应重新编译一次程序,最终一定能“拨云见日”。

另一种是逻辑错误。这类错误在程序代码不太多时,可以通过多加输出语句,检查中间值来排错。但当程序规模较大时,就要求读者学会使用 VC 中提供的 Debug 功能,通过设置断点,进行单步调试,观察每一步变量值的变化来排查错误。关于这部分内容,读者可以在网站上获取学习资料,当然主要是实践操作,在实践中提高程序调试的能力。

本书第 5 模块以实践为主,重点在于综合运用所学知识完成小型应用软件系统的开发,同时补充了有关界面处理、键盘控制及软硬件结合等相关知识内容。建议读者在学完第 1 模块后,就开始进行第 5 模块的学习。关于三个软件项目的开发,首先是学习其设计思路,同时查阅相关资料,自主学习相关的库函数及硬件基础知识。

本书结合编者模块化课程改革成果,设置了自主学习内容。读者应正确对待这方面的学习要求,严格完成学习任务,以期养成自主学习和终身学习的良好习惯。

学习一门语言,掌握理论知识是基础,运用学到的理论知识来开发软件是目标,自主学习、善于利用在线帮助是关键!

自主学习任务:

- 了解计算机语言的发展历程。
- 了解 C 语言之父丹尼斯·里奇(Dennis M. Ritchie)。
- 了解经久不衰的 C 语言的发展史及现状。
- 了解 C 语言的运行环境。
- 分析 C 语言的特点及其作为计算机语言的优势与劣势。
- 分析 C 语言的应用领域。

第 1 模块 数据的输入输出

本模块从“学生成绩管理系统”出发,介绍 C 语言程序中数据的输入输出(Input and Output)操作,包括 C 语言程序的数据表达(数据类型)、数据输入输出方式、输入输出库函数的使用和简单程序的编写,主要培养编程人员与计算机之间数据(信息)交换的能力。



知识简介

- 基本数据类型(Data Type):整型(Integer)、浮点型(Floating)、字符型(Character)。
- 常量(Constant)、变量(Variable)、标识符(Identifier)、关键字(Keyword)。
- 指针(Pointer)变量、动态(Dynamic)申请存储空间(Storage Space)。
- 基本输入输出操作库函数的使用:格式化输入输出函数 scanf() 和 printf()、字符输入输出函数 getchar()、putchar()、gets() 和 puts() 等。
- 文件(File)读写操作库函数的使用:格式化读写函数 fscanf() 和 fprintf()、字符读写函数 fgetc()、fputc()、fgets()、fputs() 和其他一些改变文件指针的常用库函数。
- 程序(Program)基本结构:编译预处理、主函数 main()、声明语句(Statement)、执行语句、注释(Comment)。



学习目标

掌握 C 程序的基本结构,掌握头文件包含的预处理指令,掌握三种基本数据类型及其常量和变量的表示与应用,掌握指针变量的声明与使用,掌握 C 程序中数据输入输出操作库函数的查找和使用,掌握简单 C 程序的编写。



能力培养要素

- 运用信息技术获取知识的能力。
- C 程序中数据(信息)的表达能力。
- C 标准库函数的查找和应用能力。
- 规范化代码的编写能力。
- 简单 C 程序的编程能力。

第 1 章 基本输入输出操作

在学生成绩的日常管理中,我们常用到表 1.1 中的学生信息记录。

表 1.1 学生信息表

序号	学号	姓名	性别	年龄	系别	班级	外语	数学	C 语言	总分	平均分	等级
1	1401101	王晓琳	F	20	计科系	计本 1 班	88	90	89	267	89.00	良好
2	1401102	毛光烈	M	19	计科系	计本 1 班	62	66	67	195	65.00	及格
3	1401201	李中立	M	22	计科系	计本 2 班	75	88	69	232	77.33	中等
4	1402101	何田田	F	21	电子系	电子 1 班	96	90	92	278	92.67	优秀
5	1402105	王和平	M	20	电子系	电子 1 班	76	85	90	251	83.67	良好

在信息量较少的情况下,可以运用简单的计算处理来分析学生记录。当数据量较大、处理过程相对复杂时,若想方便高效地处理学生记录,则应选择计算机软件系统来管理学生信息。

现在我们来着手编程实现“学生成绩管理系统”,运用计算机应用软件来管理学生的成绩。那么,首先要考虑的是如何将学生的信息(数据)告诉计算机程序?换言之,计算机应用软件如何获取这些信息?信息传输时如何用程序设计语言来表达这些数据?在计算机内部如何存储和计算这些数据?最终处理结果又是通过何种方式反馈给用户呢?



在 C 语言中,提供了相应的输入输出操作函数,以实现信息的输入输出操作。例如,从外部输入设备(键盘)读取学生信息,可以执行下列操作:

```
(1) scanf("%ld%d",&stu_no,&stu_age);  
    //从键盘读取学号和年龄分别存储于变量 stu_no 和 stu_age 中  
(2) stu_sex = getchar();    //从键盘读取字符("F"或"M")①,赋给变量 stu_sex  
    向外部输出设备(屏幕)写出信息,可以执行下列操作:  
(1) printf("学号:%ld 姓名:%s 年龄:%d\n",stu_no,stu_name,stu_age);  
    //按照格式控制要求,向屏幕写出变量 stu_no、stu_name 和 stu_age 的值  
(2) puts("计科系");    //在屏幕上显示"计科系"字符串
```

本章主要介绍 C 语言程序通过外部设备(键盘和屏幕)进行数据输入输出操作的方法,并尝试编写简单的 C 语言程序,来解决“学生成绩管理系统”中信息输入输出的问题。

下面我们先来看一段 C 语言程序。

```
#include<stdio.h>  
  
int main()
```

^① 由于程序的注释语句也在 VC++6.0 界面显示,本书注释语句中的引号形式与程序语句保持一致。

```
{  
    printf("轻松学习 C 语言! \n");    //在屏幕上输出"轻松学习 C 语言!"  
}
```

这就是一段完整的 C 语言程序代码,编辑(Edit)规范,编译(Compile)、运行(Run)正常,在 VC++6.0 中,程序运行结果如图 1.1 所示。



图 1.1 首个程序运行结果

别小看这个 C 语言程序,它可是“麻雀虽小,五脏俱全”呢!

(1) “#include”是编译预处理命令(Preprocessor Directive)之一的文件包含命令^①,在此处的作用是将写有基本输入输出函数的头文件“stdio.h”包含到本程序中,原因是我们调用了该头文件中的 printf() 输出函数。

(2) int 为函数返回类型(函数值的类型),是本章要介绍的基本数据类型之一。C 语言程序是从 main() 函数的左花括号“{”开始,执行到右花括号“}”结束,在两个括号之间,根据需求完成代码编写,实现对应的功能,这是我们学习 C 语言的核心所在。

(3) printf() 输出函数是本章介绍的格式化输出函数,其作用是按格式控制串中的要求输出相应的数据信息。双斜杠“//”是引出注释内容的,表明从此开始到本行末尾全为注释内容,这部分内容是不被编译和运行的。

(4) 程序运行结果是:输出“轻松学习 C 语言!”后回车换行,最后一行“Press any key to continue”为系统自带的一条输出信息,意为“敲任意键继续”。

 为了增强代码的可读性,要求加上必要的注释,VC++6.0 环境下,注释还有另一种形式:

/* 注释内容 */

这里用“/*”和“*/”括起来的“注释内容”即为注释部分。与前面说的“//”形式的注释形式相比,此种形式的优点是可以分多行编辑。

下面来重点学习 C 语言程序中,如何通过外部设备(键盘和屏幕)输入输出不同形式的数。根据信息读取和存储形式不同,分格式化输入输出和字符输入输出两种情况加以介绍。

1.1 格式化输入输出操作

学生信息表(表 1.1)中出现了由 0~9 组成的数值型数据,如学号 1401201 是长整型常量^②,平均分 89.00 是浮点型常量。在计算处理中,C 语言将定义 long(长整型)变量^③

① 文件包含是预编译的一种,通过 include 包含头文件,程序中就可以调用头文件中的库函数。

② 常量是程序运行过程中不变的量。常量可以直接使用,不用定义;常量是不可寻址的。

③ 变量是程序运行过程中可变的量。变量必须先定义,后使用;通过变量名访问其存储空间。