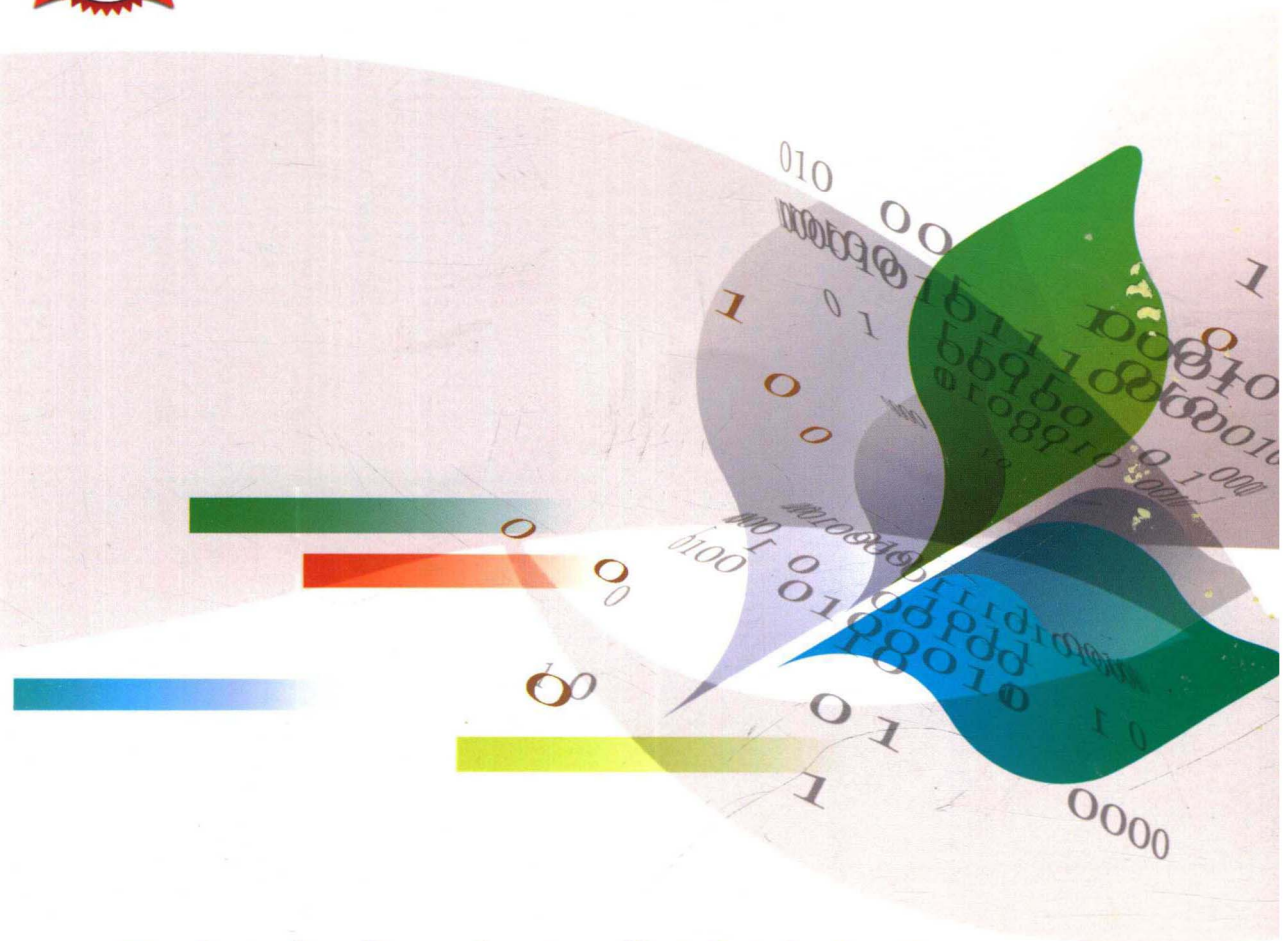
普通高等教育“十三五”规划教材



C语言程序设计（第2版）

C Programming Language, Second Edition

◎主编　耿　姝　　◎副主编　逯　柳　张　旭　孙　毅

中国工信出版集团

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

普通高等教育“十三五”规划教材

C 语言程序设计

（第 2 版）

耿 姝 主 编

逯 柳 张 旭 孙 毅 副主编

電子工業出版社
Publishing House of Electronics Industry
北京 · BEIJING

内 容 简 介

本书从程序设计的实际能力培养出发，由浅入深、深入浅出，将理论与实践有机结合，集知识传播和能力培养为一体。本书内容丰富、注重实践；突出重点、分散难点；例题广泛、结合实际。本书的宗旨在于进一步巩固对基本知识的理解和掌握，提高学生的逻辑分析、抽象思维和程序设计能力，培养学生良好的程序设计风格，进而具备编写中、大型程序的能力。

本书中的程序在按照模块化程序设计思想进行编写的同时，每一个程序都遵循软件工程方法学的编程风格，即采用缩进格式，程序中附有注释，以便于对程序的分析、理解和自学。

本书不仅可以作为高等院校学生 C 语言程序设计教材，而且还可以作为计算机等级考试的参考书和编程爱好者自学 C 语言的自学教材。

图书在版编目（CIP）数据

C 语言程序设计 / 耿姝主编. —2 版. —北京：电子工业出版社, 2016.2
普通高等教育“十三五”规划教材
ISBN 978-7-121-27845-7

I. ①C… II. ①耿… III. ①C 语言－程序设计－高等学校－教材 IV. ①TP312

中国版本图书馆 CIP 数据核字（2015）第 300421 号

策划编辑：袁 玺
责任编辑：郝黎明 特约编辑：张燕虹
印 刷：三河市兴达印务有限公司
装 订：三河市兴达印务有限公司
出版发行：电子工业出版社
北京市海淀区万寿路 173 信箱 邮编：100036
开 本：787×1092 1/16 印张：17.5 字数：504 千字
版 次：2012 年 8 月第 1 版
2016 年 2 月第 2 版
印 次：2016 年 2 月第 1 次印刷
定 价：38.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：（010）88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：（010）88258888。

前　言

计算机应用能力是21世纪人才不可缺少的基本素质。程序设计是各专业计算机应用能力培养的重要技术基础，C语言是目前国内外广泛使用的一种程序设计语言，是国内外大学讲述程序设计方法的首选语言。

全国计算机等级考试、全国计算机应用技术证书考试和全国各地区组织的大学生计算机等级考试都已将C语言列入了考试范围。学习C语言已成为广大计算机应用人员和青年学生的迫切愿望和要求。

教材是知识传播和能力培养的基础，本书从程序设计的实际能力培养出发，由浅入深、深入浅出，将理论与实践有机结合，集知识传播和能力培养为一体。本书内容丰富、注重实践；突出重点、分散难点；例题广泛、结合实际。

本书共10章，主要介绍了C语言程序设计基础知识、数据的存储与运算、三种结构化程序设计方法、数组、函数、指针、用户自定义数据类型和文件系统等。每章均配有典型习题，突出了实用性，强调理论与实践相结合，培养了学生的编程能力。

本书不仅可以作为高等院校学生C语言程序设计教材，而且还可以作为计算机等级考试的参考书和编程爱好者自学C语言的自学教材。

本书有大量的算法语句、程序语句及计算公式，对于其中的变量，为了方便读者阅读，避免歧义，不再区分正、斜体，而是统一采用正体，特此说明。

本书由耿姝主编，各章节的编写分工如下：第1章、第2章、第3章由逯柳编写；第4章、第5章、第6章由张旭编写；第7章、第8章及附录由耿姝编写；第9章、第10章由孙毅编写。

由于作者水平有限，书中难免存在疏漏与不妥之处，恳请同行与广大读者批评指正。

编　者

目　录

第 1 章　C 语言程序设计概述

1.1　程序与程序设计语言

1.1.1　程序

计算机是可以按照人们事先编写的程序高速、精确地进行数据加工、处理的电子装置。如果我们需要计算机完成什么工作，将要完成工作的步骤用诸条指令描述出来，并把这些指令存放在计算机的存储器中，需要结果时就向计算机发出一条简单的命令，计算机就会自动逐条顺序地执行指令，全部指令执行完就得到了预期的结果。这种可以被连续执行的一条条指令的集合称为计算机的程序。也就是说，程序是计算机的指令序列，是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述。

1.1.2　计算机语言

人们用自然语言讲述和书写，目的是给别人传播信息。同样，我们使用计算机语言把我们的意图表达给计算机，目的是使用计算机，让计算机给我们完成一定信息的处理。

为了使计算机进行各种工作，就需要有一套用以编写计算机程序的数字、字符和语法规则，由这些数字、字符和语法规则组成计算机的各种指令（或各种语句），就是计算机能接受的语言。

计算机每做一次动作，完成一个步骤，都是按照已经用计算机语言编好的程序来执行的。计算机语言没有自然语言那么丰富多样，而只是有限规则的集合，所以它简单易学。但是，也正因为它是根据机器的特点编制的，所以交流中无法意会和言传，更多地表现了说一不二，表现了规则的严谨。

1．程序语言的发展

计算机语言不断从低级向高级发展，其发展过程可分为三代：机器语言、汇编语言和高级语言。

机器语言是用二进制代码表示的计算机能直接识别和执行的机器指令的集合。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，它与计算机同时诞生，是第一代的计算机语言。

例如，计算 A=12+10 的机器语言程序如下：

```
10110000 00001100        ：  把 12 放入累加器 A 中
00101100 00001010        ：  10 与累加器 A 的值相加，结果仍放入 A 中
11110100                 ：  结束，停机
```

使用机器语言的缺点是编程工作量大，难学、难记、难修改，它只适合专业人员使用；而且由于不同的计算机，其指令系统不同，机器语言随机而异，通用性差，是面向机器的语言。

机器语言的优点是程序代码不需要翻译，所占空间少，执行速度快。

汇编语言将机器指令的二进制代码用英文助记符来表示，代替机器语言中的指令和数据。例如用 ADD 表示加、SUB 表示减、JMP 表示程序跳转等，这种指令助记符的集合就是汇编语言。

例如，计算 A=12+10 的汇编语言程序：

```
MOV A, 12          ：把12放入累加器A中
ADD A, 10          ：10与累加器A相加，结果存入A中
HLT                ：结束，停机
```

汇编语言的优点是克服了机器语言难读等缺点，保持了其编程质量高，占存储空间少、执行速度快的优点。

汇编语言的缺点是依赖于机器，通用性差。

汇编语言源程序必须通过汇编程序翻译成机器语言，计算机才能执行。汇编语言常用于过程控制等编程。

高级语言是一种接近于自然语言和数学公式的程序设计语言。它采用了完全符号化的描述形式，用类似自然语言的形式描述对问题的处理过程，用数学表达式的形式描述对数据的计算过程。高级语言主要是相对于汇编语言而言，它并不是特指某一种具体的语言，而是包括了很多编程语言。

例如，计算A=12+10的BASIC语言程序如下：

```
A=12+10            ：12与10相加的结果放入A中
PRINT A            ：输出A
END                ：程序结束
```

高级语言的优点是通用性强，编程效率高。它使程序员可以不用与计算机的硬件打交道，可以不必了解机器的指令系统，集中精力解决问题本身而不受机器制约，极大地提高了编程的效率。

高级语言源程序要通过翻译程序翻译成机器语言才能运行，程序的效率不如优化的汇编程序高。

机器语言和汇编语言都是面向机器的语言，属于低级语言；而高级语言又分为面向过程和面向对象两种。前一种程序设计是数据被加工的过程，后一种程序设计的关键是定义类，并由类派生对象。客观世界可以分类，对象是类的实例，对象是数据和方法的封装，对象间通过发送和接收消息发生联系。

面向过程的高级语言只是要求人们向计算机描述问题的求解过程，而不关心计算机的内部结构，它易于被人们理解和接受。典型的面向过程语言有BASIC、FORTRAN、COBOL、C、Pascal等。程序设计语言的发展如图1-1所示。

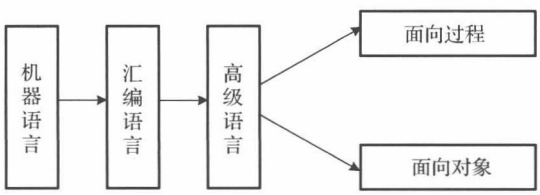


图1-1 程序设计语言的发展

面向对象的高级语言是“面向过程”的一次革命，如果说面向过程的语言要求人们告诉计算机怎么做，那么面向对象的语言只要求人们告诉计算机做什么。面向对象是通过类和对象把程序所涉及的数据结构和对它施行的操作有机地组织成模块，对数据和数据的处理细节进行最大限度的封装，从而使开发出来的软件易重用、易修改、易调试、易扩充。

2. 语言处理程序

在所有的程序设计语言中，除了用机器语言编制的程序能够被计算机直接理解和执行外，其他的程序设计语言编写的源程序都必须经过一个翻译过程才能转换为计算机所能识别的机器语言程序。实现这个翻译过程的工具是语言处理程序，即翻译程序。翻译程序也称为编译器。用非机

器语言写的程序称为源程序，通过翻译程序翻译后的程序称为目标程序。针对不同的程序设计语言编写出的程序，有各自的翻译程序，互相不通用。

翻译程序翻译源程序通常有两种方式：解释方式和编译方式。

（1）解释方式：解释方式的翻译工作由解释程序来完成，这种方式如同“口译”。解释程序对源程序进行逐句分析，若没有错误，将该语句翻译成一个或多个机器语言指令；然后立即执行这些指令；若解释时发现错误，会立即停止，报错并提醒用户更正代码，具体过程如图 1-2 所示，解释方式不生成目标程序。

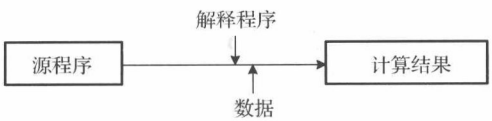


图 1-2　解释方式的执行过程

采用解释方式的优点是查找错误的语句行和修改方便。缺点是执行速度慢，采用解释方式运行的源程序，每次运行都必须重新解释，若程序较大，错误发生在程序的后面，则前面运行的结果是无效的，解释程序无法对整个程序进行优化。

（2）编译方式：翻译工作由编译程序完成。如同“笔译”，在纸上记录翻译后的结果。编译程序过程：对源程序编译产生目标程序，连接程序将目标程序和有关的程序库组合成可执行程序，编译方式的执行过程如图 1-3 所示。

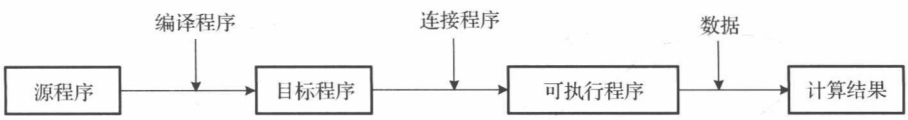


图 1-3　编译方式的执行过程

采用编译方式的优点是程序执行速度快，产生的可执行程序可以脱离编译程序和源程序独立存在并反复运行，但修改源程序后都必须重新编译生成目标程序。

一般高级语言（C/C++、Pascal、FORTRAN、COBOL 等）都是采用编译方式。

1.2　程序设计（解决什么问题、如何解决、实现方法）

计算机之所以能够产生如此大的影响，其原因不仅在于人们发明了机器本身，更重要的是人们为计算机开发出了不计其数的能够指挥计算机完成各种各样工作的程序。正是这些功能丰富的程序给了计算机无尽的生命力，是程序设计工作智慧的结晶。

所谓程序，是用计算机语言对所要解决的问题中的数据以及处理问题的方法和步骤所做的完整而准确的描述，而程序设计就是用某种程序语言编写这些解决问题的步骤的过程。对数据的描述就是指明所要处理问题中的数据结构（即数据和数据之间的关系）；对问题处理方法和步骤的描述就是算法。因此，数据结构与算法是程序设计过程中密切相关的两个方面。著名计算机科学家 Niklaus Wirth 教授关于程序提出了著名公式：程序=数据结构+算法。这个公式说明了程序设计的主要任务。

如何进行程序设计呢？一个简单的程序设计一般包含以下步骤。

1．分析问题

使用计算机解决具体问题时，首先要对问题进行充分的分析，确定问题是什么，解决问题

的步骤又是什么。针对所要解决的问题，找出已知的数据和条件，确定所需的输入、处理及输出对象。

2. 确定数据结构和算法

在分析求解问题的基础上，将所研究问题的数据和数据间关系抽象出来，确定程序中数据的类型和数据组织存储形式，即确定存放数据的数据结构。针对问题的分析和确定的数据结构，选择合适的算法加以实现。注意，这里所说的“算法”泛指解决某一问题的方法和步骤。

3. 编制程序

根据确定的数据结构和算法，用一种程序设计语言把这个解决方案严格地描述出来，也就是编写出程序代码。

4. 调试程序

在计算机上用实际的输入数据对编好的程序进行调试，分析所得到的运行结果，进行程序的测试和调整，直至获得预期的结果。

5. 分析结果

对程序执行结果进行验证和分析，发现程序中存在的问题并修改完善。

6. 写出程序的文档

程序是提供给用户使用的，如同正式的产品应当提供产品说明书一样，正式提供给用户使用的程序，必须向用户提供程序说明书。内容应包括程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据，以及使用注意事项等，为程序的使用、修改做好基础工作。

1.3 C语言的发展

C语言是世界上广泛流行的计算机高级程序设计语言，它于1973年由美国贝尔实验室设计发布。由于C语言同时具备高级语言的优点和低级语言的效率，而且拥有很好的可移植性，使它成为程序员最喜欢的语言。

C语言起源于对系统程序设计的深入研究和发展。C语言是贝尔实验室的Ken Thompson和Dennis Ritchie等人开发的UNIX操作系统的“副产品”。Thompson独自编写了UNIX操作系统的最初版本，与同时代的其他操作系统一样，UNIX最初也是用汇编语言编写的，用汇编语言编写的程序往往难以调试和改进，UNIX也不例外。Thompson意识到需要用一种更加高级的编程语言来完成UNIX系统未来的开发，于是他设计了一种小型的B语言。B语言是在BCPL（Basic Combined Programming Language）的基础上开发的，而BCPL是在CPL（Combined Programming）语言的基础上开发的，CPL可以追溯到最早的语言之一Algol60语言。

1970年，K. Thompson用B语言在PDP-7机上实现了第一个实验性的UNIX操作系统。1972年，贝尔实验室的Dennis M. Ritchie为克服B语言的诸多不足，在B语言的基础上重新设计了一种语言，由于是B语言的后继，故称为C语言。1973年，贝尔实验室的K. Thompson和Dennis M. Ritchie合作，首先用C语言重新改写了UNIX操作系统，在当时的PDP-11计算机上运行。此后，C语言作为UNIX操作系统上标准的系统开发语言，伴随着UNIX操作系统的发展，C语言越来越广泛地被人们接受和应用。

1977 年，出现了独立于具体机器的 C 语言编译版本。由于 C 语言的独立和推广，也推动了 UNIX 操作系统在各种机器上的迅速实现。

1978 年，美国电话电报公司（AT&T）贝尔实验室正式发表了 C 语言。同时，由 B.W.Kernighan 和 D.M.Ritchie 合著了著名的《The C Programming Language》一书，简称为《K&R》，也有人称之为《K&R》标准。此书中介绍的 C 语言成为后来广泛使用的 C 语言版本基础，它被称为标准 C 语言。

1983 年，美国国家标准化协会（ANSI）根据各种 C 语言版本对 C 的扩充和发展，颁布了 C 语言的新标准 ANSI C。ANSI C 比标准 C 有了很大的扩充和发展。

1987 年，美国国家标准化协会在综合各种 C 语言版本的基础上，又颁布新标准，为了与标准 ANSI C 区别，所以称为 87 ANSI C。1990 年，国际标准化组织 ISO 接受了 87 ANSI C 作为 ISO C 的标准。这是目前功能最完善、性能最优良的 C 新版本。目前流行的 C 编译系统都是以它为基础的。

1989 年，X3J11 提出了一个报告[ANSI 89]，后来这个标准被 ISO 接受为 ISO/IEC 9899—1990。

1990 年，国际标准化组织ISO（International Organization for Standards）接受 89 ANSI C 作为 ISO C 的标准（ISO 9899—1990）。1994 年，ISO 修订了 C 语言的标准。

1995 年，ISO 对 C90 做了一些修订，即 1995 基准增补 1（ISO/IEC/9899/AMD1:1995）。1999 年，ISO 再次对 C 语言标准进行了修订，在保留原来 C 语言特征的基础上，针对实际编写应用程序的需要，增加了一些功能，尤其是完善了 C++语言中的一些功能，并将其命名为 ISO/IEC9899:1999。

2001 年和 2004 年先后进行了两次技术修正。目前流行的 C 语言编译系统大多是以 ANSI C 为基础进行开发的，但不同版本的 C 编译系统所实现的语言功能和语法规则又略有差别。

2011 年 12 月，ISO 正式公布 C 语言新的国际标准草案：ISO/IEC 9899:2011。

新的标准修订了 C11 版本，提高了对 C++的兼容性，并将新的特性增加到 C 语言中。新功能包括支持多线程，基于 ISO/IEC TR 19769:2004 规范下支持 Unicode，提供更多用于查询浮点数类型特性的宏定义和静态声明功能。

1.4　C 语言的特点

C 语言发展如此迅速，而且成为最受欢迎的语言之一，主要是因为它具有强大的功能。许多著名的系统软件，如 DBASE Ⅲ PLUS、DBASE Ⅳ 都是由 C 语言编写的。用 C 语言加上一些汇编语言子程序，就更能显示 C 语言的优势了，像 PC-DOS 、WORDSTAR 等就是用这种方法编写的。从语言体系和结构上讲，它是结构化程序设计语言；但从用户应用、实现难易程度、程序设计风格等角度来看，C 语言的特点又是多方面的。

1．C 是中级语言

C 语言通常称为中级计算机语言，它把高级语言的基本结构和语句与低级语言的实用性结合起来。中级语言并没有贬义，不意味着它功能差、难以使用，或者比 BASIC、Pascal 那样的高级语言原始，也不意味着它与汇编语言相似，会给使用者带来类似的麻烦。作为中级语言，C 允许对位、字节和地址这些计算机功能中的基本成分进行操作。C 语言可以实现汇编语言的大部分功能，可以直接操作计算机硬件如寄存器，各种外设 I/O 端口等。C 语言的指针可以直接访问内存物理地址，类似汇编语言的位操作可以方便地检查系统硬件的状态等。

所有的高级语言都支持数据类型的概念。一个数据类型定义了一个变量的取值范围和在其上

操作的一组运算。常见的数据类型是整型、字符型和实数型。虽然C语言有五种基本数据类型，但与Pascal或Ada相比，它却不是强类型语言。C程序允许几乎所有的类型转换。例如，字符型和整型数据能够自由地混合在大多数表达式中进行运算。这在强类型高级语言中是不允许的。

C语言的另一个重要特点是，它仅有32个关键字，这些关键字就是构成C语言的命令。和IBM PC的BASIC相比，后者包含的关键字达159个之多。

2. C是结构式语言

虽然从严格的学术观点上看，C语言是块结构（block-structured）语言，但是它还是常被称为结构化语言。这是因为它在结构上类似于ALGOL、Pascal和Modula-2（从技术上讲，块结构语言允许在过程和函数中定义过程或函数。用这种方法，全局和局部的概念可以通过“作用域”规则加以扩展，“作用域”管理变量和过程的“可见性”。因为C语言不允许在函数中定义函数，所以不能称之为通常意义上的块结构语言）。

结构化语言的显著特征是代码和数据的分离。这种语言能够把执行某个特殊任务的指令和数据从程序的其余部分分离出去、隐藏起来。获得隔离的一个方法是调用使用局部（临时）变量的子程序。通过使用局部变量，我们能够写出对程序其他部分没有副作用的子程序。这使得编写共享代码段的程序变得十分简单。如果开发了一些分离很好的函数，在引用时，我们仅需要知道函数做什么，不必知道它如何做。切记：过度使用全局变量（可以被全部程序访问的变量）会由于意外的副作用而在程序中引入错误。

结构化语言比非结构化语言更易于程序设计，用结构化语言编写的程序的清晰性使得它们更易于维护。这已是人们普遍接受的观点了。C语言的主要结构成分是函数C的独立子程序。

在C语言中，函数是一种构件（程序块），是完成程序功能的基本构件。函数允许一个程序的诸多任务被分别定义和编码，使程序模块化。可以确信，一个好的函数不仅能正确工作且不会对程序的其他部分产生副作用。

3. C语言功能齐全

C语言具有各种各样的数据类型，包括整型、实型、字符型、数组类型、指针类型、结构体类型、联合体类型等，因此，便于实现各种复杂的数据结构，如线性表、栈、队列、树和图等的运算。

并引入了指针概念，可使程序效率更高。另外，C语言也具有强大的图形功能，支持多种显示器和驱动器。而且，计算功能、逻辑判断功能也比较强大，可以实现决策目的。

指针是C语言的一大特色，可以说是C语言优于其他高级语言的一个重要原因。就是因为它有指针，所以可以直接进行靠近硬件的操作，但是对C语言的指针操作不做保护，也给它带来了很多不安全的因素。C++在这方面做了改进，在保留了指针操作的同时又增强了安全性，受到了一些用户的支持。但是，由于这些改进增加语言的复杂度，也为另一部分人诟病。Java则吸取了C++的教训，取消了指针操作，也取消了C++改进中一些备受争议的地方，在安全性和适合性方面均取得良好的效果，但其本身解释在虚拟机中运行，运行效率低于C++/C。一般而言，C、C++、Java被视为同一系的语言，它们长期占据着程序使用榜的前三名。

4. C语言适用范围大

C语言还有一个突出的优点是适合于多种操作系统，如DOS、UNIX，也适用于多种机型。C语言拥有一个庞大的数据类型和运算符集合，这个集合使得C语言具有强大的表达能力，可以用来编写各种系统软件和应用软件。

5．程序生成的目标代码质量高、程序运行效率高

试验表明，C 语言源程序生成的运行程序的效率仅比汇编程序的效率低 10%～20%，但 C 语言的编程速度快，程序可读性好，易于调试、修改和移植，这些优点是汇编语言所无法比拟的。

6．可移植性好

C 语言程序非常容易移植。可移植性表示为某种计算机写的软件可以用到另一种机器上去。举例来说，如果为苹果机写的一个程序能够方便地改为可以在 IBM PC 上运行的程序，则称为是可移植的。C 语言在某种计算机系统上编制的程序，只需少量改动，甚至不做修改就能移植到各种不同型号的计算机上。

7．C 语言存在的不足之处

编程自由度大，编译程序查错纠错能力有限，给不熟练的程序员带来一定困难；C 语言的理论研究及标准化工作也有待推进和完善。

高级语言接近人类语言和人们习惯使用的数学用语，不依赖于具体的机器，有严格的语法规则。相对于机器语言和汇编语言，高级语言易学易用，所编写的程序易读易改，通用性更强。C 语言把高级语言的基本结构与低级语言的高效实用性很好地结合起来，不失为一个出色而有效的现代通用程序设计语言。C 语言在计算机程序语言研究方面具有一定价值，由它引出了许多后继语言。C 语言已经成为世界上广泛流行的计算机高级程序设计语言，由于 C 语言同时具备高级语言的优点和低级语言的效率，而且拥有很好的可移植性，因此它成为程序员最喜欢的编程语言之一。C 语言以功能强大、数据结构丰富、目标代码质量高、程序运行效率高、可移植性好等特点成为众多程序员学习程序设计的首选语言，对整个计算机工业和应用的发展都起了很重要的推动作用。

1.5　C 程序的基本组成

1.简单 C 程序介绍

用 C 语言的语句编写的程序称为 C 语言程序（简称 C 程序）或 C 语言源程序。本节通过几个简单的 C 程序实例，介绍 C 程序的基本组成和结构，使读者对 C 语言和 C 程序的特性有初步的了解。

【例 1.1】在屏幕上输出：Hello,World!。

C 语言源程序如下：

```
//example1.1 The first C Program
#include <stdio.h>      //头文件
void main()
{
     printf("Hello,World!\n");
}
```

```
Hello,World!
```

图 1-4　例 1.1 运行结果

运行结果如图 1-4 所示。

分析：

①void main 表示“主函数”。每个 C 程序都必须有且只能有一个 main 函数，它是每一个 C 程序执行的起始点（入口点）。void 表示该函数没有返回值。

②用{}括起来的是主函数 main 的函数体。main 函数中的所有操作（或：语句）都在这一对{}之间，即 main 函数的所有操作都在 main 函数体中。

③本程序的“主函数”main中只有一条函数调用语句，printf()是C语言的库函数，用于程序中数据的输出（显示在屏幕上），本例是将一个字符串“Hello,World!\n”的内容输出，即在屏幕上显示：Hello,World!。

④每条语句用“;”号结束语句。

⑤头文件是每一个C程序必不可少的组成部分，因为一个C程序至少要包含输入或输出函数，而这些函数将被存放在以“.h”为扩展名的头文件中，比如本例的 printf 函数就包含在头文件“stdio.h”中。

⑥/*example1.1 The first C Program*/为注释，注释只是为了改善程序的可读性（提示、解释作用），在编译、运行时不起作用（编译时会跳过注释，目标代码中不会包含注释）。注释可以放在程序任何位置，并允许占用多行，只是需要注意“/*”、“*/”匹配，不要嵌套注释。

注释与软件的文档同等重要，要养成书写注释的良好习惯，这对软件的维护相当重要。因为程序是要给别人看的，自己也许还会看自己几年前编制的程序，清晰的注释有助于读者理解程序、算法的思路。

在软件开发过程中，还可以用注释帮助调试程序，即暂时屏蔽一些不需要运行的语句，以后可以方便地恢复。

【例 1.2】计算两数之和，并输出结果。

```
main()                          /*计算两数之和*/
{
  int a,b,sum;                  /*定义变量*/
  a=11;
  b=22;                         /*以下3行为C语句*/
  sum=a+b;
  printf("sum=%d\n",sum);
}
```

```
sum=33
```

图 1-5 例 1.2 运行结果

运行结果如图 1-5 所示。

分析：

①同样，此程序也必须包含一个 main 函数作为程序执行的起点。{}之间为 main 函数的函数体，main 函数所有操作均在 main 函数体中。

②int a,b,sum;是变量声明。声明了三个具有整数类型的变量 a,b,sum。C 语言的变量必须先声明再使用。

③a=11;b=22;是两条赋值语句。将整数 11 赋给整型变量 a，将整数 22 赋给整型变量 b。a,b 两个变量的值分别为 11, 22。注意这是两条赋值语句，每条语句均用“;”结束。

也可以将两条语句写成两行：

```
a=11;
b=22;
```

由此可见，C 程序的书写可以相当随意，但是为了保证容易阅读要遵循一定的规范。

④sum=a+b;是将 a,b 两变量内容相加，然后将结果赋值给整型变量 sum。此时，sum 的内容为 33。

⑤printf("sum=%d\n", sum);是调用库函数输出 sum 的结果。%d 为格式控制，表示 sum 的值以十进制整数形式输出。程序运行后，输出（显示）：sum=33。

【例 1.3】求两个数的较大值。

```
int max(int a,int b);                    /*函数说明*/
main()                                   /*主函数*/
{
  int x,y,z;                             /*变量说明*/
int max(int a,int b);                    /*函数说明*/
  printf("input two numbers:\n");
scanf("%d%d",&x,&y);                     /*输入 x,y 值*/
  z=max(x,y);                            /*调用 max 函数*/
  printf("maxmum=%d",z);                 /*输出*/
 }
int max(int a,int b)                     /*定义 max 函数*/
{
  if(a>b)return a;else return b;         /*把结果返回主调函数*/
 }
```

运行结果如图 1-6 所示。

```
input two numbers:
4 5
maxmum=5
```

图 1-6　例 1.3 运行结果

分析：程序由两个函数组成，即主函数和 max 函数。函数之间是并列关系，可从主函数中调用其他函数。max 函数的功能是比较两个数，然后把较大的数返回给主函数。max 函数是一个用户自定义函数。因此，在主函数中要给出说明（程序第三行）。可见，在程序的说明部分中，不仅可以有变量说明，还可以有函数说明。关于函数的详细内容将在以后章节中详细介绍。在程序的每行后用/*和*/括起来的内容为注释部分，程序不执行注释部分。

程序的执行过程是，首先在屏幕上显示提示串，请用户输入两个数，回车后由 scanf 函数语句接收这两个数送入变量 x,y 中，然后调用 max 函数，并把 x,y 的值传送给 max 函数的参数 a,b。在 max 函数中比较 a,b 的大小，把大者返回给主函数的变量 z，最后在屏幕上输出 z 的值。

2. C 程序的基本组成

综合上述三个例子，我们对 C 程序的基本组成和程序结构有了一个初步了解。

①C 程序由函数构成（函数是 C 程序的基本单位），所有的 C 程序都由一个或多个函数构成。其中，main 函数必须有且只能有一个。

②被调用的函数可以是系统提供的库函数，也可以是用户根据需要自己设计编写的函数。程序的全部工作由各个函数完成。编写 C 程序就是编写一个个函数。

③main 函数（主函数）是每个程序执行的起始点。无论 main 函数在程序中的哪个位置，一个 C 程序总是从 main 函数开始执行，也是从主函数结束。

④一个函数由函数首部和函数体两部分组成：

```
函数类型 函数名(参数类型及参数名表)
```

函数体：函数首部下方用一对{}括起来的部分。函数体一般包括声明和执行两部分。

例如函数 max:

```
函数类型  函数名称  参数类型   参数名  参数类型  参数名
  ↓        ↓        ↓        ↓       ↓       ↓
int        max      (int      x,      int      y)   /*函数首部*/
{   /*函数开始*/
    int z;  /*声明部分，定义变量*/
    if(x>y)
```