

高等学校计算机基础教育教材精选

C语言程序设计基础

赵春晓 王丽君 编著



清华大学出版社

本书是根据作者多年从事《C语言程序设计》课程的教学和教材编写工作的经验编写而成的。本书可作为高等院校计算机专业及相关专业的一门基础课程教材，也可供从事计算机工作的工程技术人员参考。

本书共分10章。第1章介绍C语言的发展概况、特点、数据类型、运算符和表达式、控制语句、数组、指针、函数、结构体、共用体和枚举类型。第2章介绍C语言中的字符串处理。第3章介绍C语言中的文件操作。第4章介绍C语言中的预处理。第5章介绍C语言中的编译原理。第6章介绍C语言中的库函数。第7章介绍C语言中的宏定义。第8章介绍C语言中的变量声明。第9章介绍C语言中的变量定义。第10章介绍C语言中的变量使用。

高等学校计算机基础教育教材精选

C语言程序设计基础

赵春晓 王丽君 编著

清华大学出版社
北京

内 容 简 介

计算思维是一种人们对于一个问题描述、建立模型以及问题最后求解这一过程的科学思维能力,它已经成为人们本质的、必须具备的一种基础思维方式和能力。C 语言程序设计基础课程是训练学生思维能力的第一个最直接最重要的平台,其主要作用不仅使学生具有编程的能力,还要培养学生通过计算思维去分析问题和解决问题的能力。本书是一本以计算思维为导向的 C 语言程序设计教材,作者总结多年来讲授 C 语言程序设计课程的实际经验,以培养学生计算思维能力为主线进行教材体系编排和内容组织。在教材内容和教学方法中融入计算思维,从思维方法的高度培养学生,让学生自觉地运用计算思维来看待问题、思考问题和解决问题,从而实现教学由传授知识到培养能力的转变。

本教程的每一章后都有一节实验指导,精选了典型问题让学生去上机实践,方便学生练习,另外每一章后还配有大量习题,方便学生自学,有利于培养学生的实践能力。使学生不会感觉编程很难,会有成就感,从而更加激发学习的热情,有利于提高学生学习的积极性。

本教程的上机教学环境为 Dev-C++,所有例题在 Dev-C++ 中测试通过。Dev-C++ 是一种简单易用、体积小、轻便的可视化集成开发环境,具有关键字加亮,不同颜色显示,代码自动补全等功能,方便高效,是适合学生在学校和家庭安装与使用的自由软件。作者推荐使用 Dev-C++ 作为高校 C 语言的教学和上机实验的工具。

本教程体系编排上采用折中式组织。考虑各个学校计算机和非计算机专业所讲授的内容有所不同,将每一章分成基本内容和阅读延伸两部分。各类专业都需要讲授的基本内容部分,大约需要 64 学时。教师可以根据其专业要求选择讲授阅读延伸这部分内容,教学时数一般在 64~96。这种教程组织方式方便了不同专业的师生使用。

本书可作为高等学校计算机专业和非计算机专业本专科学生学习程序设计课程的教材,也可供程序员和编程爱好者参考使用。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C 语言程序设计基础/赵春晓,王丽君编著. —北京:清华大学出版社,2017

(高等学校计算机基础教育教材精选)

ISBN 978-7-302-48323-6

I. ①C… II. ①赵… ②王… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 216459 号

责任编辑:谢琛 薛阳

封面设计:何凤霞

责任校对:李建庄

责任印制:沈露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦 A 座

邮 编:100084

社总机:010-62770175

邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课件下载: <http://www.tup.com.cn>, 010-62795954

印 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:20

字 数:485 千字

版 次:2017 年 10 月第 1 版

印 次:2017 年 10 月第 1 次印刷

印 数:1~2000

定 价:45.00 元

产品编号:076401-01

前言

C 语言程序设计基础

作者在高校从事了三十年 C 语言程序设计课程的教学,深切地感受到 C 语言程序设计作为一门计算机基础课程,其重要性不仅仅体现在一般意义上的程序编制,更体现在如何培养学生运用计算思维解决问题的能力。在长期的教学实践中,作者遇到的最大问题是学生在上课时基本都能够听懂,可到了自己动手做习题的时候,就会觉得无从下手,不会编程。尤其是面对较复杂的应用问题时,往往束手无策。究其原因,就是不知道程序设计的思想方法,缺乏应有的计算思维能力。所有这一切,从主观上说,与教师的教学方法有关,与学生的学习态度有关;而在客观上,与教材体系编排和教学内容组织有着更大的关系。面对这样一些问题,我们在多年教学经验的基础上,决定编写本教材。

写给使用本教材的教师:

(1) 贯穿于教材中的计算思维主线

计算思维是一种人们对于一个问题描述、建立模型以及问题最后求解这一过程的科学思维能力,它已经成为人们本质的、必须具备的一种基础思维方式和能力。程序设计是将分析和解决问题的思维活动转化成计算机程序的过程。从这个角度讲,程序设计最能够体现问题求解的思想方法,是对学生进行思维训练的一个最直接、最具操作性的平台。学生编程过程中,通过问题分析、算法设计、程序实现来体验问题求解的思维训练过程。所以,学生学 C 编程的目的不应是单纯掌握 C 语言的语法规则,不是单纯学习 C 语言程序设计技术,而是学习利用计算机分析和解决问题的基本过程和思维方法。将计算思维融入程序设计教材中有助于提升学生的编程水平。从思维方法的高度培养学生,可以让学生自觉地运用计算思维来看待问题、思考问题和解决问题,从而实现教学由传授知识到培养能力的转变。

传统 C 语言程序设计教材存在的问题是教材内容过分关注语言的语法和语义细节,以语法规则为纲来组织教学内容。由于 C 语言具有大量繁杂的语法知识点,教师只能采用灌输式教学模式,花费大量课时用于讲授琐碎的语法知识,而不是重在程序设计思想和方法的传授,教学效果不好。

针对高校 C 语言程序设计课程教学中存在的问题,通过梳理现有教学内容,对教学的知识点不做大的更换,主要是改革教材体系结构,我们精心设计教学内容与案例,增加与计算思维有关的内容,以计算思维为主线进行重新组织。为了让学生学会如何运用计算机科学的基础概念进行问题求解,强调与培养计算思维能力有关的数据结构和算法实现,重点突出以“构造”为特征的问题求解方法的培养。每个教学单元从问题开始讲起,以

问题为核心规划单元内容,通过设计性内容,培养学生分析问题、解决问题、研究和探索问题的能力。在解决问题的过程中形成计算性思维。最后通过综合性的应用培养学生的创新意识。让学生应用已有知识分析、解决与实际生活或工作相关的问题。让学生在实践中获得成就感。潜移默化地培养学生的计算思维技能和应用创新能力。

(2) 问题驱动的教学方法

授人以鱼不如授人以渔。融入计算思维,教学方法是关键。从解决问题的角度出发,强调解决问题的方法、思路,这就是计算思维。计算思维用抽象与分解的方法来处理复杂的问题,通过对问题进行合适的表达以及对问题建模来使问题变得容易解决。为了让计算思维在本教材中落地,我们在每个教学单元都采用了问题驱动即基于问题的教学方法(Problem-Based Learning, PBL)。这种方法不像传统教学那样先学习理论知识再解决问题。采用 PBL 教学方法,对于教材的每一章,我们都在第一节提出若干问题和思想方法,在该章后续的内容中再介绍解决本章内容的语法规则。对于章节中的每个例题也都是按这个思路组织的。实施问题驱动的关键是精心设计提出的问题。在设计问题时选择了贴近生活和能让学生感兴趣的内容,注意问题的典型性,同时兼顾知识的覆盖面,学生利用现有的能力能够实现,同时要具有一定的代表性,对解决其他相似问题具有指导意义。

计算思维是一种思维方式,也是一种解决问题的思考过程,计算思维的目标在于解决问题。从现实角度来说,计算思维就是问题抽象、模型建立、算法设计和实现以及问题引申的过程,也就是将未知问题归纳成若干已知问题从而求解的过程。主要从三方面来突出计算思维。

首先是问题驱动,从原始问题入手。原始问题是指在现实世界中客观存在的,尚未被分解、简化、抽象的实际问题。在这个过程中编程者先通过观察,对问题进行深入的认识和理解。有了问题,说明学习的目标找到了,而只有解决问题的学习才是有意义的。生活和工作中需要的,便是学习的方向。通过学习解决问题,新问题出现了,再从解决问题入手进行学习,如此形成良性循环。教材的每一章的第一部分主要以展示经典案例为主,提出一些经典问题,结合生活中的实际应用过程,形象直观地向学生们展现 C 语言程序设计的魅力所在。

其次是问题抽象与分解,将问题同知识点联结。C 程序设计语言包括各种常量、变量、数组、结构、指针和函数等符号。抽象就是使用 C 语言的符号代替实际问题中的各种变量。为了机器的自动化运行,用顺序结构、选择结构和循环结构对过程进行分解,从程序模块化到语句结构化,在抽象分解过程中进行符号转换和编写程序。在分析问题和解决问题中训练学生有关计算思维的抽象和分解能力,寻找一个切入点将该问题同编程者已有的知识和经验进行联结;最后通过变换,将一个问题转化为另一个或分解成另几个容易解决的新问题,以达到解决原问题的目的。从问题引入知识点,用知识点带动学生对语法的掌握、对算法的理解及对设计的熟悉。这种传授知识、技能和计算思维的一体化教学模式,摒弃了原有的“重语法、轻方法、不谈思维”的错误教学方法,而是使教学目标从“知识和技能”的传授上升到“思维”的教学。

第三,注重算法思维。在计算思维技能中,算法思维是在思考使用算法来解决问题的方法。这是学习自己编写计算机程序时需要开发的核心技术。有些问题是一次性的,但

解决这些问题的方案,则可以不断发展。在同类问题重复出现时,算法思维就可以介入。没有必要重新每次从头思考,而是采用每次都行之有效的解决方案。解一个问题,想一想方法,还有哪些问题使用了相同的思想和方法。这就是教了学生计算思维。基于此,在教材中融入一些基本算法,例如迭代、递推、穷举、递归、排序、查找等。在算法讲解中,采取循序渐进的方式进行,例如对于交换的问题,我们从最简单两个量的交换,三个量的交换,到一维数组逆置问题中的多个量交换,再到二维数组转置问题中的下三角元素和上三角对应元素的交换,这样引导学生利用所学知识,一步步分析问题,从简到难,充分调动学生的好奇心和求知欲,发挥个人潜能去分析并寻找解决问题的方法,引导学生编程并在计算机上加以调试实现。

最后,培养学生掌握使用自动化思维进行信息处理的能力。计算思维算法强调利用计算机的速度和存储优势,通过严格机械化的操作时序解决实际问题。比如“鸡兔同笼”问题可以使用两个二元方程组进行数学求解,但是计算机算法则可以使用穷举法进行简单重复的操作求解,充分利用了计算机的计算速度优势。在程序设计课程中强化计算思维训练,可以让学生掌握使用自动化思维进行信息处理的能力。

(3) 教师应是学生与教材交流的中介者

教学过程是教师、学生、教材、环境诸因素交互作用的动态过程,是一个“生态系统”。教师的作用就是要引导、启发。作为老师主要不在于传授知识,而在于引导学生自己去求得知识,也就是引导学生自己去发现问题,自己去解决问题。叶圣陶将教师的这种作用比作教孩子走路:“老师讲,目的是要达到不用讲,好比帮孩子学走路,先牵着他走,扶着他走;进一步让他自己走,在旁边护着他;最后完全可以放心了,就让他自己走,护也不用护了。上课也一样,不能光灌输,要多启发,多引导。”教师要从知识的传授者、灌输者、拥有者转向教学活动的引导者、激励者、服务者。教师要大胆地接受新的教学理念,把学生转换成为课堂的重点,引导鼓励学生自主学习创新的学习能力。使教师的中介作用可以提供给学生更好的教学服务。

教师引导学生养成良好的计算思维和编程的好习惯,有意识地训练学生仔细阅读题目,找出问题的关键并抽象为数学模型,思考数学模型求解的步骤,用所学的计算机编程语言将算法编写成一个源程序在计算机上调试运行,并能对每个程序的结果加以分析,引导学生对已有算法的优缺点进行思考和分析,如何进行简化和优化,最好能给出一些能优化改进的算法。这种中介引导作用提炼并展现了隐藏在知识背后的计算思维的光芒,引起学生求知欲望和心理共鸣,就是培养了学生的计算思维能力。

(4) 简单易用的 Dev-C++ 上机环境

对于学生来说,通过上机实践,最初学习是很有趣味的,但在后续学习中却因为语法内容的增多,使用语言工具的繁琐,反而降低了学习热情。所谓的“成在实践,败在工具”。

“工欲善其事,必先利其器”,早期用得最多的 IDE 开发工具是 Turbo C。1997 年微软推出了一款 C/C++ 编译器——VC++ 6.0,VC++ 6.0 界面友好,调试功能强大,非常经典。很多高校将 VC++ 6.0 作为 C 语言的教学基础,作为上机实验的工具,至今仍然有很多企业和个人在使用。对于初学者 VC++ 6.0 是比较容易上手的,但其对标准支持得不好可能使人养成不良的编程习惯。VC++ 6.0 适合在低配置的机器上使用,在高配置的

系统上,VC++ 6.0 有一些兼容性问题。可能在配置环境的时候给初学者造成一些不必要的麻烦。微软后来对 VC++ 6.0 进行了升级,并更名为 Visual Studio(简称 VS),支持更多的编程语言,拥有更加强大的功能。但是 Visual Studio 文件很大,约有 3GB,大部分功能初学者暂时不会用到,而且安装繁琐,不用时很难卸载干净,占用资源较多,低配置的机器不能使用。

GCC(GNU Compiler Collection,GNU 编译器套装)是一套由 GNU 开发的编程语言编译器。它是一套以 GPL 及 LGPL 许可证所发行的自由软件,也是自由的类 UNIX 及苹果计算机 Mac OS X 操作系统的标准编译器。GCC(特别是其中的 C 语言编译器)也常被认为是跨平台编译器的事实标准。Linux 平台下推荐使用 GCC。

在 Windows 下,也有集成了 GCC 的开发工具,如 Dev-C++。Dev-C++ 是一款自由软件,遵守 GPL 许可协议分发源代码。Dev-C++ 集成使用了 GNU 的 GCC,对于 C99 支持最好的编译器,它的 ANSI C/C++ 兼容性远好于微软的 VS 系列。可以用此软件实现 C/C++ 程序的编辑、预处理、编译、链接、运行和调试。它是一种轻量级的 C/C++ 编程工具,具有关键字加亮,不同颜色显示等功能,还具有代码自动补全功能,可以根据程序上下提示所有可能的指令,从而提高编程效率,还有完善的调试功能。简单配置就可以上手使用,让初学者免于面对庞大 IDE 比如 Visual Studio 的困惑,让初学者更多地关注于语言本身而不是一个繁琐的 IDE。

由于 Dev-C++ 简单实用,操作简便,特别适合初学者,避免了初学者学习 C 语言编程时使用其他开发工具的困惑。Dev-C++ 是学习 C 或 C++ 的首选开发工具,适合初学者与编程高手的不同需求。Dev-C++ 是 NOI、NOIP 等比赛的指定工具,ACM 比赛测试用的也是 Dev-C++ 环境。作者推荐使用 Dev-C++ 作为高校 C 语言的教学和上机实验的工具,本教材上机环境也是使用的 Dev-C++。

(5) 因类施教的折中式体系编排

根据各个计算机和非计算机专业(例如理学类、艺体类、工程类和文史类)学生的特点和各个专业的需要,首先在教学内容的选取上加以区别对待,在教材体系编排上采用折中式组织。兼顾程序设计和语法规则两方面的需要,将每一章分成基本内容和阅读延伸两部分。在每一章的基本内容中,主要讲解程序设计的思路和方法以及 C 语言的重要语法规则,重点突出程序设计,学生们学完了这部分内容就可以编写程序了。将一些在课堂教学中难以组织的内容放在阅读延伸部分,如一些较大的应用问题、一些语法细节以及并不是所有读者都感兴趣的非主流问题等,另外各个学校之间,同一学校计算机和非计算机专业所讲授内容有所不同,程序设计课程的教学时数一般为 64~96。基于这些差异性,教师可以根据自己专业的特点选择讲授阅读延伸部分内容。也可让学生自学这部分内容。这样做极大地方便了教师的使用,改变了学生先学习大量枯燥的知识再编程的方法,大大提高了学习效率。

写给使用本教材的学生:

学习 C 语言编程,首先要解决的是书籍。教材不仅遵循学科逻辑,而且遵循学生学习的心理逻辑,形成了高度结构化的教材,不仅为教师教学提供基本的操作框架和步骤,提示教学方法,而且还为学生学习提供指引。学编程,仅靠记概念、背算法是不够的。本

教材在每章的基本内容中提供了丰富的例题,对每一个例题或者算法,要注意总结其中的算法思想和程序设计过程与方法,还有其中涉及的重要语法规则及应用方式等。每一章都配备了习题和实验问题,便于学生课后做编程练习。建议初学者把所有的习题都独立做一遍,然后对照答案的代码,看看自己的代码有哪些不足,再试着修改自己的代码。习武之人首先要学的都是各种套路,题做多了,才能做到“做题过万道,编程如有神”。

其次是学习方法,好的学习方法将事半功倍。学生们在编写完程序之后,还可以上机实践。通过写程序和上机实践,就会慢慢理解程序设计的思想,用过的方法多了,遇到问题时才有可能想到解决的思路。初学者可能会感觉编程很难,可是当你编写出了一个程序,就会觉得其实也并不是那么难,就会有成就感。反过来这种感觉会更加激发学习的热情。

最后,我们精心制作了本教材的视频课程,将做成慕课帮助学生更好地学好 C 语言编程,提供实时交互的在线编程环境,互帮互助的问答社区,师生之间、学员之间还可以在线交流。

由于时间仓促,书中难免存在不妥之处,请读者原谅,并提出宝贵意见。

作 者

2017 年 7 月

目录

C 语言程序设计基础

第 1 章 程序及其执行	1
1.1 本章知识结构图	1
1.2 本章学习导读	1
1.2.1 程序	2
1.2.2 程序设计语言	3
1.2.2.1 几种常用的高级语言	3
1.2.2.2 C 语言的词法记号	4
1.3 C 语言程序的基本结构	8
1.4 C 语言程序上机运行四部曲	10
1.5 习题 1	11
1.6 实验 1 学习 Dev-C++ 开发环境实验(2 学时)	12
1.7 阅读延伸	12
1.7.1 为什么要学习 C 语言	12
1.7.2 Dev-C++ 集成开发环境	15
第 2 章 如何设计 C 程序	19
2.1 本章知识结构图	19
2.2 本章学习导读	19
2.2.1 程序设计的关键——设计一个好的算法	20
2.2.1.1 算法及其特性	20
2.2.1.2 算法、数据结构和程序	23
2.2.1.3 算法的分类	25
2.2.1.4 算法的表示形式	26
2.2.2 结构化程序设计方法	28
2.2.3 程序设计过程——程序开发 5 步骤	29
2.2.4 习题 2	31
2.2.5 实验 2 C 程序开发过程实验(2 学时)	32
2.2.6 阅读延伸	33
2.2.6.1 程序设计风格	33

2.6.2	学习程序设计的目的——培养计算思维能力	36
2.6.3	怎样学好程序设计	37
第3章	顺序结构	39
	本章知识结构图	39
	本章学习导读	40
3.1	顺序结构的思维方法	40
3.2	C语言的数据及其运算	41
3.2.1	常量	41
3.2.2	变量	44
3.2.3	运算表达式	48
3.3	数据处理——赋值语句	51
3.4	数据输出——格式化输出库函数 printf	53
3.5	数据输入——格式化输入库函数 scanf	55
3.6	顺序结构的应用	58
3.6.1	职工工资问题	58
3.6.2	两个数交换问题	59
3.7	习题3	59
3.8	实验3 顺序结构程序设计实验(2学时)	60
3.9	阅读延伸	62
3.9.1	字符数据的输入和输出库函数	62
3.9.2	各种基本类型的变量定义	64
3.9.3	数据的机内形式和机外形式	65
3.9.4	sizeof 运算符	66
3.9.5	数据类型转换	67
3.9.6	C语言程序预处理命令	68
第4章	选择结构	72
	本章知识结构图	72
	本章学习导读	72
4.1	逻辑选择的思维方法	73
4.2	if语句与条件判断	75
4.2.1	if语句的一般形式	75
4.2.2	关系运算	76
4.2.3	逻辑运算	77
4.3	三种选择结构	80
4.3.1	用if语句实现单路选择结构	80
4.3.2	用if语句实现双路选择结构	81

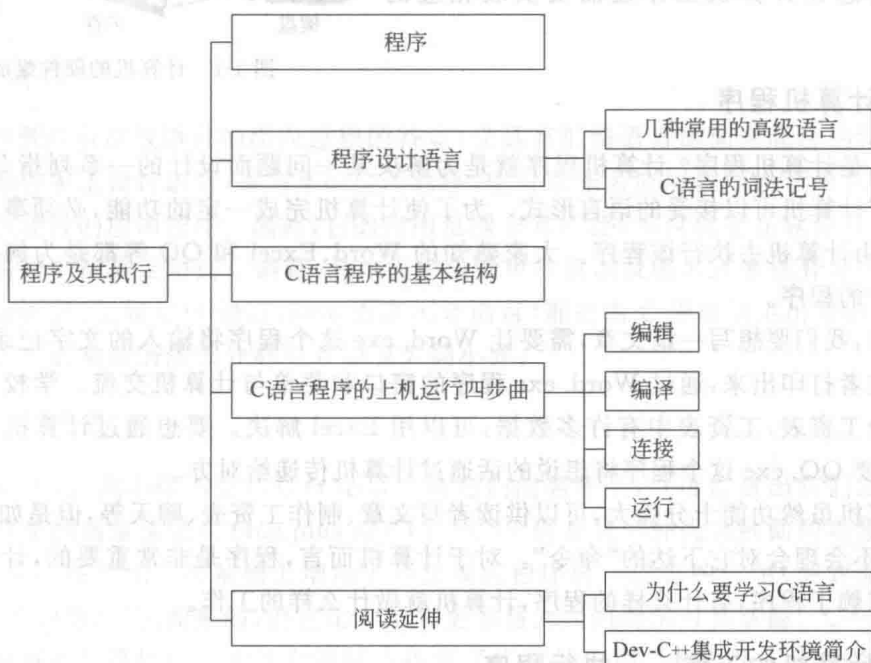
4.3.3	用 if 语句实现多路选择结构	82
4.4	switch 语句与多路选择结构	84
4.5	选择结构的应用	87
4.5.1	书店收银问题	87
4.5.2	三个数的排序问题	88
4.6	习题 4	90
4.7	实验 4 选择结构程序设计实验(4 学时)	91
4.8	阅读延伸	96
第 5 章 循环结构		98
本章知识结构图		98
本章学习导读		98
5.1	循环控制的思想方法	99
5.2	三种循环控制语句	100
5.2.1	while 语句	100
5.2.2	do...while 语句	102
5.2.3	for 语句	103
5.3	两种循环结构——计数型和非计数型	105
5.4	循环结构的嵌套	107
5.5	循环结构中的两个辅助控制语句	108
5.6	循环结构的应用	111
5.6.1	出租车计价问题	111
5.6.2	图形输出问题	112
5.7	习题 5	114
5.8	实验 5 循环结构程序设计实验(6 学时)	117
5.9	阅读延伸	121
5.9.1	迭代问题	121
5.9.2	递推问题	121
5.9.3	穷举问题	124
第 6 章 数组变量		129
本章知识结构图		129
本章学习导读		129
6.1	为什么要使用数组	130
6.2	一维数组的定义和使用	131
6.3	一维数组编程举例	133
6.4	二维数组的定义和使用	136
6.5	二维数组编程举例	139

6.6	字符数组的定义和使用	141
6.6.1	字符数组和字符串	141
6.6.2	字符数组的定义	142
6.6.3	字符数组的使用	143
6.7	字符数组编程举例	147
6.8	习题 6	148
6.9	实验 6 数组与字符串实验(6 学时)	150
6.10	阅读延伸	154
6.10.1	数组综合应用——歌手评分问题	154
6.10.2	二维数组的物理结构	156
第 7 章	函数	158
	本章知识结构图	158
	本章学习导读	158
7.1	为什么要使用函数	159
7.2	如何使用函数	161
7.2.1	函数定义	161
7.2.2	函数调用	162
7.2.3	函数原型	164
7.3	函数嵌套调用	165
7.4	数组名做函数参数问题	166
7.4.1	数组名参数的传址方式	166
7.4.2	参数中不指定数组元素个数	168
7.4.3	多维数组作为函数参数	171
7.5	变量的作用域和生存期	172
7.6	习题 7	174
7.7	实验 7 模块化程序设计实验(6 学时)	175
7.8	阅读延伸	178
7.8.1	模块化程序设计应用——歌手评分问题	178
7.8.2	函数的递归调用	180
7.8.3	变量的存储类别	184
7.8.4	函数的存储类别	191
7.8.5	C 语言程序的内存布局	192
第 8 章	指针变量	196
	本章知识结构图	196
	本章学习导读	196
8.1	指针变量概述	197

8.1.1	什么是指针变量	197
8.1.2	如何定义和使用指针变量	197
8.2	数组指针变量	201
8.2.1	一维数组元素的指针变量	201
8.2.2	二维数组元素的指针变量	205
8.3	字符串指针变量	207
8.4	指针变量做函数参数	209
8.5	指针函数	218
8.6	指针数组	220
8.7	习题 8	223
8.8	实验 8 指针编程实验(4 学时)	226
8.9	阅读延伸	228
8.9.1	二维数组的行指针变量	228
8.9.2	函数指针变量	231
8.9.3	多级指针变量	233
8.9.4	main 函数参数	235
8.9.5	动态内存分配问题	236
第 9 章	用户构造数据类型	239
	本章知识结构图	239
	本章学习导读	239
9.1	结构类型	240
9.1.1	结构类型的定义	240
9.1.2	结构变量的定义	241
9.1.3	结构变量的使用	243
9.1.4	结构指针变量	246
9.2	结构数组	248
9.2.1	定义结构数组	248
9.2.2	结构数组的初始化	249
9.2.3	结构数组的指针变量	249
9.2.4	结构数组指针做函数参数	251
9.3	用户自定义类型	252
9.4	习题 9	253
9.5	实验 9 结构数组的应用实验(4 学时)	255
9.6	阅读延伸	259
9.6.1	基础数据结构之链表	259
9.6.2	联合类型	265
9.6.3	枚举类型	268

第 10 章 数据文件	270
本章知识结构图	270
本章学习导读	270
10.1 文件类型	271
10.2 文件的打开与关闭	272
10.2.1 文件的打开函数 fopen	273
10.2.2 文件关闭函数 fclose	275
10.3 文件的顺序读写	275
10.3.1 字符读写函数 fgetc 和 fputc	276
10.3.2 字符串读写函数 fgets 和 fputs	280
10.3.3 数据块读写函数 fread 和 fwrite	282
10.3.4 格式化读写函数 fscanf 和 fprintf	285
10.4 文件的随机读写	287
10.5 文件的检测	290
10.6 习题 10	292
10.7 实验 10 文件的应用实验(2 学时)	293
10.8 阅读延伸	295
10.8.1 文件的应用——用户登录检测问题	295
10.8.2 文件使用中的几个问题	298
附录 A ASCII 码表	299
附录 B C 语言常用标准函数库	300
附录 C C 语言的关键字和运算符	303
参考文献	304

本章知识结构图



本章学习导读

程序及其执行是学习程序设计的两大基本问题。本章主要包括什么是计算机程序、怎样用C程序设计语言书写程序。在介绍了C语言的词法记号、C语言程序的基本结构后,讨论了C语言程序的编辑、编译、连接和运行等内容。

另外,在本章阅读延伸一节中还针对程序设计初学者给出了为什么要学习C语言及Dev-C++集成开发环境介绍。

1.1 程 序

1. 计算机的硬件组成

要使用计算机从事程序设计工作,必须要先了解计算机是怎样组成的。我们目前使用的个人计算机主要包括中央处理器(CPU)、内存、外存、键盘和显示器等,如图 1-1 所示。其中,CPU 是计算机的核心部件,其主要功能是执行存放在内存中的程序,计算机本身是无生命的机器。要想让计算机工作还需要安装相应的软件。

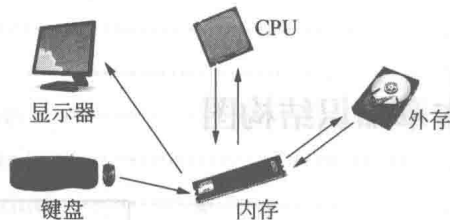


图 1-1 计算机的硬件组成

2. 计算机程序

什么是计算机程序?计算机程序就是为解决某一问题而设计的一系列指令或语句,它们具有计算机可以接受的语言形式。为了使计算机完成一定的功能,必须事先编好程序,然后由计算机去执行该程序。大家熟知的 Word、Excel 和 QQ 等都是为解决某一问题而设计的程序。

例如,我们要想写一篇文章,需要让 Word. exe 这个程序将输入的文字记录下来,然后保存或者打印出来,通过 Word. exe 程序的窗口和菜单与计算机交流。学校财务处要制作一份工资表,工资表中有许多数据,可以用 Excel 解决。要想通过计算机跟他人聊天,就需要 QQ. exe 这个程序将想说的话通过计算机传递给对方。

计算机虽然功能十分强大,可以供读者写文章、制作工资表、聊天等,但是如果没程序,它就不会理会对它下达的“命令”。对于计算机而言,程序是非常重要的,计算机的工作完全依赖于程序,有什么样的程序,计算机就做什么样的工作。

3. 计算机的工作——执行程序

计算机唯一的函数就是执行程序。计算机如何执行程序呢?

在一台计算机上,程序从某种外部设备,通常是硬盘,被加载到内存。然后由 CPU 去执行该程序。例如,当我们使用 Word 程序时,需要单击 Word 程序图标,它就被从硬盘加载到内存,然后由 CPU 去执行该程序中的指令完成我们所做的工作。整个过程都是由程序控制的。所以,计算机工作的原理就是存储程序和程序控制原理。冯·诺依曼结构的计算机即存储程序的计算机。计算机的本质是程序的机器。

可是,计算机是不懂得人类使用的自然语言的,那么程序是如何将人类的旨意传达给计算机又让它去执行的呢?这里,就需要用到接下来要讨论的程序设计语言。通过程序设计语言书写程序,才能让计算机依程序行事。

1.2 程序设计语言

1.2.1 几种常用的高级语言

程序设计语言(Programming Language)也称为编程语言,俗称“计算机语言”,编程语言的描述一般可以分为语法(Syntax)及语义。语法是说明编程语言中,哪些符号或文字的组合方式是正确的,语义则是对于编程的解释。

在过去的几十年间,大量的高级语言被发明、被取代、被修改或组合在一起成为新的计算机语言。经过实践的检验,现在还在业界普遍流行的计算机程序设计语言主要有以下几种。

1. C 语言

C 语言既具有高级语言的面向过程的特点,又具有汇编语言的面向底层的特点。它可以作为操作系统设计语言,编写系统应用程序,也可以作为应用程序设计语言,编写不依赖计算机硬件的应用程序。因此,它的应用范围非常广泛,不仅仅是在软件开发上,而且在各类科研中都需要用到 C 语言,具体应用包括单片机以及嵌入式系统开发等。当前流行的高级语言(例如 C++ 语言、Java 语言、C# 语言)都是由 C 语言衍生出来的。因此,所有这些语言的基础语法部分都与 C 语言大同小异。

2. C++ 语言

说到 C 语言,就不得不提到 C++ 语言。从它们的名字上,就可以看出它们之间的关系。C++ 这个词通常读做 C plus plus 或 CPP。C++ 语言是一种优秀的面向对象的程序设计语言,它是在 C 语言的基础上增加了一些现代程序设计语言的机制(例如面向对象思想、异常处理等)发展而来的,但它比 C 语言更容易为人们所学习和掌握。C++ 以其独特的语言机制在计算机科学的各个领域得到了广泛的应用。相对于 C 语言的面向过程的设计方法,C++ 的面向对象的设计思想让它有了一个质的飞跃,使得 C++ 更加适合于对性能要求较高的、大型的复杂系统的开发。

3. Java 语言

Java 是一种可以撰写跨平台应用软件的面向对象的程序设计语言,是由 Sun Microsystems 公司于 1995 年 5 月推出的 Java 程序设计语言和 Java 平台(即 Java SE, Java EE, Java ME)的总称。Java 技术具有卓越的通用性、高效性、平台移植性和安全性,广泛应用于个人 PC、数据中心、游戏控制台、科学超级计算机、移动电话和互联网,同时拥有全球最大的开发者专业社群。在全球云计算和移动互联网的产业环境下,Java 更具备了显著优势和广阔前景。

除了在上面介绍的几种程序设计语言之外,还有很多优秀的程序设计语言没有介绍,