高职高专“十二五”
计算机类专业规划教材

C语言程序设计案例教程

童夏敏　张万臣　主　编
杨万成　王　冬　副主编

中国电力出版社
CHINA ELECTRIC POWER PRESS

高职高专“十二五”
计算机类专业规划教材

C语言程序设计案例教程

主　编　童夏敏　张万臣
副主编　杨万成　王　冬
编　写　王　磊　贾民政　梁　宇
主　审　郭景峰

中国电力出版社
CHINA ELECTRIC POWER PRESS

内 容 提 要

本书为高职高专“十二五”计算机类专业规划教材。全书充分结合高职高专学生特点，以程序设计为主线，采用案例驱动模式，通过案例和问题引入主要教学内容，重点讲解程序设计的思想和方法。主要内容包括 C 语言概述与数据类型、三种基本结构的程序设计方法、数组、函数、指针、结构体与共用体、文件等。

本书可作为高等职业技术学院、高等专科学校、成人高校及本科院校中的二级职业技术学院计算机及相关专业的教材，也可作为对 C 语言程序设计感兴趣的读者的自学用书。

图书在版编目（CIP）数据

C 语言程序设计案例教程 / 童夏敏，张万臣主编. —北京：中国电力出版社，2014.7

高职高专“十二五”计算机类专业规划教材

ISBN 978-7-5123-5730-3

Ⅰ. ①C… Ⅱ. ①童… ②张… Ⅲ. ①C 语言－程序设计－高等职业教育－教材 Ⅳ. ①TP312

中国版本图书馆 CIP 数据核字（2014）第 060077 号

中国电力出版社出版、发行

（北京市东城区北京站西街 19 号 100005 http://www.cepp.sgcc.com.cn）

航远印刷有限公司印刷

各地新华书店经售

*

2014 年 7 月第一版 2014 年 7 月北京第一次印刷

787 毫米×1092 毫米 16 开本 15 印张 358 千字

定价 **30.00** 元

丛书编委会成员

丛书编写院校名单

北京农业职业学院
北京印刷学院
北京信息职业技术学院
北京工业职业技术学院
北京电子科技职业学院
北京农业职业学院
江苏食品药品职业技术学院
江苏经贸职业技术学院
江苏农牧科技职业学院
常州机电职业技术学院
泰州师范高等专科学校
扬州市职业大学
徐州工程学院
南通市广播电视大学
南通职业大学
苏州市职业大学
义乌工商职业技术学院
浙江警官职业学院
南昌师范学院
萍乡高等专科学校
重庆文理学院
四川职业技术学院
四川工商职业技术学院
四川交通职业技术学院
成都职业技术学院
内江师范学院
攀枝花学院
武汉软件工程职业学院
山东服装职业学院
山东信息职业技术学院
山东大王职业学院
淄博职业学院
辽宁建筑职业学院
辽宁理工职业学院
营口职业技术学院
大连海洋大学职业技术学院
许昌学院
郑州升达经贸管理学院
郑州铁路职业技术学院
河南化工职业学院
黄河科技学院
河北建材职业技术学院
河北软件职业技术学院
廊坊职业技术学院
黄山学院
太原师范学院
肇庆工商职业技术学院
广东工程职业技术学院
佛山职业技术学院
广西经贸职业技术学院
新疆工程学院
珠海城市职业技术学院

前　言

C 语言是一种应用十分广泛的计算机语言，其功能丰富、表达能力强、使用灵活方便、应用面广、目标程序效率高、可移植性好，既具有高级语言的优势，又具有低级语言的许多特点，特别适合编写系统软件，已经成为计算机类本科生、高职高专学生及中专生的必修课。

本书内容丰富、结构清晰、图文并茂，易于教师进行教学与学生自学。全书采用案例驱动方式进行讲解，以程序案例为主导，将知识点融入实例，以案例带动知识点的学习。在按案例进行讲解时充分注意保证知识的相对完整性和系统性，使读者通过学习实例掌握 C 语言的程序设计方法和技巧。

全书共分 11 章，主要介绍了程序逻辑与程序设计语言、C 语言程序设计的基本概念、数据类型和运算符应用、三种基本结构的程序设计方法、数组、函数、指针、结构体与共用体、编译预处理和文件等。在每章之后提供的习题和实训内容，突出了实用性，强调理论与实践相结合，有助于培养学生解决实际问题的能力。

本书由长期从事“C 语言程序设计”课程教学的多位老师合作完成。由河北建材职业技术学院的童夏敏、张万臣主编，河北建材职业技术学院的杨万成、塔里木大学的王冬副主编，沧州职业技术学院的王磊、北京工业职业技术学院的贾民政、河南化工职业学院的梁宇编写。全书各章节的编写分工如下：第 2 章、第 4 章、第 5 章由童夏敏编写，第 1 章、第 8 章、第 9 章由张万臣编写，第 6 章、第 7 章、第 11 章由杨万成编写，第 3 章由王冬和王磊编写，第 10 章和附录由贾民政和梁宇编写。全书由童夏敏和张万臣统稿和定稿，由燕山大学郭景峰主审。

限于编者水平，书中不妥与疏漏之处在所难免，恳请专家和读者批评指正，可发邮件至 wangtong1971@126.com。

编　者

2014 年 1 月

目　录

第1章 程序与C语言简介

计算机是现代社会中的常用工具。无论是生活，还是工作，人们都离不开计算机。计算机如此重要，是因为计算机的功能十分强大，可以帮助人们做许多工作。那么，为什么计算机功能如此强大呢？原因有两方面：一方面是计算机硬件工作的速度越来越快，存储器容量也越来越大；另一方面是计算机的程序种类越来越多，程序的功能越来越强大。所以为了使计算机更好地为人类服务，需要为它编写各类不同的程序。编写程序时，要认真考虑清楚程序的数据结构和算法，并且还要用一种程序设计语言将程序表示出来。

本章的主要内容包括：

- 程序与程序设计
- 计算机语言
- C语言简介
- C语言开发环境与程序开发过程

1.1 程序与程序设计

1.1.1 程序

程序（Program）是为实现特定目标或解决特定问题而用计算机语言编写的命令序列的集合，是为实现预期目的而进行操作的一系列语句和指令。一般分为系统程序和应用程序两大类。那么，如何编写程序呢？著名计算机科学家沃思（Nikiklaus Wirth）给出了答案，他提出了一个公式：程序=数据结构+算法。即一个程序主要包括两个方面的内容。

（1）对数据的描述。在程序中要指定数据的类型和数据的组织形式，即数据结构（Data Structure）。

（2）对操作的描述。即操作步骤，也就是算法（Algorithm）。

实际上，一个程序除了以上两个要素外，还应当采用程序设计方法进行设计，并且用一种计算机语言来表示。因此，算法、数据结构、程序设计方法和语言工具四个方面是程序编写人员所应具备的知识。

1.1.2 算法

算法对于程序是十分重要的，可以说是程序的灵魂。本部分将专门介绍算法的基本知识，为后面的学习建立一定的基础。

1. 算法的概念

算法是指解题步骤的准确而完整的描述，是解决问题的一系列清晰指令。算法代表着用

系统的方法描述解决问题的策略机制。也就是说，能够对一定规范的输入，在有限时间内获得所要求的输出。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。

2. 算法的特征

算法有以下 5 个特征。

（1）有穷性（Finiteness）。算法的有穷性是指算法必须能在执行有限个步骤之后终止。

（2）确切性（Definiteness）。算法的每一步骤必须有确切的定义。

（3）输入项（Input）。一个算法有 0 个或多个输入，以刻画运算对象的初始情况。所谓 0 个输入是指算法本身定出了初始条件。

（4）输出项（Output）。一个算法有一个或多个输出，以反映对输入数据加工后的结果。没有输出的算法是毫无意义的。

（5）可行性（Effectiveness）。算法中执行的任何计算步骤都是可以被分解为基本的可执行的操作步，即每个计算步都可以在有限时间内完成（也称之为有效性）。

3. 算法的描述方法

1966 年，波姆（Bohm）和贾可皮尼（Jacopini）证明了任何单入口单出口没有死循环的程序都可以由三种基本的控制结构构造出来。这三种基本结构就是顺序结构、选择结构和循环结构。它们作为表示一个良好算法的基本单元。

顺序结构是按照一定的顺序，依次执行并完成指定的功能。顺序结构的特点是程序从入口开始，自上而下按顺序执行所有操作，直到出口为止。

选择结构表示程序的处理步骤出现了分支，它需要根据某一特定的条件选择其中的一个分支执行。选择结构有单选择、双选择和多选择三种形式。单选择结构是指程序中有一个分支可以根据条件选择执行或者不执行；双选择结构是指程序中有两个分支一个条件选择，执行时只能根据条件选择一个且必须选择一个分支执行；多选择结构是指程序中有多个条件，每个条件下都有一个分支，执行时按顺序判断条件，如果某个条件满足就执行该条件下的分支执行。这三种结构不管选择哪个分支执行，最后流程都一定到达结构的出口处。

循环结构表示程序反复执行某个或某些操作，直到循环条件不成立时才可终止循环。循环结构的基本形式有两种：当型循环和直到型循环。

这三种基本结构作为表示一个良好算法的基本单元，任何一个算法的结构都是由这三种基本结构按一定顺序排列起来的。

算法的描述方法很多，常用的有自然语言、流程图、N-S 图、伪代码、计算机语言等。

（1）用自然语言表示算法。自然语言就是人们日常使用的语言，可以是汉语、英语或其他语言。下面通过实例来说明用自然语言来描述算法的方法。

【例 1-1】 已知 a 的值是 4，b 的值是 14，将 a、b 的值互换，互换后 a 的值为 14、b 的值为 4，然后输出交换后 a、b 的值。

算法分析：由于 a、b 两个变量的值不能直接交换，直接交换将使得 a、b 的值相等，所以解决该问题的方法是引入第三个变量，作为交换的中间临时变量。设第三个变量为 c，其交换步骤可以用自然语言描述如下。

步骤 1：把 4 赋给变量 a。

步骤2：把14赋给变量*b*。

步骤3：将变量*a*的值赋给变量*c*。

步骤4：将变量*b*的值赋给变量*a*。

步骤5：将变量*c*的值赋给变量*b*。

步骤6：输出变量*a*和变量*b*的值。

步骤7：算法结束。

【例1-2】 输出*a*、*b*两个不同数中的较大数。

算法分析：这是一个比较经典的选择结构实例。对输入的两个数进行判断，将其中的较大数输出。用自然语言描述算法如下。

步骤1：输入*a*和*b*的值。

步骤2：判断*a*和*b*的大小，如果*a*大于*b*，执行第3步，否则执行第4步。

步骤3：输出*a*的值。

步骤4：输出*b*的值。

步骤5：算法结束。

【例1-3】 求1+2+3+…+100。

算法分析：这是一个求和算法。如果一步一步求和，需要写99个步骤，显然是不可取的，应当找一种通用的表示方法。这里设*p*为被加数，*q*为加数，并将每一步的和放到*p*中。用自然语言描述算法如下。

步骤1：使*p*=1。

步骤2：使*q*=2。

步骤3：使*p*+*q*，和仍放在*p*中，可表示为*p*+*q*→*p*。

步骤4：使*q*的值加1，即*q*+1→*q*。

步骤5：如果*q*不大于100，返回重新执行步骤3、步骤4和步骤5。否则，算法结束。最后得到*p*的值就是1+2+3+…+100的和。

由上述例子看出，[例1-1]中的操作步骤是自上而下顺序执行的，称之为顺序结构；[例1-2]中的操作步骤是根据条件判断决定执行哪个操作，这种结构称之为选择结构；[例1-3]中不仅包含了判断，而且需要重复执行第3、第4步骤，并且一直延续到条件“*q*大于100”为止，这种结构称之为循环结构。

使用自然语言表示的算法通俗易懂，但文字冗长，容易出现歧义，特别是对于包含分支和循环的算法。因此除了很简单的问题，一般不用自然语言描述算法。

（2）用流程图表示算法。以特定的图形符号加上说明表示算法的图，称为流程图或框图。流程图有时也称为输入—输出图。该图直观地描述一个工作过程的具体步骤。流程图使用一些标准符号代表某些类型的动作。美国国家标准化协会ANSI规定了一些常用的流程图符号，已为世界各国程序工作者普遍采用。流程图符号如图1-1所示。

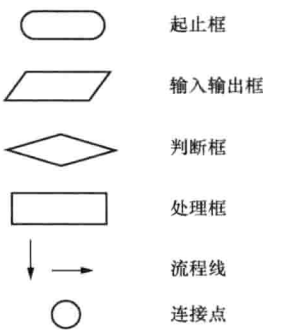

图1-1 流程图符号

【例1-4】 将[例1-1]的算法用流程图表示。流程图如图1-2所示。

【例1-5】 将[例1-2]的算法用流程图表示。流程图如图1-3所示。

【例1-6】 将[例1-3]的算法用流程图表示。流程图如图1-4所示。

用流程图表示算法直观形象，易于理解，不会产生“歧义性”。流程图便于交流，适于初学者使用。对于一个程序工作者来说，会看、会用流程图是十分必要的。

（3）用 N-S 图表示算法。N-S 图也被称为盒图或 CHAPIN 图。它由一些特定意义的图形、流程线及简要的文字说明构成。它能明确地表示程序的运行过程。在使用过程中，人们发现流程线不一定是必需的。为此，人们设计了一种新的流程图。它把整个程序写在一个大框图内，这个大框图由若干个小的基本框图构成，这种流程图简称 N-S 图。

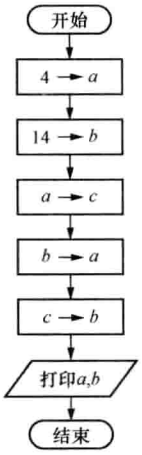

图 1-2 ［例 1-1］算法流程图

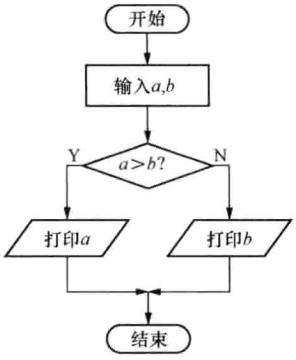

图 1-3 ［例 1-2］算法流程图

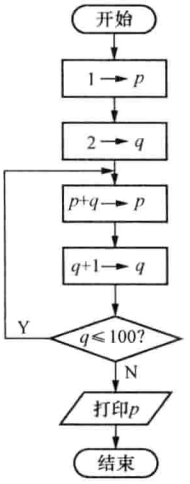

图 1-4 ［例 1-3］算法流程图

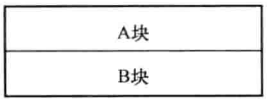

图 1-5 顺序结构

图 1-5 表示顺序结构，它由 A 块和 B 块两个框组成。图 1-6 表示选择结构，当条件为真时执行 A 块，为假时执行 B 块。图 1-7 和图 1-8 表示循环结构，其中图 1-7 表示当型循环结构，图 1-8 表示直到型循环结构。

用 N-S 图分别表示［例 1-1］～［例 1-3］的算法，如图 1-9～图 1-11 所示。

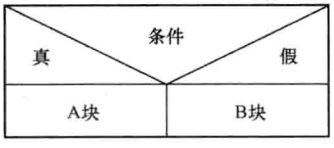

图 1-6 选择结构

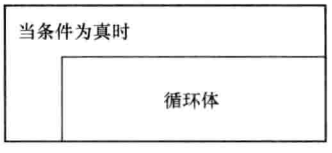

图 1-7 当型循环结构

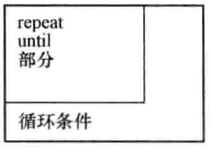

图 1-8 直到型循环结构

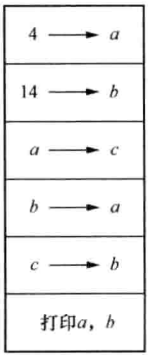

图 1-9 ［例 1-1］算法表示

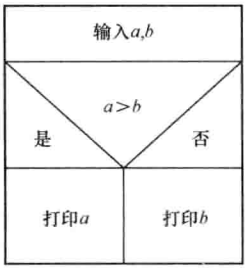

图 1-10 ［例 1-2］算法表示

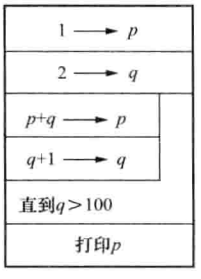

图 1-11 ［例 1-3］算法表示

（4）用伪代码表示算法。伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。用伪代码表示算法时，可以用英文伪代码，也可以用汉字伪代码，还可以中、英文混用。用伪代码写算法并无固定的、严格的语法规则，只要用清晰易读的形式把意思表达清楚即可。

【例 1-7】 输入三个数，打印输出其中最大的数。可用如下的伪代码表示算法。

Begin（算法开始）
输入 A，B，C
IF A>B 则 A→Max
否则 B→Max
IF C>Max 则 C→Max
Print Max
End（算法结束）

使用伪代码的目的是为了使被描述的算法可以容易地以任何一种编程语言实现。因此，伪代码必须结构清晰、代码简单、可读性好，并且类似自然语言，最好介于自然语言与编程语言之间。使用伪代码，不用拘泥于具体算法实现，主要将整个算法运行过程的结构用接近自然语言的形式描述出来。描述时可以使用任何一种熟悉的文字，关键是把程序的意思表达出来。

（5）用计算机语言表示算法。前面介绍的都是描述算法，实际上每一个算法最终都得转换成程序，即用计算机语言实现算法。因为计算机只能识别用计算机语言编写的程序，而无法识别流程图和伪代码。只有这样，通过让计算机执行程序，才能实现算法所要实现的功能。

和前面几种形式不同的是，用计算机语言表示算法必须严格遵守所用语言的语法规则。下面将前面介绍过的算法用C语言描述出来。

【例 1-8】 将［例 1-1］（变量 *a*，*b* 的值互置）用C语言表示。

```
#include <stdio.h>
void main()
{int a,b,c;                          /*定义 a,b,c 为整型变量*/
a=4;                                /*给 a 赋以整数 4*/
b=14;                               /*给 b 赋以整数 14*/
c=a;                                /*把 a 的值赋给 c*/
a=b;
b=c;
printf("a=%d,b=%d\n",a,b);          /*输出变量 a,b 的值*/
}
```

【例 1-9】 将［例 1-2］（输出 *a*，*b* 两个不同数中的大数）用C语言表示。

```
#include <stdio.h>
void main()
{
int a,b,c;
scanf("%d%d",&a,&b);
if(a>b)printf("%d\n",a);
else printf("%d\n",b);
}
```

【例 1-10】 将［例 1-3］（求 1+2+3+…+100）用 C 语言表示。

```
#include <stdio.h>
void main()
{int p,q;
   p=1;q=2;
   while(q<=100)
   {p=p+q;q=q+1;}
   printf("1+2+3+…+100=%d\n",p);
}
```

这里只要求读者能大体看懂程序的流程即可，有关 C 语言的使用规则将在后面的章节详细介绍。

总之，学习程序设计的关键是掌握程序设计的算法。掌握了算法就是掌握了程序设计的灵魂。掌握算法后，再学习一些有关的计算机语言知识，就能顺利地编写程序了。

1.2 计 算 机 语 言

计算机语言是指计算机能够接收和处理的具有一定格式的语言，是进行程序设计时最重要的工具之一。计算机语言经历几个发展阶段，下面介绍计算机语言的发展历史。

1. 机器语言

机器语言是第一代计算机语言，它是用二进制代码 0 和 1 表示的计算机能直接识别和执行的一种机器指令的集合。它是计算机的设计者通过计算机的硬件结构赋予计算机的操作功能。机器语言具有灵活、直接执行和速度快等特点。

用机器语言编写的程序称为机器语言程序，是计算机唯一能直接识别并执行的语言。用机器语言编写程序，编程人员要首先熟记所用计算机的全部指令代码和代码的涵义。编程序时，程序员得自己处理每条指令和每一数据的存储分配和输入/输出，还得记住编程过程中每步所使用的工作单元处在何种状态。这是一件十分烦琐的工作，编写程序花费的时间往往是实际运行时间的几十倍或几百倍。而且，编出的程序全是 0 和 1 的指令代码，直观性差，还容易出错。现在，除了计算机生产厂家的专业人员外，绝大多数的程序员已经不再去学习机器语言了。

2. 汇编语言

汇编语言是第二代计算机语言，它是一种借用助记符表示的程序设计语言，它每条指令都对应着一条机器语言代码。在汇编语言中，用助记符代替机器指令的操作码，用地址符号或标号代替指令或操作数的地址，如此就增强了程序的可读性。使用汇编语言编写的程序，机器不能直接识别，必须由“汇编程序”翻译成机器语言程序，才能够在计算机上运行。这种“汇编程序”称为汇编语言的翻译程序。汇编语言适用于编写直接控制机器操作的底层程序。

汇编语言与机器联系仍然比较紧密，不容易使用，二者都属于低级语言。低级语言依赖于所在的计算机系统，也称为面向机器的语言。由于不同的计算机系统使用的指令系统可能不同，因此使用低级语言编写的程序移植性较差。

3. 面向过程的结构化语言

面向过程的结构化语言是第三代语言，从第三代语言开始都是高级语言。高级语言编写

的程序易读、易修改、移植性好。但使用高级语言编写的程序不能直接在机器上运行，必须经过语言处理程序的转换，才能被计算机识别。按照转换方式的不同，可将高级语言分为解释型和编译型两大类。

所谓解释型转换，是将编写的程序逐句翻译，翻译一句执行一句，即边翻译边执行，其中转换工作是由解释器自动完成的。常见的解释型语言包括BASIC语言和Perl语言。解释型转换方式的优点是比较灵活，可以动态地调整和修改程序；缺点是效率比较低，不能生成独立的可执行文件，即程序的运行不能脱离其解释器。

编译型语言编写的程序经过翻译等处理后，可以脱离其语言环境而生成一个可以独立执行的文件。例如C语言、Pascal语言等大多数编程语言都属于编译型语言。

面向过程的结构化语言具有以下特点。

（1）采用模块分解与功能抽象的方法，自顶向下，逐步求精。

（2）按功能划分为若干个基本的功能模块，形成一个树状结构。各模块间的关系尽可能简单，功能上相对独立。每一个功能模块内部都是由顺序、选择或循环三种基本结构组成。

面向过程的结构化语言能有效地将一个比较复杂的任务分解成若干个易于控制和处理的子任务。任务的分解有利于程序的设计与维护。C语言即属于面向过程的结构化语言。

4. 面向对象的语言

面向对象的语言是第四代语言。由于面向过程的程序是按照流水线方式执行的，即一个模块执行结束前，不能执行其他模块，也无法动态地改变程序的执行方向。而在实际处理事务时，总期望每发生一件事情就可以进行处理，即程序应该从面向过程改为面向具体的应用功能（即对象）。

20世纪80年代初期，面向对象程序设计语言开始出现。这种语言的目标是实现软件的集成化，把相互联系的数据及对数据的操作封装成通用的功能模块，各功能模块可以相互组合，完成具体的应用。各功能模块还可以重复使用，而用户不必关心其功能是如何实现的。C++、Java等是典型的面向对象的语言。

5. 非过程性语言

非过程性语言是一种新型的语言。它只需程序员具体说明问题的规则并定义一些条件即可。意思就是你只要说做什么，具体怎么做不需描述。语言自身内置了方法把这些规则解释为一些解决问题的步骤，这就把编程的重心转移到描述问题和其规则上。

因此，非过程性语言更适合于思想概念清晰但数学概念复杂的编程工作。例如，数据库查询SQL语言和逻辑式语言Prolog就是非过程性语言的代表。SQL只需程序员和用户对数据库中数据元素之间的关系和欲读取信息的类型予以描述，逻辑式语言的语义基础是基于一组已知规则的形式逻辑系统，被广泛应用于各种专家系统的实现。

6. 管理解析语言

管理解析语言基于高层次的业务需求，涵盖企业管理软件开发的特定概念和抽象，由低层次的实现细节和具体事物抽象而来，据有字典、单据、报表、工作流、审批流等管理业务描述的快速实现，以最小的、不可拆分的业务规则作为管理解析语言的基本粒度，按照管理逻辑进行组合，形成特定管理业务的标准实现。

YiGo语言是第一个实现管理解析思想的计算机语言，拥有软件开发的原子逻辑及很多管理业务的分子操作及其界面元素，实现了对硬件、操作系统、数据库的透明操作。

管理解析语言是一种高科技语言，现在引用的领域不多。

1.3 C 语 言 简 介

1.3.1 C 语言的产生和发展

C 语言是 1972 年由美国的丹尼斯·里奇（Dennis Ritchie）设计发明的，并首次在 UNIX 操作系统的 DEC PDP-11 计算机上使用。它由早期的编程语言 BCPL（Basic Combined Programming Language）发展演变而来。在 1970 年，AT&T 贝尔实验室的肯·汤普森（Ken Thompson）根据 BCPL 语言设计出较先进的并取名为 B 的语言，最后导致了 C 语言的问世。而 B 语言之前还有 A 语言，取名自欧洲国家女性的常用名艾达（Ada）。

C 语言是一种计算机程序设计语言，它既具有高级语言的特点，又具有汇编语言的特点。1978 年后，C 语言已先后被移植到大、中、小及微型机上。它可以作为工作系统设计语言，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。它的应用范围广泛，具备很强的数据处理能力，不仅在软件开发上，而且各类科研都需要用到 C 语言。

随着 C 语言的发展，它出现了许多版本。由于没有统一的标准，这些 C 语言之间出现了一些不一致的地方。为了改变这一状况，美国国家标准协会（ANSI）根据 C 语言问世以来的各种版本，对 C 语言进行了改进和扩充，制定了 ANSI C 标准，成为现行的 C 语言标准。目前，在计算机上广泛使用的 C 语言编译系统有 Borland C++、Turbo C、Microsoft Visual C++（简称 VC++）等。其中比较常用的是 Turbo C 和 VC++。

1.3.2 C 语言的特点

C 语言作为第三代面向过程的结构化语言的代表，具有许多优点，主要有以下几点。

（1）C 语言简洁、紧凑。C 语言简洁、紧凑，而且程序书写形式自由，使用方便、灵活。C 语言一共有 32 个关键字，9 种控制语句，程序书写自由。

（2）C 语言是高、低级兼容语言。C 语言又称为中级语言，它介于高级语言和低级语言（汇编语言）之间，既具有高级语言面向用户、可读性强、容易编程和维护等优点，又具有汇编语言面向硬件和系统并可以直接访问硬件的功能。

（3）C 语言是一种结构化的程序设计语言。结构化语言的显著特点是程序与数据独立，从而使程序更通用。这种结构化方式可使程序层次清晰，便于调试、维护和使用。

（4）C 语言是一种模块化的程序设计语言。所谓模块化，是指将一个大的程序按功能分割成一些模块，使每一个模块都成为功能单一、结构清晰、容易理解的函数，适合大型软件的研制和调试。

（5）C 语言可移植性好。C 语言是面向硬件和操作系统的，但它本身并不依赖于机器硬件系统，从而便于在硬件结构不同的机器间和各种操作系统间实现程序的移植。

（6）C 语言运算功能丰富。C 语言不仅提供了 34 种运算符，还提供了强大的库函数，从而使 C 语言的运算类型极为丰富。

（7）C 语言数据结构丰富。C 语言具有现代化语言的各种数据结构，C 语言的数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型等，能用来实现各种复杂的数据结构运算。