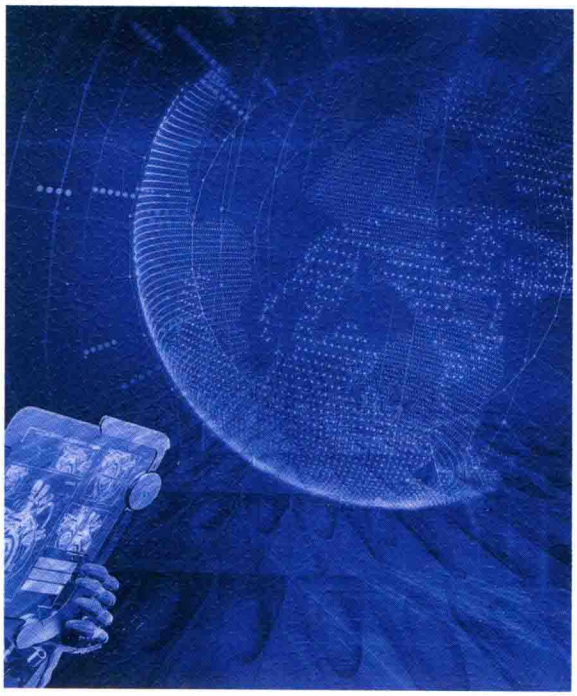
高等学校计算机应用规划教材

高级语言程序设计

(C语言)

- C语言概述
- C语言程序设计基础
- 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
- 函数
- 指针
- 结构体、共用体与自定义类型
- 编译预处理
- 文件



阳小兰　主　编

吴亮　钱程　彭玉华　副主编

清华大学出版社

高等学校计算机应用规划教材

高级语言程序设计

(C语言)

阳小兰 主编

吴亮 钱程 彭玉华 副主编

清华大学出版社

北京

内 容 简 介

C 语言具有强大的功能和灵活的处理能力，是风靡全球的高级程序设计语言之一。本书全面深刻地讲解 C 语言程序设计。首先讲述 C 语言的程序结构、开发环境、数据类型、运算符、表达式；然后讨论顺序结构、选择结构、循环结构程序设计，分析一维数组、二维数组、字符数组、字符串、函数定义、嵌套调用、递归调用，介绍指针变量、结构体、共用体、自定义类型、编译预处理；最后呈现宏定义、条件编译、文件分类、缓冲区、文件常用操作等主题。在阅读全书后，读者将彻底了解 C 语言编程，将有能力和信心开发自己的 C 应用程序。

本书结构清晰、案例丰富，可用作高等学校计算机相关专业的教材，也可用作程序设计人员的培训教材，并可供广大编程爱好者参考。

图书在版编目(CIP)数据

高级语言程序设计(C 语言) / 阳小兰　主编. —北京：清华大学出版社，2018

(高等学校计算机应用规划教材)

ISBN 978-7-302-49942-8

Ⅰ. ①高…　Ⅱ. ①阳…　Ⅲ. ①C 语言－程序设计－高等学校－教材　Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 066163 号

责任编辑： 刘金喜　韩宏志
封面设计： 牛艳敏
版式设计： 思创景点
责任校对： 孔祥峰
责任印制： 丛怀宇

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

印 装 者： 三河市少明印务有限公司
经　　销： 全国新华书店
开　　本： 185mm×260mm　　**印　　张：** 20　　**字　　数：** 462 千字
版　　次： 2018 年 5 月第 1 版　　**印　　次：** 2018 年 5 月第 1 次印刷
印　　数： 1～2500
定　　价： 49.80 元

产品编号：079218-01

前　言

C 语言是广泛使用的高级程序设计语言之一，具有强大的功能和灵活的处理能力，它既可编写系统程序，又可编写应用程序，深受程序设计者喜爱。目前，很多高校都将 C 语言作为高级语言程序设计的首选语言。

作者在多年 C 语言教学、研究和实践积累的基础上，吸收国内外 C 语言程序设计课程的教学理念和方法，依据 C 语言程序设计课程教学大纲的要求编写了本书。本书遵循教学规律，按照由浅入深、循序渐进的原则，精心设计，合理安排。本书语言组织简明易懂，既突出阐明原理和方法，又保证有一定的实用性，有一定的广度和深度。

本书共分 11 章：

第 1 章“C 语言概述”介绍程序及算法的概念、C 语言的发展历程和特点、C 语言的程序结构以及 C 语言程序的开发环境。

第 2 章“C 语言程序设计基础”介绍 C 语言数据的表现形式、C 语言的数据类型、C 语言的运算符和表达式、数据类型转换及位运算。

第 3 章“顺序结构程序设计”介绍 C 语言的基本语句、字符数据的输入输出、格式输入输出，并列举顺序结构程序示例。

第 4 章“选择结构程序设计”介绍关系运算符与关系表达式、逻辑运算符与逻辑表达式、条件运算符与条件表达式、if 语句和 switch 语句，并列举选择结构程序设计示例。

第 5 章“循环结构程序设计”介绍 while 循环、do…while 循环、for 循环三种结构，讨论循环结构中常用的 break 语句和 continue 语句，并列举循环嵌套和循环结构程序示例。

第 6 章“数组”介绍一维和二维数组的定义、引用和初始化，并讨论字符数组与字符串。

第 7 章“函数”介绍函数定义、函数调用、数组作为函数的参数、函数的嵌套调用与递归调用、变量的作用域以及存储方式。

第 8 章“指针”介绍指针的概念、指针变量、指针与数组、指针与字符串、指向函数的指针、返回指针的函数以及指针数组。

第 9 章“结构体、共用体与自定义类型”介绍结构体的概念、结构体数组、指向结构体类型数据的指针、共用体以及用 typedef 定义数据类型。

第 10 章“编译预处理”介绍带参数与不带参数的宏定义、文件包含及条件编译。

第 11 章“文件”介绍文件的分类、缓冲区及文件类型的指针；讨论文件的常用操作，包括文件的打开与关闭、文件的读写、文件的定位等。

本书每一章都列举大量程序案例，在编程实践中讲解知识点，实现“从做中学”的教育理念。在例题的编排上由浅入深、逐层递进，内容紧扣基础、面向应用，循序渐进地引导学生学习程序设计的思想和方法。每章末尾都给出一定数量的习题，以此训练和培养学生设计程序的能力。读者可访问 http://www.tupwk.com.cn/downpage，输入本书书名或 ISBN，

下载课件和源代码。

本书可作为高等学校计算机相关专业的教材，也可作为程序设计人员的培训教材，并可供广大编程爱好者参考。

本书在武昌理工学院信息工程学院的指导下，由阳小兰负责统稿。第 4、5、6、7、8、10 章及附录由阳小兰编写，第 1、2、3 章由吴亮编写，第 9 章由钱程编写，第 11 章由彭玉华编写。

本书在编写过程中得到了武昌理工学院信息工程学院的领导与同仁的大力支持，也得到了清华大学出版社的大力支持，在此表示衷心感谢。在编写过程中，我们力求做到严谨细致、精益求精，但由于时间仓促、编者水平有限，书中疏漏和不妥之处在所难免，敬请各位读者和同行专家批评指正。

编　者

2017 年 12 月于武昌理工学院

目　录

第1章　C语言概述

本章教学内容：

- 结构化程序设计的基本概念
- 算法的基本概念与特征
- C 语言的程序结构
- C 语言程序的开发环境

本章教学目标：

- 理解程序、程序设计和算法的相关概念
- 了解程序设计语言的发展历程及 C 语言的特点
- 能正确运用 C 语言的关键字及标识符
- 掌握 C 语言源程序的结构，能熟练编写简单的 C 程序
- 能运用 VC++集成开发环境创建、编辑、链接和运行简单的 C 程序

1.1　程序设计及算法

1.1.1　程序及程序设计

1. 程序

在日常生活中，程序是为进行某项活动或过程所规定的步骤，可以是一系列动作、行动或操作，如新生报到程序、银行取款程序、面包制作程序。在计算机世界中，软件将一组程序组织起来，每个程序由一组指令组成。程序是计算机能识别和执行的一组指令，告诉计算机如何完成一项具体任务，如完成银行取款程序需要以下 5 个步骤。

第 1 步：带上存折去银行；

第 2 步：填写取款单并到相应窗口排队；

第 3 步：将存折和取款单递给银行职员；

第 4 步：银行职员办理取款事宜；

第 5 步：拿到钱并离开银行。

只要让计算机执行这个程序，计算机就会自动地、有条不紊地进行工作。程序就是为了让计算机执行某些操作或解决某个问题而编写的一系列有序指令的集合。计算机的一切操作都由程序控制，离开程序，计算机将一事无成。

对于编写程序的软件开发人员来说，程序是以某种程序设计语言为工具编制出来的指令序列，它表达了人类解决现实世界问题的思想。计算机程序是用计算机程序设计语言所要求的规范书写出来的一系列步骤，它表达了软件开发人员要求计算机执行的指令。

数据是程序操作的对象，操作数据的目的是对数据进行加工处理，以得到程序期望的结果。一个程序通常主要包括以下两方面的信息：

(1) **对数据的描述。**在程序中要指定用到哪些数据，并指定这些数据的类型和组织形式，这就是数据结构。

(2) **对操作的描述。**即要求计算机执行操作的步骤，也就是算法。

2. 程序设计

程序设计是软件构造活动中的重要组成部分，是人们借助计算机语言，告诉计算机要做什么(即处理哪些数据)，如何处理(即按什么步骤来处理)的过程。例如，C 语言程序设计是以C语言为工具，编写各类 C 语言程序的过程。程序设计的过程通常应当包括分析问题、设计算法、编写程序、运行程序、分析结果、编写程序文档等不同阶段。

(1) **分析问题：**即分析、研究给定的条件以及最终目标，找出解决问题的规律，选择解题的方法，完成实际问题。

(2) **设计算法：**即设计出解题的方法和具体步骤。

(3) **编写程序：**即将算法翻译成所需的计算机程序设计语言，再对用该语言编写的源程序进行编辑、编译和链接。

(4) **运行程序：**即运行可执行程序，得到运行结果。

(5) **分析结果及调试：**即分析运行结果是否合理，如不合理需要对程序进行调试。

(6) **编写程序文档：**程序是供别人使用的，如同正式的产品应当提供产品说明书一样，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。

专业的程序设计人员通常称为程序员，程序员采用不同的程序设计方法来设计和开发程序。程序设计方法通常有结构化程序设计与非结构化程序设计之分，本书所涉及的 C 语言采用结构化程序设计方法。结构化程序设计思想采用模块分解、功能抽象、自顶向下、分而治之等方法，从而有效地将一个较复杂的程序系统设计任务分解成许多易于控制和处理的子程序，便于开发和维护。

C 语言是以函数形式提供给用户的，这些函数既可方便地调用，也可由多种循环、条件语句控制程序流向，从而使程序完全结构化。从程序流程的角度看，程序可以分为三种基本结构，即顺序结构、选择(分支)结构、循环结构。三种结构的流程图如图 1-1 所示。三种基本结构可组成各种复杂的 C 语言程序，这三种基本结构将在本书的第 3 章、第 4 章和第 5 章进行详细讲解。

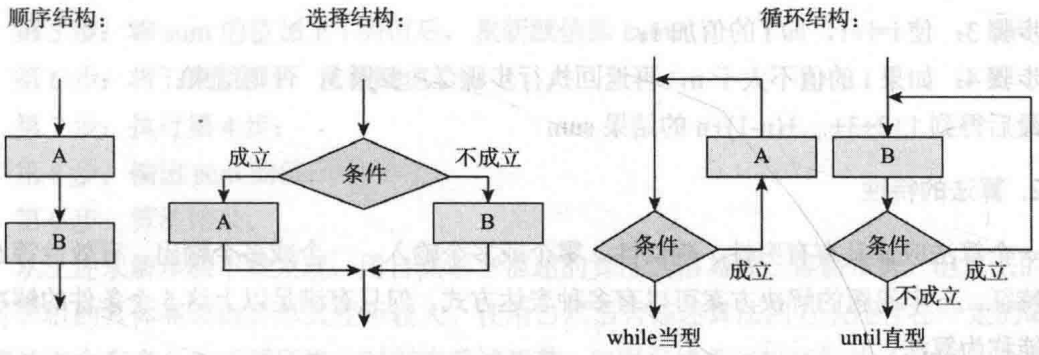


图 1-1　三种结构的流程图

1.1.2　算法

计算机系统中的任何软件，都是由大大小小的各种软件组成部分构成，各自按照特定的算法来实现。用什么方法来设计算法，所设计算法需要什么样的资源，需要多少运行时间、多少存储空间，如何判定一个算法的好坏，这些都是实现一个软件时必须予以解决的。算法的好坏直接决定所实现软件性能的优劣，因此，算法设计与分析是程序设计中的一个核心问题。

1. 算法的基本概念

著名计算机科学家沃思(Nikiklaus Wirth)提出一个公式：算法 + 数据结构 = 程序，算法是程序的灵魂，数据结构是程序的加工对象。实际上，一个程序除了算法和数据结构这两个要素外，还应当采用结构化程序设计方法进行程序设计，并用某一种计算机语言表示。因此，算法、数据结构、程序设计方法和语言工具是一个程序设计人员应该掌握的知识。

算法是解决问题的方法和具体步骤，如求长方形的面积问题的算法如下：

步骤 1：接收用户输入的长方形长度和宽度两个值；

步骤 2：判断长度和宽度的值是否大于零；

步骤 3：如果大于零，将长度和宽度两个值相乘得到面积，否则显示输入错误；

步骤 4：显示面积。

下面用原始解题步骤和计算机算法表示，给出解决 sum=1+2+3+…+(n-1)+n 的算法。

(1) 原始解题步骤算法表示：

步骤 1：先求 1+2，得到 1+2 的结果：3

步骤 2：将步骤 1 的结果加 3，得到 1+2+3 的结果：6

步骤 3：将步骤 2 的结果加 4，得到 1+2+3+4 的结果：10

步骤 4：将步骤 3 的结果加 5，得到 1+2+3+4+5 的结果：15

……

步骤 n-1：将步骤 n-2 的结果加 n，得到 1+2+3+…+(n-1)+n 的结果：sum

(2) 用计算机算法表示：

步骤 1：使 sum=0 和 i=1；

步骤 2：使 sum=sum+i，结果仍放在 sum 中；

步骤 3：使 i=i+1，即 i 的值加 1；

步骤 4：如果 i 的值不大于 n，再返回执行步骤 2、步骤 3，否则结束；

最后得到 1+2+3+…+(n-1)+n 的结果 sum。

2. 算法的特性

一个算法应该具有有穷性、确切性、零个或多个输入、一个或多个输出、有效性等 5 个重要特征。一个问题的解决方案可以有多种表达方式，但只有满足以上这 5 个条件的解决方案才能称为算法。

(1) **有穷性**：无论算法有多么复杂，都必须在执行有限步骤之后结束并终止运行，即算法的步骤必须是有限的。任何情况下，算法都不能陷入无限循环中，也就是说一个算法的实现应该在有限时间内完成。如求 sum=1+2+3+…+(n-1)+n 的算法是执行语句 sum=sum+i;累加运算到 i 等于 n 后终止。

(2) **确切性**：算法的每一个步骤必须有确切的定义，算法中对每个步骤的解释是唯一的，每个步骤都有确定的执行顺序，即上一步在哪里，下一步是什么，都必须明确，无二义性。

(3) **零个或多个输入**：输入是指在执行算法时需要从外界取得的必要信息。一个算法有零个或多个输入，这些输入的信息有的在执行过程中输入，而有的已被嵌入到算法中。一个算法可以没有输入，如求 sum=1+2+3+…+(n-1)+n 的算法就没有输入；也可以有多个输入，如求长方形的面积问题的算法有长方形长度和宽度两个输入。

(4) **一个或多个输出**：输出是算法的执行结果，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。一个算法必须有一个输出，没有输出结果的算法是没有任何意义的。如求长方形的面积问题的算法有长方形的面积一个输出；如求 sum=1+2+3+…+(n-1)+n 的算法有累加和 sum 一个输出。

(5) **有效性**：又称可行性。算法的有效性指的是算法中待实现的运算，都是基本的运算，原则上可由人们用纸和笔，在有限时间里精确完成。算法首先必须是正确的，都是能够精确执行的。如对于任意一组输入，包括合理的输入与不合理的输入，总能得到预期的输出。如果一个算法只是对合理的输入才能得到预期的输出，在异常情况下却无法预料输出的结果，它就不是正确的。

3. 算法的描述

算法的常用表示方法有使用自然语言描述算法，使用流程图描述算法，使用伪代码描述算法 3 种。

(1) **使用自然语言描述算法**：所谓“自然语言”指的是日常生活中使用的语言，如汉语、英语或数学语言。使用自然语言描述求 sum=1+2+3+4+5+…+(n−1)+n 的算法如下：

第 1 步：给定一个大于 0 的正整数 n 的值；

第 2 步：定义一个整型变量 i，设其初始值为 1；

第 3 步：再定义一个整型变量 sum，其初始值设置为 0；

第 4 步：如果 i 小于等于 n，则转第 5 步，否则执行第 8 步；

第 5 步：将 sum 的值加上 i 的值后，重新赋值给 sum；

第 6 步：将 i 的值加 1，重新赋值给 i；

第 7 步：执行第 4 步；

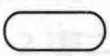
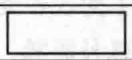
第 8 步：输出 sum 的值；

第 9 步：算法结束。

从上述求解步骤不难发现，用自然语言描述的算法通俗易懂，容易掌握，但算法的表达与计算机的具体高级语言形式差距较大。使用自然语言描述算法的方法还存在一定的缺陷，当算法中含有多分支或循环操作时很难表述清楚。使用自然语言描述算法还很容易造成歧义(又称二义性)，可能使他人对同一句话产生不同的理解。

(2) **使用流程图描述算法**：流程图也叫框图，它是用各种几何图形、流程线及文字说明来描述求解过程的框图。流程图的符号采用美国国家标准化协会(ANSI)规定的一些常用符号，如表 1-1 所示。流程图使用一组预定义的符号来说明如何执行特定任务，sum=1+2+3+…+(n-1)+n 的算法流程图如图 1-2 所示。流程图是算法的一种图形化表示方式，流程图直观、清晰，更有利于人们设计与理解算法。

表 1-1　常用流程图的符号

符　号	名　称	作　　用
	起止框	表示算法的开始和结束符号
	输入 输出框	表示算法过程中，从外部获取信息(输入)，然后将处理过的信息输出
	判断框	表示算法过程中的分支结构。菱形框的 4 个顶点中，通常用上面的顶点表示入口，根据需要用其余顶点表示出口
	处理框	表示算法过程中，需要处理的内容，只有一个入口和一个出口
	流程线	在算法过程中指向流程的方向
	连接点	在算法过程中用于将画在不同地方的流程线连接起来
	注释框	对流程图中某些框的操作进行必要的补充说明，可以帮助读者更好地理解流程图的作用。不是流程图中的必要部分

(3) **使用伪代码描述算法**：伪代码是一种介于自然语言与计算机语言之间的算法描述方法。它结构性较强，比较容易书写和理解，修改起来也相对方便。其特点是不拘泥于语言的语法结构，而着重以灵活的形式表现被描述对象。它利用自然语言的功能和若干基本控制结构来描述算法。伪代码没有统一的标准，可以自己定义，也可以采用与程序设计语言类似的形式。如使用伪代码描述求 sum=1+2+3+4+5+…+(n−1)+n 的算法如下。

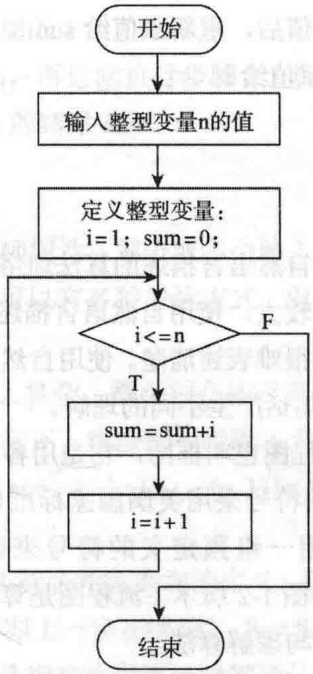


图 1-2　求和的算法流程图

算法开始：

第 1 步：输入 n 的值；

第 2 步：置 i 的初值为 1；

第 3 步：置 sum 的初值为 0；

第 4 步：当 i <=n 时，执行下面的操作

第 4.1 步：使 sum =sum + i;

第 4.2 步：使 i = i + 1;

(循环体到此结束)

第 5 步：输出 sum 的值；

算法结束。

伪代码是一种用来书写程序或描述算法时使用的非正式、透明的表述方法。它并非是一种编程语言，这种方法针对的是一台虚拟计算机。伪代码通常采用自然语言、数学公式和符号来描述算法的操作步骤，同时采用计算机高级语言(如 C、Pascal、VB、C++、Java 等)的控制结构来描述算法步骤的执行顺序。伪代码书写格式比较自由，容易表达出设计者的思想，写出的算法很容易修改，但用伪代码写的算法不如流程图直观。

1.2　程序设计语言

1.2.1　程序设计语言的发展历程

人与人之间交流的主要语言工具是各国的语言，如汉语、英语、俄语、法语等。那么，

在信息化高速发展的今天，我们人类与计算机交流信息也需要语言，需要一种人和计算机都能识别的语言，这就是用于编写计算机程序的程序设计语言。程序设计语言，要解决的问题有两个，一个是方便软件开发人员“表达”，一个是让计算机“听懂”。围绕着这两个问题，程序设计语言大约经历了机器语言、汇编语言、高级语言三个发展阶段。

1. 第一代程序设计语言：机器语言

计算机发明之初，人们只能用计算机的语言去命令计算机做事情。作为机器的计算机，只懂电路通(用 1 表示)与不通(用 0 表示)两种状态，人们只能写出一串串由“0”和“1”组成的指令序列交由计算机执行，这种计算机能识别的语言就是机器语言。由“0”和“1”组成的二进制数，是程序设计语言的基础，如用“10000000” 表示加法指令，“10010000”表示减法指令。

机器语言是完全面向机器的语言，可由计算机直接识别和运行，拥有极高的执行效率，这是机器语言的最大优点。机器语言只有“0”“1”两种信息，难学、难写、难记，给软件开发人员阅读、编写和调试程序等操作带来极大不便，难以推广使用；并且面向机器的机器语言相当依赖机器，硬件设备不同的计算机，它的机器语言也有差别，编写的程序缺少通用性。编写机器语言要求软件开发人员相当熟悉计算机的硬件结构，所以初期只有极少数计算机专业人员会编写机器语言。

2. 第二代程序设计语言：汇编语言

考虑到机器语言难以理解记忆，后来人们用一些简洁、有意义的英文字母、符号串来替代一个特定指令的二进制串，比如，用“ADD”代替“10000000”表示加法，“ADD　A, B”表示 A、B 两个操作数相加。这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言称为汇编语言，即第二代程序设计语言。

然而计算机并不认识这些符号，需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。由于要请“翻译”，所以汇编语言相对机器语言，执行效率有所降低。汇编语言的实质和机器语言是相同的，都是直接对硬件操作，只不过指令采用了英文缩写的标识符，更容易识别和记忆。汇编语言同样十分依赖于机器硬件，不同的计算机硬件设备需要不同的汇编语言指令，不利于在不同计算机系统之间移植。所以，现在的汇编语言一般在专业程序设计人员中使用，主要用于控制系统、病毒的分析与防治、设备驱动程序的编写。

3. 第三代程序设计语言：高级语言

机器语言和汇编语言都更“贴近”机器，更“依赖”机器，是面向机器的程序设计语言，统称为低级语言。为克服低级语言的缺点，将程序设计的重点放在解决问题的方法(即算法)上，于是产生了面向过程和面向对象的第三代程序设计语言，即高级语言。高级语言更接近人们习惯使用的自然语言和数学语言，如用“a+b”表示 a、b 两个变量相加，“sin(a)”表示对变量 a 进行正弦计算。

高级语言是绝大多数编程者的选择。与低级语言不同，用高级语言编写的程序可在不同

的计算机系统中运行，这个特性大大减轻了软件开发人员的负担，使他们不用了解计算机底层的知识，而将精力放在应用系统逻辑上。所以，用高级语言编写的程序与硬件设备无关，适合开发解决各种实际应用问题的应用软件。

用高级语言编写的程序不能直接在操作系统上运行，执行时需要根据计算机系统的不同，将程序代码翻译成计算机可直接运行的机器语言。这个工作一般都由高级语言系统自动进行翻译处理。一般将用高级语言编写的程序代码称为“源程序”，将翻译后的机器语言代码称为“目标程序”。计算机将源程序翻译成目标程序，有两种翻译方式，一种是“解释”方式，一种是“编译”方式。对应于这两种翻译方式，高级语言又可分为解释性语言和编译性语言。

综上所述，程序设计语言越低级，就表明越靠近机器，执行效率越高；程序设计语言越高级，就表明越靠近人的表达与理解，机器依赖程度越低。程序设计语言的发展，是从低级到高级，直到可用人类的自然语言来描述。程序设计语言的发展也是从具体到抽象的发展过程，从面向过程发展到面向对象。在以后的教学中，C 语言作为面向过程的高级语言代表，C++、Java 作为面向对象的高级语言代表。

1.2.2　C 语言的发展历程

数十年来，全球涌现了 2500 多种高级语言，每种语言都有其特定的用途。随着程序设计语言的发展，优胜劣汰，现在应用比较广泛的仅 100 多种。C 语言是目前世界上最流行、使用最广泛的高级程序设计语言之一。与其同时代的很多高级语言已经消亡，C 语言凭借其兼顾高级语言与低级语言的特性，作为高级语言的鼻祖留存至今。

C 语言的原型是 ALGOL 60 语言，1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL (Combined Programming Language)语言。1967 年，剑桥大学的 Matin Richards 对 CPL 语言进行了简化，于是产生了 BCPL 语言。1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改，并为它起了一个有趣的名字“B 语言”。意思是将 CPL 语言煮干(Boiled)，提炼出它的精华。并且他用 B 语言写了第一个UNIX 操作系统。1973 年，B 语言也给人“煮”了一下，美国贝尔实验室的 Dennis M. Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他取了 BCPL 的第二个字母作为这种语言的名字，这就是 C 语言。

为使 UNIX 操作系统推广，1977 年 Dennis M. Ritchie 发表了不依赖于具体机器系统的 C 语言编译文本《可移植的 C 语言编译程序》。1978 年 Brian W. Kernighan 和 Dennis M. Ritchie 出版了名著 *The C Programming Language*，从而使 C 语言成为目前世界上最广泛流行的高级程序设计语言。1988 年，随着微型计算机的日益普及，出现了许多 C 语言版本。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。为改变这种情况，美国国家标准研究所(ANSI)为 C 语言制定了一套 ANSI 标准，成为现行的 C语言标准。

现行的 C 语言国际标准有 2 个，分别是 ANSI/ISO 9988-1990 和 ISO/IEC 9989-1999，分别在 1989 年和 1999 年通过，也就是我们常说的 C89 和 C90。目前不同软件公司提供的各种 C 语言编译系统多数并未完全实现 C99 建议的功能，本书的叙述以 C99 标准为依据，书中的程序基本上都可在目前所用的编译系统(如 Visual C++ 6.0，Turbo C++ 3.0，GCC)上编译和运行。

1.2.3 C语言的特点

C语言是一种比较特殊的高级语言，它的主要特色是兼顾了高级语言和汇编语言的特点，简洁、丰富、可移植，程序执行效率高。C语言是一种用途广泛、功能强大、使用灵活的过程性编程语言，既可用于编写应用软件，又能用于编写系统软件。因此C语言问世以后得到迅速推广，并应用至今。C语言主要特点如下：

(1) **语言简洁、紧凑、使用方便、灵活。**C语言只有32个关键字、9种控制语句，程序书写形式自由，源程序代码短。

(2) **运算符丰富。**C语言有34种运算符和15个等级的运算优先级顺序，使表达式类型多样化，可实现在其他语言中难以实现的运算。

(3) **数据类型丰富。**C语言包括整型、浮点型、字符型、数组类型、指针类型、结构体类型、共用体类型；C99又扩充了复数浮点类型、超长整型、布尔类型等；尤其是指针类型数据，使用十分灵活，能用来实现各种复杂的数据结构的运算。

(4) **具有结构化的控制语句。**C语言是结构化语言，有顺序、选择(分支)、循环三大结构，含9种控制语句。C语言是完全模块化语言，用函数作为程序的模块单位，便于实现程序的模块化。

(5) **语法限制不太严格，程序设计自由度大。**C语言允许程序编写者有较大的自由度，因此放宽了语法检查，如对数组下标越界不做检查；对变量的类型使用比较灵活，如整型与字符型数据可以通用。

(6) **C语言允许直接访问物理地址，从而可以直接对硬件进行操作。**C语言可以像汇编语言一样对位、字节和地址这3个最基本的工作单元进行操作，能实现汇编语言的大部分功能，可用来编写系统软件。

(7) **用C语言编写的程序可移植性好。**C的编译系统简洁，很容易移植到新系统；在新系统上运行时，可直接编译“标准链接库”中的大部分功能，不需要修改源代码；几乎所有计算机系统都可以使用C语言。

(8) **生成目标代码质量高，程序执行效率高。**用不同的程序设计语言编写相同功能的程序，C语言生成的目标代码比其他语言生成的目标代码质量高，执行效率高。一般只比汇编程序生成的目标代码效率低10%～20%。

1.3 C语言的程序结构

1.3.1 C语言程序的基本词汇符号

任何一门高级语言，都有自己的基本词汇符号和语法规则，就像汉语有汉字和语法一样。程序代码都是由这些基本词汇符号，根据该高级语言的语法规则编写的。以C语言为例，C语言有自己的字符集、关键字、标识符等各种基本词汇符号，各种语法规则将在后续章节中学习。