普通高等教育“十一五”国家级规划教材

HZ BOOKS 华章教育

高等院校精品课程系列教材·省级

C语言程序设计教程

第4版

朱鸣华 罗晓芳 董明 孟军 汪德刚◎编著

精品课程

C Programming

Fourth Edition

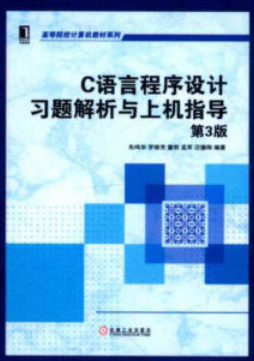
机械工业出版社
China Machine Press

C语言是一种应用非常广泛的结构化程序设计语言，既适合于编写应用程序，又特别适合于编写系统软件。

本书是结合作者多年的教学经验编写而成的，在内容编排上尽量体现出易学的特点，在文字叙述上力求条理清晰，在教材体系上重视理论与实践相结合，以便于读者低起点、高效率地掌握C语言的编程技巧。

本版在第3版的基础上进行了如下的修订和调整：

- 程序设计语言平台升级为Visual C++ 2010，书中实例均在Visual C++ 2010环境下调试通过，并在《C语言程序设计习题解析与上机指导 第3版》中对Visual C++ 2010环境进行了详细的介绍，方便学生自主学习。
- 增加了部分章节中的课后习题以及趣味程序设计实例，以激发学生的学习兴趣。
- 为满足学时的安排和教学需要，重新调整了章节的组合，删除了第2章“算法与程序设计基础”和第12章“面向对象程序设计与C++基础”。



C语言程序设计习题解析与上机指导 第3版
书号：978-7-111-63270-2
定价：49.00元

上架指导：计算机\程序设计
ISBN 978-7-111-63415-7
9 787111 634157 >
定价：59.00元

投稿热线：(010) 88379604
读者信箱：hzjsj@hzbook.com
客服电话：(010) 88361066 88379833 68326294

华章网站：www.hzbook.com
网上购书：www.china-pub.com
数字阅读：www.hzmedia.com.cn

普通高等教育“十一五”国家级规划教材

高等院校精品课程系列教材·省级

C语言程序设计教程

第4版

朱鸣华 罗晓芳 董明 孟军 汪德刚 编著

精品课程

C Programming

Fourth Edition

机械工业出版社
China Machine Press

图书在版编目（CIP）数据

C 语言程序设计教程 / 朱鸣华等编著 . —4 版 .—北京：机械工业出版社，2019.8（2019.12 重印）
（高等院校精品课程系列教材）

ISBN 978-7-111-63415-7

Ⅰ. C… Ⅱ. 朱… Ⅲ. C 语言 - 程序设计 - 高等学校 - 教材 Ⅳ. TP312.8

中国版本图书馆 CIP 数据核字（2019）第 164826 号

本书介绍利用 C 语言进行程序设计的基本知识。全书共 11 章，主要内容包括：C 语言的基本概念，数据类型、运算符与表达式，数据的输入和输出，选择结构，循环结构，数组，函数，编译预处理，指针，结构体与共用体，文件等。每章配有大量的习题，便于读者巩固所学知识，掌握程序设计的基本方法和编程技巧。

本书力求概念叙述准确、严谨，语言通俗易懂，适合作为高等院校理工科非计算机专业的 C 语言程序设计课程教材，也可供工程技术人员参考。

出版发行：机械工业出版社（北京市西城区百万庄大街 22 号　邮政编码：100037）

责任编辑：迟振春	责任校对：李秋荣
印　刷：中国电影出版社印刷厂	版　次：2019 年 12 月第 4 版第 2 次印刷
开　本：185mm × 260mm　1/16	印　张：18
书　号：ISBN 978-7-111-63415-7	定　价：59.00 元

客服电话：（010）88361066　88379833　68326294　投稿热线：（010）88379604
华章网站：www.hzbook.com　读者信箱：hzjsj@hzbook.com

前　言

《C语言程序设计教程》自2007年2月出版发行第1版以来，被多所学校程序设计课程选用，是学习C语言程序设计的理想教材。

为了适应计算机科学技术的发展，更好地满足人工智能、互联网+形势下高校计算机教学的需求，本教材进行了第4版修订。第4版共分11章，在原教材的基础上进行了语言平台的升级及内容的修订，主要调整如下：

1）程序设计语言平台升级为Visual C++ 2010，书中实例均在Visual C++ 2010环境下调试通过，并在《C语言程序设计习题解析与上机指导　第3版》中对Visual C++ 2010环境进行了详细的介绍，方便学生自主学习。

2）对各章节的文字叙述进行了完善和修改。

3）增加了部分章节中的课后习题，以及趣味程序设计实例，以激发学生的学习兴趣。

4）每章都配有精心设计的例题和习题，并配有实验指导教材。

5）为满足学时的安排和教学需要，重新调整了章节的组合，删除了第2章“算法与程序设计基础”和第12章“面向对象程序设计与C++基础”。

第4版秉承原来版本内容全面、衔接有序、通俗易懂、习题丰富以及实践性强的特点，符合社会发展的需要，便于高校程序设计课程的教学安排，以及结合与之配套的国家精品在线开放课程开展线上线下混合式教学实践。

本书第1～3和5章由罗晓芳编写，第4、7和8章由朱鸣华、汪德刚编写，第6和11章由董明编写，第9和10章由孟军编写，全书由朱鸣华、罗晓芳统稿。

第4版的修订是在第3版的基础上进行的，感谢参与第3版编写工作的刘旭麟、李慧、杨微、孙大为、赵晶。在本书的编写过程中还得到了大连理工大学程序设计基础课程教学团队各位老师的大力支持和帮助，在此表示诚挚的谢意。由于编者水平有限，书中难免存在疏漏和谬误之处，敬请广大读者指正。

编　者

2019年5月

目　录

第1章

C语言概述

1.1 程序设计的基本概念

计算机的产生是20世纪重大的科技成果之一。计算机的飞速发展大大促进了知识经济的发展和社会信息化的进程。人与计算机交流最通用的手段是程序设计语言。当人们想利用计算机解决某个问题时，必须用程序设计语言安排好处理步骤，并存入计算机内供计算机执行，这些用程序设计语言安排好的处理步骤称为**计算机程序**，程序是计算机操作指令的集合。

一个计算机程序主要包括两方面的内容：其一是关于程序实现算法的操作步骤描述，即动作描述；其二是关于算法操作对象的描述，即数据描述。曾经发明Pascal语言的著名计算机科学家沃思（Niklaus Wirth）关于程序设计提出了一个著名的公式：

程序 = 算法 + 数据结构

这个公式说明了程序设计的主要任务，也说明了在程序设计过程中“算法”与“数据结构”密不可分的关系。用程序设计语言编制一个能完成某项任务的计算机程序的过程叫作**程序设计**。用计算机解决一个实际问题，首先应进行程序设计，而程序设计主要包括对数据以及处理问题的方法和步骤的完整而准确的描述。

数据是操作的对象，操作的目的是对数据进行处理，以得到期望的结果。对数据的描述，就是指明在程序中要用到数据的哪些类型和组织形式，即数据结构；对问题的方法和步骤的描述，即计算机进行操作的步骤，也就是所采用的算法。

对于程序设计初学者来说，要学会如何设计一个正确的程序，首先要认真考虑和设计数据结构及操作步骤。一个正确的程序通常包含两方面的含义：一是书写正确，二是结果正确。书写正确是程序在语法上正确，符合程序语言的规则；而结果正确通常是指对应于正确的输入，程序能够产生所期望的输出。程序设计除了以上两大要素之外，还涉及程序设计的思想和所用的具体语言工具及环境，可以详细地描述为：

程序设计 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

这4个方面是程序设计人员应具备的基本知识。其中，算法是灵魂，解决“做什么”和“怎么做”的问题，不了解算法就谈不上程序设计。程序中的操作语句就是对算法的实现。算法是

从计算机操作的角度对解题过程的抽象，数据结构是从如何组织被处理对象的角度进行抽象。本书不是以数据结构和算法为主展开讨论的，而是着重介绍利用 C 语言进行程序设计的基本方法。

程序设计方法是从宏观的角度处理问题的方法，如结构化程序设计、面向对象的技术等。工具包括使用的程序设计语言及相关的编译系统和调试工具。

程序设计的另一个关键是必须选择且掌握一种程序设计语言，因为程序设计语言是人和计算机直接交流的工具。

程序设计语言通常分为三类：机器语言、汇编语言和高级语言。

机器语言是指由二进制代码组成的，不需翻译就可以被计算机直接执行的指令的集合。这是一种面向机器的语言，执行效率高，但通用性和可读性都很差。

为了克服这些缺点，产生出一种符号语言，也称为**汇编语言**。对于用汇编语言编写的程序，计算机不能直接识别，需要汇编程序把它翻译成机器代码。它比机器语言使用起来方便，但通用性仍然很差。

人们把直接表示数学公式和解题方法的语言称为**高级语言**。这种语言直观通俗，非常接近于人们的“自然描述”语言，便于编写、阅读、修改和维护，通用性强。用高级语言编写的源程序，机器也是不能识别的，必须通过编译程序或解释程序进行翻译，最终生成机器语言程序。目前程序设计语言有很多，新的语言不断涌现。各类语言都有其特点和适用的领域。本书介绍 C 语言。

最后一项指的是，要选择一个合适的**集成开发环境**（Integrated Development Environment，IDE）。IDE 是集成了程序员语言开发中需要的一些基本工具、基本环境和其他辅助功能的应用软件。IDE 一般包含四个主要组件：**源代码编辑器**（editor）、**编译器**（compiler）、**解释器**（interpreter）和**调试器**（debugger）。开发人员可以通过图形用户界面（GUI）访问这些组件，并且实现整个代码编译、调试和执行的过程。现在的 IDE 也提供帮助程序员提高开发效率的一些高级辅助功能，比如代码高亮、代码补全和提示、语法错误提示、函数追踪、断点调试等。

1.2 C 语言发展简史

C 语言是一种通用的程序设计语言，由于它很适合用来编写编译器、操作系统，并进行嵌入式系统开发，因此被称为“系统编程语言”，但它同样适用于编写不同领域中的应用程序。

1. C 语言的产生

C 语言是一种被广泛应用的计算机高级程序设计语言，是在 B 语言的基础上发展起来的，它经历了不同的发展阶段。

早期的系统软件设计均采用汇编语言，例如，大家熟知的 UNIX 操作系统。尽管汇编语言在可移植性、可维护性和描述问题的效率等方面远远不及高级程序设计语言，但是一般的高级语言有时难以实现汇编语言的某些功能。

那么，能否设计出一种集汇编语言与高级语言的优点于一身的语言呢？这种思路促成了 UNIX 系统的开发者（美国贝尔实验室的 Ken Thompson）于 1970 年设计出了既简单又便于硬件操作的 B 语言，并用 B 语言写了第一个 UNIX 操作系统，这个操作系统先在 PDP-7 上实现，1971 年又在 PDP-11/20 上实现。

B 语言的前身是 BCPL（Basic Combined Programming Language），它是英国剑桥大学的 Martin Richards 在 1967 年基于 CPL 语言设计的，而 CPL 语言又是在 1963 年基于 ALGOL 60 产生的。

1972 ～ 1973 年，贝尔实验室的 D. M. Ritchie 在 B 语言的基础上设计出 C 语言，该语言弥补了 B 语言过于简单、功能有限的不足。

1973 年，Ken Thompson 和 D. M. Ritchie 合作将 90% 以上的 UNIX 代码用 C 改写。随着改写 UNIX 操作系统的成功，C 语言也逐渐被人们接受。

1987 年以后，C 语言已先后被移植到大、中、小、微型机上，并独立于 UNIX 和 PDP，从而得到了广泛应用。

2. C 语言的发展和应用

1978 年，B. W. Kernighan 和 D. M. Ritchie 合写了一本经典著作——《C 程序设计语言》(The C Programming Language，中文版、影印版均已由机械工业出版社引进出版)，它奠定了 C 语言的基础，被称为标准 C。

1983 年，美国国家标准学会（ANSI）根据 C 语言问世以来的各种版本对 C 的发展和扩充，制定了新的标准，称为 ANSI C。1987 年又公布了新标准，称为 87 ANSI C。目前流行的多种版本的 C 语言编译系统都是以此为基础的。

在 ANSI 标准化后，C 语言的标准在相当一段时间内都保持不变，直到 20 世纪 90 年代才进行了改进，这就是 ISO 9899:1999（1999 年出版）。这个版本就是通常提及的 C99。它于 2000 年 3 月被 ANSI 采用。

3. C 语言和 C++ 语言交融发展

由于 C 语言是面向过程的结构化和模块化的程序设计语言，当处理的问题比较复杂、规模庞大时，就显现出一些不足，由此面向对象的程序设计语言 C++ 应运而生。C++ 的基础是 C，它保留了 C 的所有优点，增加了面向对象机制，并且与 C 完全兼容。绝大多数 C 语言程序可以不经修改直接在 C++ 环境中运行。

1.3　C 语言的特点

C 语言作为一种古老而常青的经典编程语言，具备了现代程序设计的基本结构和元素，其语法是许多语言的基础。目前 C 语言在各类语言排行榜中始终名列前茅，它具有以下优点：

1）**兼具高级、低级语言的双重能力**。C 语言允许直接访问物理地址，能进行位操作，能实现汇编语言的大部分功能，可以直接对硬件进行操作，所以又被称为中级语言。

2）**生成的目标代码质量好，程序执行效率高**。C 语言具有汇编语言的许多特性，一般只比汇编程序生成的目标代码效率低 10% ～ 20%，可以开发出执行速度很快的程序。

3）**语言简洁，结构清晰**。C 程序通常是由若干函数组成的，强大的函数功能为程序的模块化和结构化提供了保证，因此程序简洁清晰，可读性强。

4）**语言表达能力强**。C 语言运算符丰富，例如，在 C 语言中，把括号、赋值、强制类型转换等都作为运算符处理。C 语言具有现代化语言的各种数据结构，如整型、字符型、数组型、指针型、结构体和共用体等，而且具有结构化的控制语句。

5）**程序通用性、可移植性好**。C 语言没有依赖于硬件的输入 / 输出语句，而采用系统的库函数进行输入 / 输出操作，因此 C 语言不依赖于任何硬件系统，这种特性使得用 C 语言编写的程序很容易移植到其他环境中。

当然，C 语言也有自身的不足，和其他高级语言相比，其语法限制不太严格，例如，对变量的类型约束不严格，影响程序的安全性，对数组下标越界不进行检查等。从应用的角度来看，C 语言比其他高级语言较难掌握。

总之，C 语言既具有高级语言的特点，又具有汇编语言的特点；既是一个成功的系统设计语言，又是一个实用的程序设计语言；既能用来编写不依赖计算机硬件的应用程序，又能用来编写各种系统程序。它是一种深受欢迎、应用广泛的程序设计语言。

1.4 简单C语言程序举例

在这一节中，我们通过两个简单的C语言程序例子来介绍C语言的程序结构，并对C语言的基本语法成分进行相应的说明，以便使读者对C语言程序有一个大致了解。

【例1-1】 计算矩形的面积。

```
#include <stdio.h>                          //1：编译预处理
int main( )                                 //2：主函数
{                                           //3：函数体开始
    float  h, w, area;                      //4：声明部分，定义变量
    h=10.5;                                 //5：以下4条C语句为执行部分，给变量h和w赋值
    w=20.5;
    area=h*w;                               //7：计算矩形的面积
    printf("area=%6.2f\n", area);           //8：输出area的值
    return 0;                               //9：返回值为0
}                                           //10：函数体结束
```

运行结果：

```
area=215.25
```

每行中以“//”开始的右边的文本表示程序注释的内容。

第1行：是一个编译预处理，在程序编译前执行，指示编译程序如何对源程序进行处理。它以“#”开头，结尾不加分号，以示和C语句的区别。

第2行：main表示主函数，每一个C程序都必须有一个主函数，int表示主函数为整型。函数体由第3行和第9行的一对花括号括起来。

第4行：是变量声明部分，定义变量h、w和area为实型变量。

第5和6行：是两条赋值语句，给变量h赋值10.5，w赋值20.5。

第7行：将算术表达式h*w的值赋予变量area。

第8行：调用函数printf输出矩形面积值。

第9行：向操作系统返回一个零值，如果程序不能正常执行，则会自动向操作系统返回一个非零值，一般为-1。

上面的主函数构成了一个完整的程序，称为**源程序**。它以文件的方式存在，文件中包含函数的源程序代码。C语言规定保存C源程序文件的扩展名为“.c”。

【例1-2】 计算两个矩形的面积之和。

```
/*本程序用来计算两个矩形的面积之和。包括主函数int main( )、
一个子函数double  area(double  h, double  w )*/
#include <stdio.h>                          //1：编译预处理
double  area(double  h, double  w )         //2：定义函数area
{
  double s;
  s=h*w;
  return s;                                 //6：返回s的值，return是关键字
}
int main( )                                 //8：主函数
{
  double  h1, h2, w1, w2, s1, s2;           //10：声明部分，定义变量
  h1=10.5; w1=20.5;
  h2=1.5*h1; w2=1.5*w1;                     //12：计算变量h2，w2的值
  s1=area(h1, w1);                          //13：调用area函数，将得到的返回值赋给变量s1
  s2=area(h2, w2);
  printf("area=%6.2f \n ", s1+s2);          //15：输出两个矩形的面积之和
  return 0;
}
```

运行结果：

```
area=699.56
```

本程序包括主函数 main、函数 area（被主函数调用）和一个编译预处理指令。

最前面两行中 /*……*/ 内的文本也表示程序注释的内容，这种方式一般用于表示多行注释。

第 2 行：从该行开始到第 6 行定义函数 area，包括函数类型、函数名和函数体等部分。

第 13 和 14 行：调用函数 area，将两次调用的返回值分别赋给变量 s1 和 s2。

第 15 行：计算并输出两个矩形的面积之和。

上面两个函数构成了一个完整的程序，称为**源程序**。可以把这两个函数放在一个文件中，当程序语句多的时候也可以分别以函数为单位放在两个以上的文件中，保存 C 源程序文件的扩展名为“.c”。

1.5　C 语言程序的组成与结构

通过以上两个例子，我们对 C 语言程序的组成和结构有了初步和直观的了解，总结如下：

1）一个 C 语言程序的主体结构是由一个或若干个函数构成的。这些函数的代码以一个或若干个文件的形式保存。这些函数中必须有且只能有一个名为 main 的主函数。

2）主函数 main 是程序的入口，它可以出现在程序的任何位置。一个 C 程序总是从主函数 main 开始执行，最后结束于主函数。

3）C 程序中的函数包括：主函数 main，用户自定义函数（例如，例 1-2 中的 area），系统提供的库函数（例如，输出函数 printf）。

4）函数由函数头和函数体两部分组成，函数头由函数类型的定义、函数名和参数表组成，函数体由声明部分（所使用变量和函数的说明）和若干执行语句组成。

5）语句由关键字和表达式组成，每个语句和声明部分的结尾都必须加分号。复合语句的开头和结尾使用左、右花括号 { }。

关键字是由 C 语言系统规定的具有特定功能的固定字母组合。例如，例 1-2 中的 int、double 和 return 就是关键字。

用运算符将操作对象连接起来、符合 C 语言语法的式子称为**表达式**。表达式的组成元素有：变量、常量、函数调用、运算符。这些组成元素是以标识符和关键字等形式存在的。例如，例 1-2 中的 h2=1.5*h1 和 w1=20.5 都是表达式。

6）程序中“/*　*/”内的文字是程序的注释部分，是便于阅读理解程序的解释性附加文本，程序编译器完全忽略注释部分的内容。此外，在程序调试时，也可以将一部分代码转换为注释保留，而不必删除，以提高程序调试的效率。

另外，一些 C 语言开发工具还支持用“//”标识注释部分，如果某行程序代码前面插入符号“//”，该符号后面的部分就变为注释行，并且本行有效，不能跨行。一般情况下，如果注释内容在程序中占用多行，习惯用“/*　*/”，而单行注释内容用“//”标识即可。

从以上分析可以发现，C 程序的组织和构造与日常文章的结构很类似，如表 1-1 所示。

表 1-1　文章和 C 语言对应的层次结构

文章的层次结构	对应 C 程序的层次结构	文章的层次结构	对应 C 程序的层次结构
文章	程序	句子	语句
章节	文件	词组	表达式
段落	函数	字	常量、变量、关键字、运算符等
开头第一段	主函数		

在一般语言的学习过程中，首先学字、词组，然后造句，阅读范文，最后写作文。现在学习计算机语言，我们也同样遵循这个规律，即先学习常量、变量的类型和定义方法，然后依次学习表达式、语句和函数等，同时阅读一些程序范例，最后编写程序。当然二者也有本质上的区别，一般语言的学习以形象思维为主，而计算机语言的学习是以逻辑思维为主。C 语言程序的层次结构如图 1-1 所示。

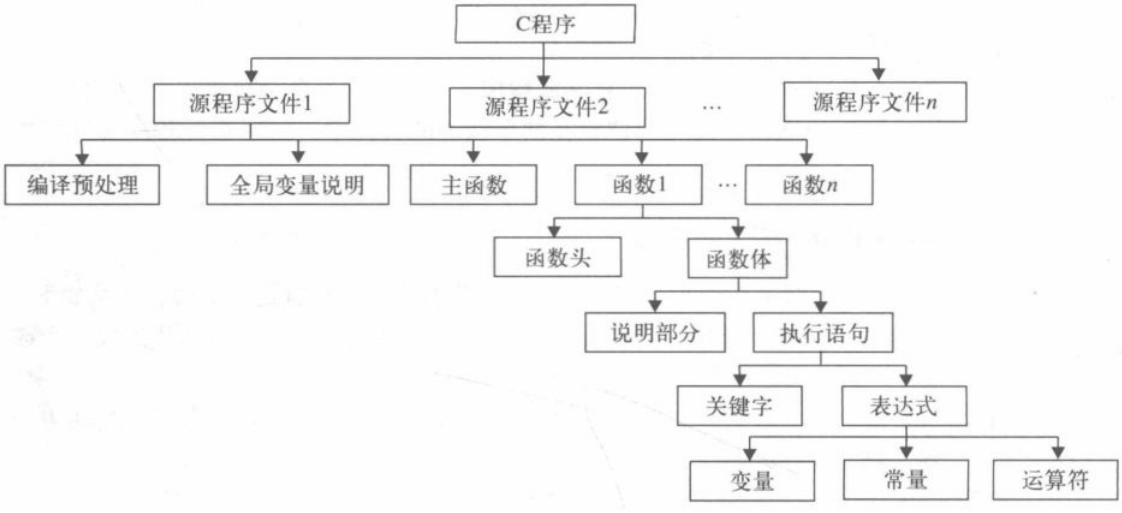


图 1-1 C 语言程序的层次结构

1.6 C 语言程序的开发步骤

一个 C 语言程序从最初编写到得到最终结果，大致经过以下几个步骤：

1）**编辑源程序**。选择一种 C 语言开发工具软件（IDE），输入编写好的程序代码，称之为源程序，它以文件的方式存在，文件的扩展名为“.c”。

2）**编译源程序**。为了使计算机能执行高级语言源程序，必须把源程序转换为二进制形式的目标程序，这个过程称为编译源程序。

编译是以源程序文件为单位分别进行的，每一个源程序文件对应生成一个目标文件，目标文件的扩展名为“.obj”。

编译过程中对源程序的全部内容进行检查，例如检查程序中关键字的拼写是否正确，根据程序的上下文检查语法是否有错等，编译结束后，系统显示所有的编译出错信息。

一般编译系统的出错信息有两种：一种是**错误**（error）信息，这类错误出现后，系统不生成目标文件，必须改正后重新编译；另一种是**警告**（warning）信息，是指一些不影响程序运行的不合理现象或轻微错误。例如，程序中定义了一个变量，却一直没有使用，出现这类警告信息，系统仍可以生成目标文件。

3）**连接目标文件**。编译结束，得到一个或多个目标文件，此时要用系统提供的“连接程序”（linker）将一个程序的所有目标文件和系统的库文件以及系统提供的其他信息连接起来，最终形成一个可执行的二进制文件，可执行文件的扩展名为“.exe”。

4）**运行程序**。运行最终形成的可执行文件，得到运行的结果。

5）**结果分析**。分析程序的运行结果，如果发现结果不对，应检查程序或算法是否有问题，修改程序后再重复上面的步骤。

C 语言程序的开发步骤如图 1-2 所示。

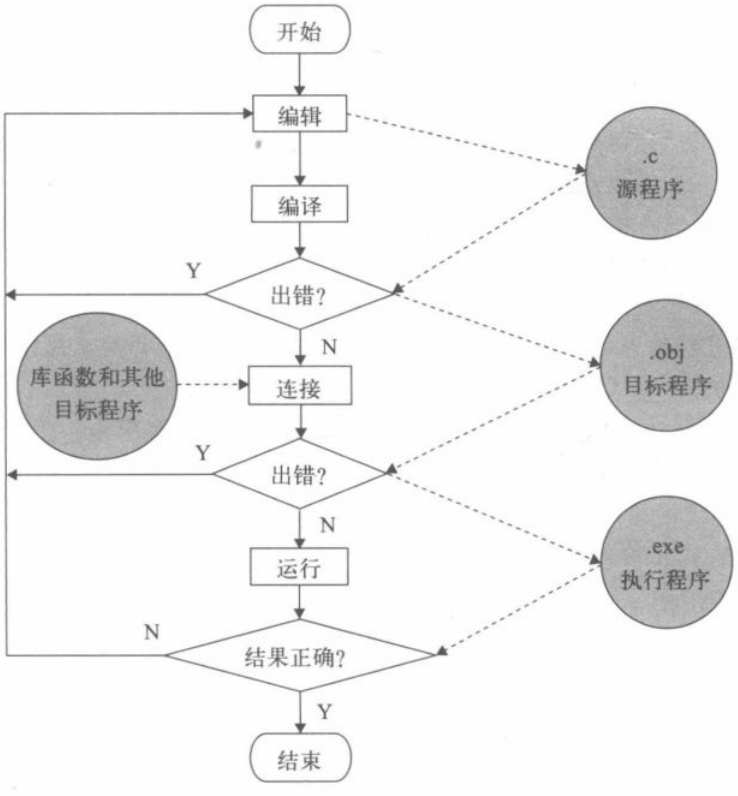


图 1-2　C 语言程序的开发步骤

小结

本章首先叙述了程序设计的基本概念以及 C 语言产生和发展的历史过程，然后与其他高级语言进行对比，列举了 C 语言的特点，再通过两个简单的 C 程序实例，描述了 C 语言程序的基本组成和结构特点，最后介绍了 C 语言程序开发各个步骤的内容。

通过本章的学习，读者应该对程序设计的概念有初步的认识，对 C 语言总体结构和开发步骤有初步的了解。建议学习本章内容后，尽快在计算机上编译、运行一个简单的 C 语言程序。在今后的学习中，读者会发现有些问题用文字叙述很难领会，但上机编程后，很容易理解，即所谓“在编程中学习编程”。

习题

一、简答题

简要回答下列问题。

1. 程序的定义是什么？程序主要由几部分组成？
2. C 语言的主要特点有哪些？
3. C 语言程序是由哪些部分组成的，各部分的作用是什么？

二、选择题

以下各题在给定的四个答案中选择一个正确答案。

1. 以下叙述正确的是（　　）。

A. C 语言允许直接访问物理地址，可以直接对硬件进行操作

B. C 语言程序不用编译，即可被计算机识别运行

C. C 语言不允许直接访问物理地址，不可以直接对硬件进行操作

D. C 语言程序只需编译，不需连接即可被计算机运行

2. 在一个 C 程序中（　　）。

A. main 函数出现在所有函数之前，C 程序不一定都有 main 函数

B. main 函数可以在任何地方出现，一个 C 程序必须有且仅有一个 main 函数

C. main 函数必须出现在所有函数之后，一个 C 程序只能有一个 main 函数

D. main 函数出现在固定位置，一个 C 程序可以有多个 main 函数

三、填空题

1. C 语言开发工具直接输入的程序代码是__A__文件，经过编译后生成的是__B__文件，经过连接后生成的是__C__文件。

2. C 语言源文件的后缀是__A__，经过编译后生成的文件的后缀是__B__，经过连接后生成的文件的后缀是__C__。

四、编写程序题

输入下面的程序，上机调试并运行。

```
#include  <stdio.h>
int main( )
{
   float  r, s;
   r=15.5;
   s=2*3.14*r;
   printf("r=%4.2f, s=%4.2f\n",  r,  s);
   return 0;
}
```