高职高专电子信息类“十三五”规划教材

C语言程序设计基础

主　编　李建忠
副主编　周贤来　杨本胜

西安电子科技大学出版社
http://www.xduph.com

高职高专电子信息类“十三五”规划教材

C语言程序设计基础

主　编　李建忠
副主编　周贤来　杨本胜
参　编　郑燕逵　吕　雪　李文胜　肖　红

西安电子科技大学出版社

内 容 简 介

本书贯彻“任务驱动，学中练，练中学”的教学理念，根据高职高专电子信息类专业“C语言程序设计基础”课程的教学要求和特点，按程序的功能结构组织内容，把数据结构、算法等融入到C语言结构和程序设计的过程中，全面介绍了C语言体系内容和基本程序设计方法，内容精练，注重思路方法的传递，突出了实践和应用。全书每一章内容均分为“理论学习”和“实操训练”两部分，且每一章后均编排了“认知理解、辩解分析、程序设计”三个层次的实训项目。

本书可作为高等职业院校“C 语言程序设计基础”课程的教材，也可作为应用型本科“C语言程序设计”课程的教材。

图书在版编目(CIP)数据

C语言程序设计基础/李建忠主编. —西安：西安电子科技大学出版社，2016.8
高职高专电子信息类“十三五”规划教材
ISBN 978-7-5606-4237-6

Ⅰ. ① C… Ⅱ. ① 李… Ⅲ. ① C语言—程序设计—高等职业教育—教材 Ⅳ. ① TP312

中国版本图书馆 CIP 数据核字(2016)第 176382 号

策　　划　毛红兵
责任编辑　买永莲
出版发行　西安电子科技大学出版社(西安市太白南路 2 号)
电　　话　(029)88242885　88201467　　邮　　编　710071
网　　址　www.xduph.com　　电子邮箱　xdupfxb001@163.com
经　　销　新华书店
印刷单位　陕西华沐印刷科技有限责任公司
版　　次　2016 年 8 月第 1 版　2016 年 8 月第 1 次印刷
开　　本　787 毫米×1092 毫米　1/16　印　张　18
字　　数　423 千字
印　　数　1～3000 册
定　　价　38.00 元

ISBN 978-7-5606-4237-6/TP

XDUP 4529001-1

前　言

“C 语言程序设计基础”是大多数高职院校计算机信息类专业的一门核心专业基础课。该课程不仅要使学生掌握编程语言工具，而且要学习数据结构、算法、程序设计的基本方法，培养学生程序设计和调试的能力，更重要的是培养学生基于计算机工作原理的程序设计思维方式，为后续专业课的学习奠定基础。由高职人才培养定位和模式决定，该课程包含了传统计算机信息类专业的多门专业基础课内容，不仅要学生建立起够用的基础知识，而且要培养软件开发设计的多种能力，这就给该课程教学的内容体系、教学方式与方法及教材设计提出了值得研究的课题。

教材是教学内容和教学方法的载体，是教学质量的基本保证。近几年，针对高职人才培养的要求与特点，“C 语言程序设计基础”课程的教材进行了多角度的改革与创新，但普遍存在一种倾向：强调实践能力，引入任务及项目驱动，不太重视理论体系的严谨性和完整性。C 程序设计基础本身具有严密的理论体系，实践应用能力在相应理论知识的指导下才能获得。一些教材采用任务及项目驱动模式，在特定项目下引入所需理论知识，势必割裂了理论知识的内在联系，在教学效果上形成“空中楼阁”、“一盘散沙”的现象，更谈不上培养学生思维方法。但如果偏重理论体系，又会形成“纸上谈兵”，更不符合高职学生的培养要求。本书结合教学实践，力图体现以下特色：

(1) 把 C 语言形式、程序结构、数据与算法、程序设计方法相融合，在此基础上组织内容体系。本书综合地考虑了程序的各种功能结构关系，按照由简单到复杂，由浅入深的认知规律组织了篇章内容，把数据结构、算法、程序设计方法等基础知识贯穿于 C 语言结构和程序设计的过程中。每一章标题都有一个主标题和一个副标题，例如“第 6 章数组——批量数据的表示与处理”，主标题表示 C 语言内容，副标题表示程序功能、结构、方法方面的内容。在篇章内容的组织上，尤其注重把 C 语言基本内容与程序功能、结构、方法有机联系起来，使学生每学一部分内容，其编程能力就可上升一个程度。

(2) 注重理论够用，突出应用与实践。在每一章都编排理论学习和实操训练两部分内容。理论部分贯穿着应用，着力挖掘内涵与规律，突出重点与难点，条理清晰，语言简练，篇幅短小。实操训练部分贯穿着任务驱动，突出实践与应用。

(3) 注重思路方法的传递，可激发学生学习兴趣，培养自主学习能力。在知识讲解中，贯穿问题引导的教学理念，先根据内容恰当地提出问题，然后围绕问题讲解内容，并给出相应的程序。

本书由广州松田职业学院机电与信息工程系多年从事“C 语言程序设计基础”课程教学的骨干教师，结合教学实际，在开展教学改革与研究的基础上，汇集相关教研成果和实践经验编写而成。其中，李建忠担任主编，周贤来、杨本胜担任副主编，郑燕逵、吕雪、

李文胜、肖红参与了本书编写的研讨与审定，并交叉编写了相应章节内容。

本书的编写工作得到了广州松田职业学院领导和教务处及相关老师的高度重视和大力支持，在此表示衷心的感谢。

编者力图在书中反映“C 语言程序设计基础”课程教学改革与研究成果的一些思想与方法，使本书具有一定特色，但因水平限制，书中难免存在不妥或疏漏之处，敬请广大读者批评、指正。

编　者

2016 年 5 月

目　录

第 1 章　C 语言程序设计概述

理论学习

在学习 C 语言程序设计的具体内容之前，对程序设计语言、程序设计的基本方法、数据结构与算法、程序设计语言的内容体系、C 语言的基本特点、C 语言程序结构有一个初步了解，有利于明确学习目的与内容，并确立正确的思维方式与学习方法。本章正是从这个角度出发，对上述内容作简要概述。

1.1　程序设计语言

为什么要学习程序设计语言？

从计算机外部看它的工作，似乎计算机具有人的智能，能实现人的各种意图。而事实上，计算机是程序控制的高度自动化的信息处理工具。要让计算机来实现人的意图，必须将意图编写成程序，输入并存储到计算机的存储器中，只有当计算机执行程序时，才会表现出按人的意图进行规定的操作。也就是说，没有程序，计算机就表现不出任何功能。

程序是把解决某一问题的方法步骤向计算机的表达或描述。这种表达或描述必须采用人与计算机能进行交互的语言。这种与计算机交互的语言就是程序设计语言。

编程语言有哪些？各具有什么特点？

随着计算机应用技术的发展，程序设计语言也经历了不同阶段的发展，出现了多种语言。从人与计算机的交互性而言，程序设计语言可分为低级语言和高级语言。低级语言包括机器语言、汇编语言，高级语言则分为面向过程的高级语言和面向对象的高级语言。

机器语言是计算机能够直接识别的语言，它是一组二进制编码指令。在计算机应用初期，人们用机器语言编写程序。但是机器语言是冗长的二进制代码，难理解、难记忆、难编程，只有少数计算机专业人员才会使用。随着计算机应用技术的发展，计算机语言一直朝着“人性化”的方向发展，先后出现了汇编语言和不同种类的高级语言。

汇编语言是用一组便于人们记忆的助记符号来表示机器语言的指令。汇编语言中的一条指令对应一条机器指令。机器语言和汇编语言都是面向具体计算机的语言，每一种类的计算机都有自己独特的机器语言和汇编语言。由于它们依赖具体的计算机，所以被称为“低

级语言”。

高级语言不依赖于具体的计算机，是在各种计算机上都通用的程序设计语言。高级语言接近人们习惯使用的自然语言和数学语言，易于学习和使用。高级语言出现于 20 世纪 50 年代。人们认为，高级语言的出现是计算机发展史上一次惊人的成就，通过它，非计算机专业人员能方便地编写程序，并使用计算机来解决各专业领域的问题。随着计算机应用技术的发展，高级语言又分化为面向过程的语言和面向对象的语言。

1．面向过程的语言

所谓过程，其实质是某个输入集合到某个输出集合的一个映射，可以用某种描述形式来说明这种映射的细节。从解决问题的角度来讲，可以把解决问题的方式抽象为较大的过程，再把这个较大的过程分解为较小的过程，一直到具体的操作步骤。这些过程都可以用某种高级语言按照一定的顺序和规则进行描述。这种类型的程序设计语言称为面向过程的高级语言。也就是说，用面向过程的语言编程只需根据解决问题的过程，把过程的每一步按顺序用语言描述出来，也称为基于算法的编程语言。面向过程的编程语言有 BASIC、FORTRAN、COBOL、PASCAL、C、Ada、LISP 等。

2．面向对象的语言

面向对象的语言是用客观世界中描述事物的方法来描述一个程序要描述的事物，使程序设计更接近现实世界的思维方式，更简捷高效。面向对象的程序设计是以处理的数据为对象，根据对象的属性与操作进行封装并定义接口来组织程序。常用的面向对象的程序设计语言有 Visual C++、Visual Basic、Dephi、Java 等。

计算机编程语言只是提供了一种人与计算机进行交互的方法或工具。计算机唯一能识别的语言只有机器语言，所有非机器语言编写的程序都要转换为机器语言，才能在计算机中执行。这种转换一般称为“编译”。编译由系统软件来完成。所以，为了方便用户使用，任何高级语言都有一个实现编译等操作功能的语言处理软件。C 语言的语言处理软件有多种，如 Touber C、Visual C++ 6.0 等。

如何将非机器语言程序转换为计算机能够识别的机器语言代码？

由非机器语言编写的程序一般称为源程序，能被计算机直接执行的程序称为可执行程序。把一个源程序变换成一个可执行程序，一般要经过图 1.1 所示的处理。

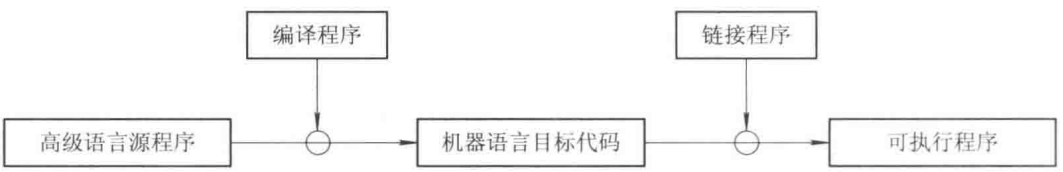


图 1.1　源程序转换处理步骤

高级语言的种类很多，不同的语言各有不同的特点与使用场合，但从原理上看，各种语言都包含一些相同的功能和结构。对初学者来说，只要选择有代表性的语言，掌握了程序设计语言的规律和内在功能结构，就很容易学习和应用其他语言。现在一般都把 C 语言作为程序设计的入门语言。

1.2　程序设计的基本方法

程序设计需要哪些知识与技术？

程序设计涉及多方面的知识和技术，可以将程序设计所包含的内容表示为

程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具

程序设计是以上四个方面知识的综合运用与贯通。事实上，这四个方面中的每一个内容都属于有关专门课程的范畴。本书的重点是通过 C 语言来学习程序设计，但对算法、数据结构、程序设计方法应有一个初步了解，以便于理解和设计 C 语言程序。

1.2.1　数据结构与算法

数据结构与算法具有密切的联系。数据结构是加工对象，算法则是对数据结构加工处理的方法。不同的数据结构可能需要采取不同的算法，不同的算法可以对不同的数据进行加工处理。

1．数据结构

何谓数据结构？数据结构在程序中如何表示？

数据结构指的是数据的组织形式，例如，字符数组、方程组系数矩阵、人员的基本信息表、反映部门组织机构关系的树形图、反映网络结构及信息的网状图等都是不同的数据结构。

一般情况下，数据元素之间不是相互孤立的，常常存在某些逻辑上的联系，这些联系与存储无关，独立于计算机。我们把客观事物本身的逻辑关系称为逻辑结构。

数据要能被计算机处理，就必须存储在计算机中，把数据元素及其关系在计算机内的表示称为数据的存储结构，它是数据的逻辑结构在计算机存储器中的映射。

组织和存储数据的目的是能被计算机处理，即对数据施加各种运算。因此，可以在逻辑结构上定义运算集合，而在存储结构上实现这些运算。

在程序设计语言中，一般以数据类型来表示一种数据结构，定义一种数据类型既表示了其逻辑结构，同时也就定义了其存储结构。例如 C 语言中整型、实型、字符型、数组、结构体与共用体等都表示相应的数据结构。

2．算法

何谓算法？算法如何表示？

所谓算法，是解决某一问题所采取的方法和步骤。程序设计中的算法即把解决问题的每一步骤具体化为计算机的操作。也即算法是解决计算机在什么情况下应该“做什么”和“怎样做”的问题的。

软件行业把软件开发分为设计和编码两个不同的阶段。所谓设计就是算法的设计，编码则是对算法以编程语言来表示。算法的设计就是算法的分析与表示。算法设计没有固定的模式。同一个问题可有多种算法，不同的设计者可以设计出不同的算法。下面通过一个

实例来介绍算法及算法设计。

例如：一个班有 30 个学生，要求将一门课考试成绩不及格的学生的学号和成绩打印出来。

假设用 c 表示学生的学号，c_i 代表第 i 个学生的学号，用 s 表示学生的成绩，s_i 代表第 i 个学生的成绩，则问题的算法步骤如下：

(1) 1⇒i(相当于设立了一个指针，先指向第一个学生)；

(2) 如果 $s_i<60$，则打印出学号 c_i 和成绩 s_i，否则不打印；

(3) i+1⇒i(相当于修改指针，使其指向下一个学生)；

(4) 如果 i≤30，返回步骤(2)，继续执行，否则算法结束。

算法要采用一定的形式表示出来，其目的是便于用一种计算机编程语言把算法变换成能在计算机上实现的程序。算法的描述有多种方式，如图解方式、语言方式等。其中，流程图是最容易理解的一种最基本的算法描述方法。为了使读者在 C 语言学习中会分析并表示算法，下面对流程图作一简单介绍。

流程图采用一些规定的图形符号来表示相应的处理流程。常用的图形元素如图 1.2 所示。

上例用流程图表示则如图 1.3 所示。

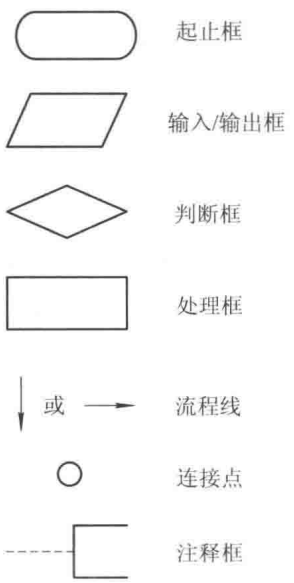


图 1.2　流程图中常用的图形元素

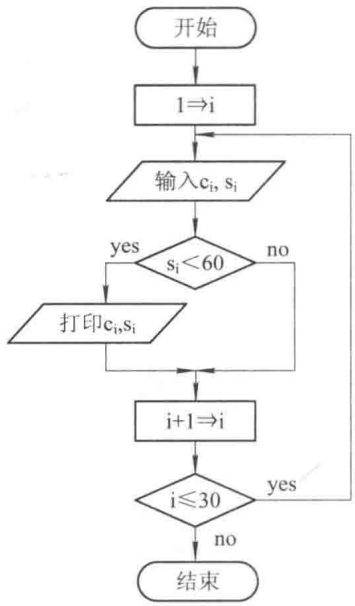
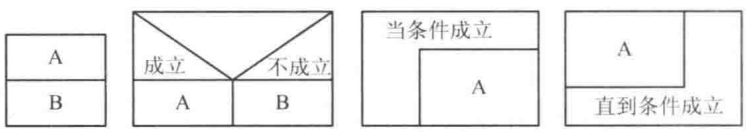


图 1.3　算法流程图

流程图的另一种表示形式是 N_S 图。N_S 图是去掉流程线，把算法写在一个框内的流程图。几种基本程序结构的 N_S 图符号如图 1.4 所示。图 1.3 算法的 N_S 图表示如图 1.5 所示。



(a) 顺序结构　(b) 选择结构　(c) 当型循环结构　(d) 直到循环结构

图 1.4　几种基本程序结构的 N_S 图符号

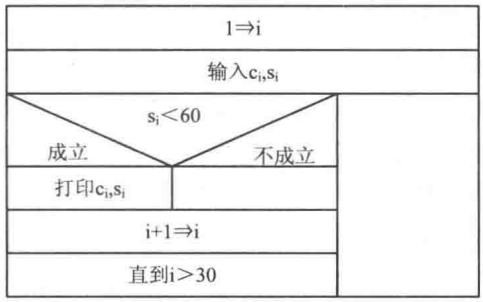


图 1.5　N_S 图示例

1.2.2　程序设计方法

程序设计的基本方法有哪些？

程序设计方法涉及非常丰富的内容，但模块化程序设计思想始终贯穿在所有程序设计的过程中。

模块化程序设计方法的基本思想就是抽象和分解。

抽象是人类认识世界的基本法则之一。现实世界中的某些事物、状态或过程之间总是存在着某些相似的方面，把这些方面集中和概括起来，暂时忽略它们之间的差异，或者说抽出事物的本质特性而暂时不考虑其细节，这就是抽象。

抽象包含了系统的观点和分层的观点，即可以把程序设计问题看成一个系统，这个系统可用高级的抽象概念来理解和构造，这些高级的抽象概念又可用较低级的抽象概念来理解和构造，如此进行下去，直到最低层次的模块可用某种程序设计语言的语句来表示为止。

实际上，对系统的每一次抽象都是向更具体的方面推进的过程，也是进一步分解的过程。因此，在抽象的过程中，同时伴随着分解。当我们遇到一个较大、较复杂的问题时，不应急于编写程序代码，而应该首先把问题自顶向下、逐步分解，细化成一个个程序模块，然后再对每一个模块进一步细化。处于不同层次的模块应该只考虑本模块内部的问题而不必考虑其他模块内部的问题，顶层模块控制系统的主要功能并影响全局，底层模块则完成对数据的具体处理。

信息隐蔽和局部化是模块化程序设计方法的另一基本思想。这一思想意即在模块化的程序设计过程中，一个模块内包含的信息不能被不需要这些信息的模块访问。也就是说，模块的划分应该遵循独立性原则，即模块彼此之间互相依赖的程度要小，而模块内部各元素彼此结合要紧密。

结构化程序设计是程序设计的重要方法。结构化程序设计是以模块设计为中心的，任何简单或复杂的算法都可以由顺序结构、选择结构和循环结构这三种基本结构组合而成。所以，这三种结构就被称为程序设计的三种基本结构。

1.3　程序设计语言的内容体系

程序设计语言包括哪些基本内容？应以怎样的思想方法学习程序设计语言？

程序设计语言是人与计算机进行交互的语言。它既具有自然语言的要素和规律，又有

计算机处理所要求的一些特点。

为了从自然语言体系来联想程序设计语言体系，图 1.6 对自然语言和程序设计语言进行了对比。从图中可以看出，程序设计语言与自然语言具有可类比的体系内容，但它们又有很大差别。自然语言中，在字词及符号的基础上，依据语法规则可造句。而程序设计语言中，每一条语句表示计算机内部的一个功能操作，且有严格的格式规定，必须按照规定书写，否则计算机系统就不能识别。程序依据算法，由相应的功能语句构成，要求计算机实现一个特定的功能。学习了简单数据对象和简单算法的结构语句就可以编写简单程序，进一步学习复杂数据对象、复杂算法的结构语句就可以编写复杂功能程序了。

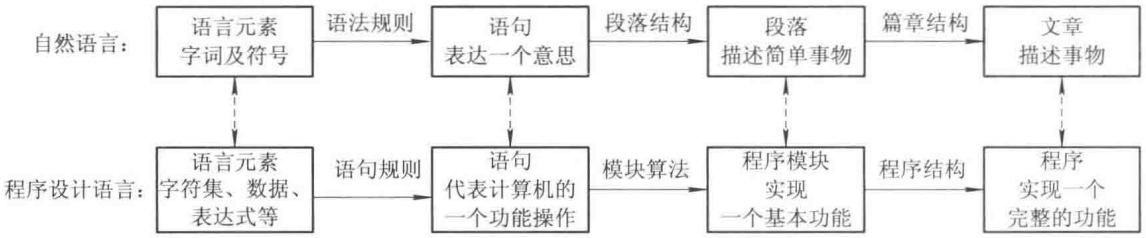


图 1.6　自然语言和程序设计语言的对比

1.4　C 语言的特点

为什么要学习 C 语言？

C程序设计语言是20世纪70年代初贝尔实验室的D.M.Ritchie在总结之前高级语言优、缺点的基础上，并针对 UNIX 操作系统的研发而设计出的高级编程语言。随着 UNIX 操作系统的闻名，C 语言很快风靡世界，成为最流行的程序设计语言。在 C 语言的发展初期，出现了多种版本，标准不一。1983 年，美国国家标准化协会(ANSI)制定了 C 语言标准，称为 ANSI C。随着计算机应用技术的发展，C 语言也经历了不断更新与发展的过程。1990 年，国际标准化组织 ISO(International Standard Organization)将 C89 作为国际标准。此后又经历了几次修订与增补，最新的标准版本是 C99。本书参照 C99 的标准来介绍 C 语言程序设计。

C 语言内部具有许多功能结构和应用特点，这些特点只有在学习 C 语言的过程中才能领悟，下面仅对 C 语言的一般特点作简要介绍。

(1) 语言简洁，使用灵活，易于学习和使用。C 语句接近于自然语言和数学语言，功能与语义紧密联系，容易理解和应用。

(2) 数据类型丰富。C 语言包含丰富的数据类型，涵盖了实际中所面临的各种数据结构，几乎不需要用户自己定义数据结构，给程序设计带来了很大便利。

(3) 数据处理功能强。C 语言包含数值运算符、字符运算符、逻辑运算符和字/位运算符，它们与丰富的数据类型相结合，构成了灵活多样的表达式，可以实现其他高级语言难以实现的运算。

(4) 具有低级语言对计算机的一些硬件资源进行控制的功能。C 语言不但具有其他高级