高等学校计算机专业
面向项目实践规划教材

C语言程序设计项目式教程

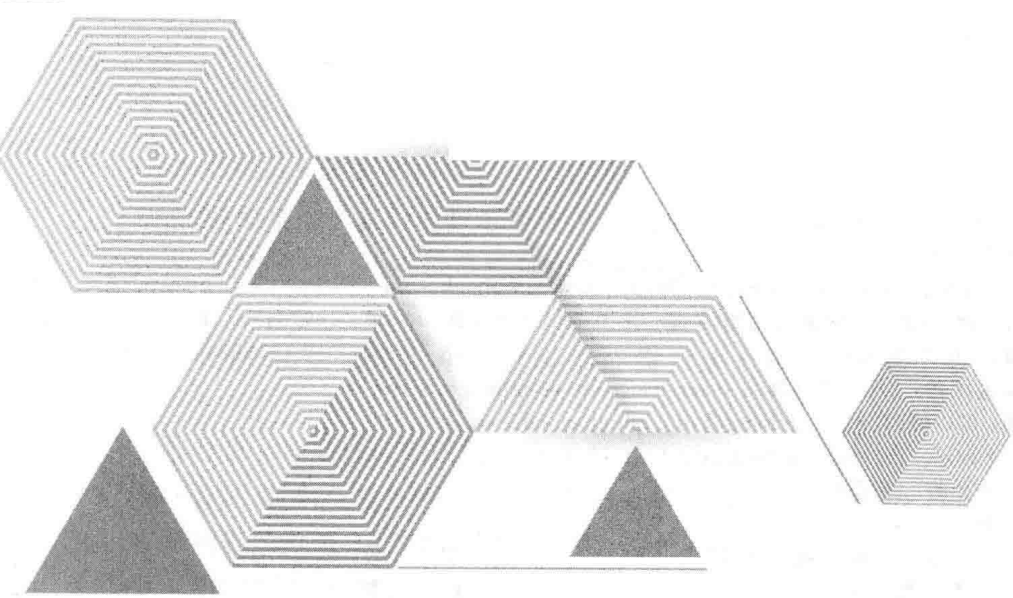
◎ 巨同升 李业刚 李增祥 编著

清华大学出版社

高等学校计算机专业
面向项目实践规划教材

C语言程序设计项目式教程

◎ 巨同升 李业刚 李增祥 编著



清華大學出版社
北京

内 容 简 介

本书在教学内容的编排上，采用“项目驱动知识”的方式，即根据每一章项目案例的需求，合理地安排每一个知识主题的切入点，从而将C语言中枯燥难懂的语法知识分解到全书各章中，并力求通过程序实例归纳出来。

本书在讲解程序实例时，采用“逐步构造法”写出程序，即通过编程思路、算法设计、程序原型等环节一步一步地构造出完整的程序，从而加深读者对编程方法的理解和掌握。

在本书的各主要章节中，分别提供了若干个项目式案例，供读者学习参考之用。仔细研究这些案例，将有助于提高读者的程序设计能力。

本书内容依据当前最新版的C语言参考手册编写，兼顾C89与C99标准。内容包括引论、基本数据与运算、顺序结构程序设计、选择结构程序设计、循环结构程序设计、数组、指针、字符与字符串处理、函数、函数的进一步讨论、编译预处理命令、结构体与共用体、位运算、文件等。

本书依据Visual C++2010 Express和DEV C++5.11集成开发环境进行讲述，符合当前软件的发展趋势，便于读者上机调试程序。

本书教学内容的编排顺畅合理，编程方法的讲解新颖独特，特别适合于初学者自学。本书可作为高等院校各专业学生学习C语言程序设计的教材和参考书。

图书在版编目(CIP)数据

C语言程序设计项目式教程/巨同升，李业刚，李增祥编著. —北京：清华大学出版社，2018
(高等学校计算机专业面向项目实践规划教材)
ISBN 978-7-302-48929-0

Ⅰ. ①C…　Ⅱ. ①巨…　②李…　③李…　Ⅲ. ①C语言－程序设计－教材　Ⅳ. ①TP312.8

中国版本图书馆CIP数据核字(2017)第285901号

责任编辑：贾　斌　薛　阳
封面设计：刘　键
责任校对：焦丽丽
责任印制：李红英

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 装 者：北京鑫海金澳胶印有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：18.75　　**字　　数**：454千字
版　　次：2018年2月第1版　　**印　　次**：2018年2月第1次印刷
印　　数：1～1500
定　　价：49.00元

产品编号：071599-01

探寻C语言学习之道

C语言作为一门专业型的语言，具有功能强大、运行效率高、实用性强等特点。但是，若作为学习程序设计的入门语言，C语言却存在着诸多不足之处，比如C语言的语法过于灵活，C语言的指针功能过于强大等。凡此种种，往往会给初学者造成诸多的困惑，甚至严重打击初学者学习程序设计的自信心。

C语言难学似乎是初学者的一个共识，那么C语言到底难在哪里，如何才能破解C语言难学的困局呢？其实，C语言学习的难点主要在于其语法。而造成C语言语法难学的主要因素包括以下几方面：

(1) C语言提供了多种功能独特的运算符，诸如自增(自减)运算符、复合赋值运算符、条件运算符、逗号运算符、位运算符等。

(2) C语言允许将赋值表达式、自增(自减)表达式嵌入到其他表达式中，导致了C语言语句的表达形式灵活多变。

(3) C语言中指针的使用无处不在、功能异常强大。

(4) C语言中大大扩展了逻辑运算量的类型。

(5) C语言中大括号与分号的位置，若稍有变化，则往往会导致完全不同的含义。

以上特色一方面造就了C语言优异的性能，另一方面也给初学者埋设了诸多的“陷阱”。

下面从三个方面探寻正确的C语言学习之道。

1. 如何学习C语言的语法

其实，只要采取了正确的学习策略，C语言语法难学的问题是可以破解的。

首先，需要明确语法在程序设计中的地位。学习C语言的最终目的是为了学会编写程序解决现实问题，因此编程能力的培养是学习的核心。而语法是编程的基础，是为编程服务的，因此语法的学习应当紧紧围绕编程这个核心，脱离了编程的语法是毫无意义的。

是不是说必须系统地牢固地掌握了C语言的语法，才能学好编程呢？其实并非如此。对于程序设计来说，更重要的是确定编程的总体思路或者说是算法，而不是具体实现中的语法。既往的经验表明，只需要掌握少量最常规的语法，就可以编写出解决一般问题的程序。至于有些非常规语法，即使是专业的程序员都极少用到，更别说是初学者了。

因此，在学习时不要过于看重语法知识的系统性与连贯性，而应当根据程序设计的需求，循序渐进地积累语法知识。例如，C语言中的运算符与表达式特别丰富，若将这些内容集中到一章中学习，则既枯燥乏味，又难以深入理解；若根据语法与程序设计的内在联系，

将这些内容分布到适当的章节中讲述，则既容易理解，又便于学以致用。例如，自增（自减）运算符和逗号运算符在学习循环结构之前就几乎不会用到，完全可以延后到循环程序部分再学习。

初学者应当优先学习那些既容易理解又频繁使用的常规语法，而应尽量避免研究那些既晦涩难懂又极少使用的非常规语法。比如，形如"j=i++ +i++ +i++、a+=a-=a+a"这样的表达式，在实际编程中几乎不可能出现，因此并无研究的必要。再比如，printf函数中各种格式说明符的详尽用法、整型数据的内存表示形式及相互转化、扩展的逻辑运算量及逻辑运算的短路、for语句的各种变式、通过指针引用二维数组的元素、行指针变量、指向函数的指针、链表等，这些内容初学者最好暂时不要深究。

当然，并不是说完全不研究这些非常规语法，而是要选择恰当的学习方式和时机。正确的学习方式是在编程实践中研究语法，包括在阅读其他人写好的程序时发现语法知识点，以及在自己编写程序、调试程序的过程中查阅并掌握需要用到的语法知识。这种方式具有更好的针对性，因而能够获得更好的学习效果。而正确的学习时机，则是在比较熟练地掌握了常规语法并能够编写一般难度的程序之后，再来研究这些非常规语法。采取这种策略相当于降低了知识之间的跨度，从而能够更好地理解和掌握知识。

2. 如何培养基本的编程能力

编程能力的培养需要一个长期积累的过程。那么，如何才能逐步地积累编程的经验呢？

首先是要尽量多地阅读其他人写好的程序，能够看懂程序实现的功能，分析出每条语句的作用，即如何一步步实现程序功能的。

然后上机调试阅读过的程序，从最简单的程序入手，将程序代码一条一条地录入、编辑，然后编译、运行。在调试程序的过程中，能够发现在书面上静态分析程序时所不能发现的问题，然后经过查阅资料、主动思考、改正错误的过程，即可获取新的知识和技能。这种收获是仅仅通过书面方式学习所不能得到的。因此，可以说不厌其烦地反复调试程序是学好程序设计的制胜法宝，这种说法一点都不为过。

在不断阅读已有程序的同时，还要经常地自己编写程序。从模仿已有的程序入手，尝试编写简单的程序。编写程序的过程最好在电脑上完成，一边编写、一边调试运行，然后根据调试中发现的问题及时地修正程序。在不断地改正错误的过程中，编程能力将会得到有效的提高。

对于具有一定复杂度的程序，可以首先尝试实现其中的一部分功能，待现有的程序调试运行成功之后，再在此基础上扩展一部分功能，然后如此循环往复，直至最终获得功能完善的程序。

3. 如何让编程水平更上一层楼

在具备了基本的编程能力之后，如何才能使得自己的编程水平更上一层楼呢？将程序设计应用于解决现实问题是提高编程能力的行之有效的方法，而面向项目的学习就是一种体现这种思路的卓有成效的培养学生综合分析问题、解决问题能力的教学模式。

所谓项目，是指来源于现实中的具有一定复杂度的问题，通常需要学生运用多方面的知

识综合分析、统筹规划，才能解决。

面向项目的学习，需要学生自行查阅资料，准备与项目相关的知识。通过这种方式所获得的是最牢固的最有机的知识，更重要的是提高了学生自主学习的能力。来自于现实中的项目，往往是错综复杂的，在分析问题的过程中需要舍弃非本质的内容，提取出本质的核心问题，从而可以培养学生综合分析问题、统筹规划和解决复杂问题的能力。

前言

FOREWORD

C语言是目前世界上使用最广的高级程序设计语言，被广泛地应用于系统程序设计、数值计算、自动控制等诸多领域。

C语言的产生颇为有趣，C语言实际上是UNIX操作系统的一个副产品。1972年，美国贝尔实验室的Dennis Ritchie为了开发UNIX操作系统，专门设计了一种新的语言——C语言。由于C语言具有强大的功能和很高的运行效率，兼具高级语言的直观性与低级语言的硬件访问能力，因而很快从贝尔实验室进入了广大程序员的编程世界。

由于Dennis Ritchie设计C语言的初衷是用于开发UNIX操作系统，因此C语言称得上是一门专业语言。这使得C语言在具有强大的功能和较高的运行效率的同时，也在一定程度上存在语法晦涩难懂、不便于初学者掌握的不足之处。

因此，C语言似乎不太适合作为程序设计初学者的入门语言。不过在现代人效率观念的驱使下，仍有许多学校将C语言选作初学者的入门语言。

其实，这样选择也未尝不可。只不过在教学中应当思考如何采取有效的应对策略，使初学者避开那些晦涩难懂的语法，从C语言最基本、最实用的编程方法入手，力争使学习者尽快地学会程序设计的基本方法，进而达到应用编程解决实际问题的境界。

从学习者的角度来说，要注意抓住C语言学习的要害所在——编程方法，而不要沉溺于C语言的语法细节之中。因为学习C语言的目的是为了编写程序解决实际问题，而过于细致地研究C语言的语法对于提高编程能力并没有太大的帮助。

针对上述问题，本书作者在教学内容的编排上，采用了“项目驱动知识”的方式，即根据各章项目案例的需求，合理地安排每一个知识主题的切入点，从而将C语言中枯燥难懂的语法知识分解到全书各章中，并力求通过程序实例归纳出来。

本书在讲解程序实例时，采用“逐步构造法”写出程序，即通过编程思路、算法设计、程序原型等环节一步一步地构造出完整的程序，从而加深读者对编程方法的理解和掌握。

学习知识的最终目的是运用知识解决现实中的问题，而面向项目的教学就是一种紧密结合现实问题的、能够有效地提高学习者综合分析问题和解决问题能力的教学模式。在本书的各主要章节中，分别提供了若干个项目式案例，供读者学习参考之用。仔细研究这些案例，将有助于提高读者的程序设计能力。

本书第1章、第2章由李业刚编写，第3章、第11章由李增祥编写，第13章、第14章由淄博技师学院史国兴编写，其余各章由巨同升编写。全书由巨同升统筹并定稿。

在本书的编写过程中，作者得到了山东理工大学计算机科学与技术学院广大同仁的大力支持与帮助，在此表示感谢。

由于作者水平所限，书中难免存在不足之处，敬请广大专家和读者批评指正。

编　者

2018 年 1 月于山东理工大学

目录

CONTENTS

第1章

引　论

计算机之所以能够解决五花八门的问题，主要是通过软件实现的。这是因为一种计算机的硬件系统一经设计完成，就是基本固定不变的；而如何充分发挥硬件系统的功能，则要完全依赖于软件。

软件是程序、数据及相关文档的集合，其中程序是软件的主体。

1.1　程序与程序设计语言

所谓程序就是用于完成特定任务的一组指令的序列。从广义上来说，一篇菜谱、一段操作指令都是程序；而狭义的程序则特指计算机程序。

编写计算机程序需要有专门的程序设计语言。程序设计语言的发展经历了机器语言、汇编语言和高级语言三个时代。

机器语言的指令都是二进制代码，非常不便于编程者使用，而且其程序难以在具有不同指令系统的计算机之间移植。

汇编语言有所改进，其指令采用助记符表示，使用起来直观一些。不过，汇编语言指令与机器语言指令本质上是相同的，其程序仍然难以在具有不同指令系统的计算机之间移植。因而，机器语言与汇编语言统称为低级语言。

为了克服低级语言的缺点，从20世纪50年代开始，计算机科学家着手研究设计一种更加通用的、与具体计算机硬件无关的、表达方式接近于人类自然语言和数学语言的程序设计语言，这种语言称为高级语言。

从那时开始，计算机专家们陆陆续续地研究开发了数以千计的程序设计语言。就目前而言，常用的程序设计语言也有数十种之多。

各种程序设计语言的使用排名情况，可以查看TIOBE网站(https://www.tiobe.com/tiobe-index)的程序设计语言社区排行榜。

1.2　C语言的发展及特点

1.2.1　C语言的发展

在程序设计语言的发展历程中，从来没有一种语言像C语言一样具有如此广泛而长久

的影响力。主要体现在如下三个方面：

(1) C语言是编写操作系统的第一选择，同时也是编写其他系统软件的优先选择。

(2) C语言在嵌入式系统程序设计方面具有独特的地位，C语言是除了汇编语言之外使用最多的单片机编程语言。

(3) 鉴于C语言获得了巨大的成功，人们相继开发了许许多多的“C-like”程序设计语言，从而构成了庞大的C家族，包括C++、C#、Objective-C、Java、PHP、Swift等。

C语言的诞生颇为有趣，C语言实际上是UNIX操作系统的一个副产品。1972年，美国贝尔实验室的Dennis Ritchie为了开发UNIX操作系统，专门设计出了一种新的程序设计语言——C语言。由于C语言具有强大的功能、较高的运行效率，兼具高级语言的直观性与低级语言的硬件访问能力，因而很快从贝尔实验室进入了广大程序员的编程世界。

1.2.2 C语言的标准化

1. 传统C语言

早期的C语言没有统一的标准和规范，直到1978年Brain Kernighan和Dennis Ritchie合著的 *The C Programming Language* 一书出版，才使这种状况得以改变。在这本书中定义的语法很快就成了当时事实上的C语言规范，通常将这种规范称为“传统C语言”。

2. C89标准

在C语言的发展过程中，出现了很多C语言的“方言”版本，C语言的标准化成为一个紧迫的问题。1983年，美国国家标准化协会(ANSI)开始着手进行C语言的标准化工作，直到1989年12月颁布实施。1990年，国际标准化组织接受了这一标准，并将其颁布为国际标准。通常将这一标准称为“C89标准”。

目前，几乎所有的C语言编译器都能够支持C89标准。

3. C99标准

1999年，国际标准化组织再次颁布了新的C语言标准，通常称之为“C99标准”。目前，大部分C语言编译器都能够支持C99标准。

4. C11标准

2011年，国际标准化组织再次颁布了迄今为止最新的C语言标准，通常称之为“C11标准”。目前，几种主流的C语言编译器能够部分地支持C11标准。

本书内容主要依据C89标准讲解，部分内容同时兼顾C99标准。

1.2.3 C语言的特点

C语言之所以具有如此强大持久的生命力，成为经久不衰、最受欢迎的程序设计语言之一，是由C语言自身所具有的如下显著特点所决定的。

1. 简洁

C语言的关键字与保留标识符特别简短，并采用一对大括号定义程序块，从而使得C语言的程序异常简洁。

2. 灵活

由于C语言定义了若干功能独特的运算符，加之赋值运算可以嵌入到其他表达式中，