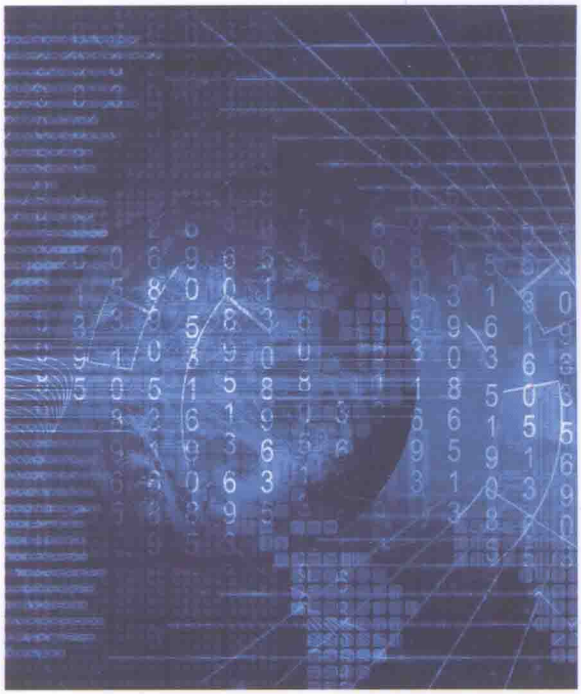
高等学校计算机应用规划教材

程序设计基础
——C语言

- 实例讲解、循序渐进
- 强化算法、侧重应用
- 实例丰富、一题多解
- 贴心提示、图文并茂
- 习题丰富、资源完善



金兰　主　编
梁洁　副主编

清华大学出版社

高等学校计算机应用规划教材

程序设计基础
——C语言

主　编　金兰

副主编　梁洁

清华大学出版社

北　京

内 容 简 介

C 语言是国内外广泛使用的编程语言，已被大多数高等学校作为典型的计算机教学语言。本书分 10 章，内容包括：C 语言概述，数制和基本数据类型，运算符和表达式、输入输出，控制结构，数组，函数，指针，结构体与共用体，文件，综合应用案例，以及 4 个附录。

本书深入浅出，例题丰富，侧重程序设计思维的构建和程序算法的分析与设计。本书采用“提出问题→算法分析→程序实现→说明归纳”的步骤组织教材内容，符合读者的认知规律，强化了算法的分析和设计，有助于帮助读者建立良好的思维模式，培养读者分析问题和解决问题的能力，掌握软件开发的工作原理和系统方法。本书中的典型程序一题多解，有助于新旧知识对比学习，融会贯通，启迪思维，拓展读者的程序设计能力和灵活运用的能力。

本书可作为高等学校各相关专业“程序设计基础”、“C 语言程序设计”课程的教材，也可作为程序开发人员的培训教程，还可作为全国计算机等级考试、编程爱好者学习参考用书。

本书还特别为任课教师免费提供整套教学资源(电子课件、全部程序源代码和习题参考答案等)，请从 http://www.tupwk.com.cn/downpage 下载。本书所配的相关上机环节指导书《程序设计基础上机指导——C 语言》，建议与本书配套使用。

本书封面贴有清华大学出版社防伪标签，无标签者不得销售。
版权所有，侵权必究。侵权举报电话：010-62782989　13701121933

图书在版编目(CIP)数据

程序设计基础：C 语言/金兰　主编. —北京：清华大学出版社，2016
(高等学校计算机应用规划教材)
ISBN 978-7-302-42444-4

Ⅰ. ①程…　Ⅱ. ①金…　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 299523 号

责任编辑：刘金喜
封面设计：孔祥峰
版式设计：思创景点
责任校对：牛艳敏
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，010-62794504

印 装 者：清华大学印刷厂
经　　销：全国新华书店
开　　本：185mm×260mm　　印　　张：25.25　　字　　数：583 千字
版　　次：2016 年 2 月第 1 版　　印　　次：2016 年 2 月第 1 次印刷
印　　数：1～2000
定　　价：43.00 元

产品编号：065458-01

前　言

程序设计基础的入门课程——C 语言是目前广泛应用的程序设计语言之一。它具有功能强大、使用灵活、可移植性好的特点，同时兼备低级语言和高级语言的优点，可用于编写系统软件和应用软件。另外，C 语言的语法规则清晰，便于掌握和记忆，因此适合作为大多数人学习计算机程序设计的入门语言。通过本书的学习，可以加深学生对计算机系统的认识；建立良好的计算机思维模式；培养学生模块化、结构化编程方法与技巧；训练学生运用计算机分析问题和解决问题的实践能力；熟练使用 Visual C++ 6.0 开发环境进行 C 语言编程、调试、运行等各个环节的基本操作，为今后进一步学习打下坚实的基础。

本书是作者在多年 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而成的。

本书每一章都配备了大量的例题讲解，所有程序例题均在 Visual C++ 6.0 平台中调试通过。程序例题采用了“问题提出→问题分析→算法分析→程序实现→说明归纳”的步骤来讲解，符合读者的认知规律，对例题的重点难点位置强化算法的分析和设计，有助于建立读者良好的思维模式，培养读者分析问题和解决问题的能力。本书最后通过一个综合应用案例——学生学籍管理系统，按照软件工程的思想，沿着“需求分析→总体设计→详细设计→编码实现”的软件开发流程，完整地开展系统的分析设计与实现，有助于读者掌握软件开发的工作原理和系统方法。

全书共分为 10 章，具体内容如下：

第 1 章　讲述计算机编程语言的发展过程、在 Visual C++ 6.0 集成开发环境中编写和调试控制台程序的步骤和方法。

第 2～3 章　讲解数据类型、运算符和表达式的使用方法、基本输入输出函数的应用。

第 4 章　讲述运用三种基本的控制结构(顺序、选择和循环)进行编程的方法。

第 5～6 章　讲解数组和字符串的运用、函数的使用、变量的作用域与生存期、编译预处理命令。

第 7～8 章　讲解指针、结构体、共用体的使用方法和链表的相关操作。

第 9 章　讲解文件操作的标准库函数的应用。

第 10 章　完整讲解一个综合应用案例——学生学籍管理系统的分析设计与实现的全过程。

本书具有以下特色：

1. 实例丰富。本书不仅理论完备，还通过 100 多个实例夯实基础，100 多个课后习题巩固练习，并通过分布在本书第 6、8 和 10 章的 3 个综合应用案例——学生成绩统计程序、学生成绩查询系统、学生学籍管理系统全面提升实战开发能力。

2. 一题多解。典型实例可采用多种算法来设计和实现，有助于新旧知识对比学习，融会贯通，启迪思维，拓展读者的程序设计能力和灵活运用能力。

3. 贴心提示。为了便于读者阅读，书中还穿插了一些说明、注意和思考等小贴士，体例约定如下：

“说明”：进一步阐述相关知识点的应用，力求规范、全面。

“注意”：指出学习过程中需要特别注意的一些知识点和内容，让读者加深印象。同时，还为读者提供建议及解决问题的方法。

“思考”：读者可利用课余时间独立思考、解决提出的问题，进一步深入学习训练。

4. 习题丰富。本书每章最后提供了大量习题，涵盖了每章知识的重难点内容。题型灵活多样，包括选择题、填空题、阅读程序填空题及编程题，方便读者课后巩固练习。

本书可作为高等学校各相关专业“程序设计基础”、“C 语言程序设计”课程的教材，也可作为程序开发人员的培训教程，还可作为全国计算机等级考试、编程爱好者的学习参考用书。

本书还特别为任课教师免费提供整套教学资源(电子课件、全部程序源代码和习题参考答案等)，请从 http://www.tupwk.com.cn/downpage 下载。本书还配有相关上机环节指导书《程序设计基础上机指导——C 语言》(ISBN 978-7-302-42445-1)，建议与本书配套使用。

本书的统稿工作由金兰负责，第 1、2、3、4、5、7、10 章及附录由金兰编写，第 6、8 章由梁洁编写，第 9 章由姚炜编写。在本书编写过程中，王育勤教授给予了诸多的鼓励和关心。书稿中例题和习题程序的调试过程，杨景莹和杨明等做了大量工作。本书在编写过程中得到了许多同行的帮助，还参阅了许多相关资料，在此衷心的感谢。因编者水平有限，书中难免会有疏漏和错误之处，恳请广大读者给予指正。

服务邮箱：wkservice@vip.163.com

编　者

试读结束，需要全本PDF请购买 www.ertongbook.com

目　录

第 1 章　C 语言概述 …… 1
1.1　计算机编程语言 …… 1
1.1.1　机器语言 …… 1
1.1.2　汇编语言 …… 2
1.1.3　高级语言 …… 3
1.2　第一个 C 程序 …… 5
1.3　C 程序的上机步骤 …… 7
1.3.1　单文件的 C 程序的上机步骤 …… 8
*1.3.2　多文件的 C 程序的上机步骤 …… 11
1.4　C 程序的调试 …… 17
课后习题 1 …… 21

第 2 章　数制、基本数据类型 …… 23
2.1　整数数制 …… 23
2.1.1　十进制数 …… 23
2.1.2　二进制数 …… 23
2.1.3　八进制数 …… 25
2.1.4　十六进制数 …… 26
2.2　C 程序常见符号分类 …… 27
2.3　数据类型 …… 29
2.3.1　为什么引入数据类型 …… 29
2.3.2　类型修饰符 …… 30
*2.3.3　C99 标准中的新增类型 …… 31
2.4　常量 …… 32
2.4.1　整型常量 …… 32
2.4.2　实型常量 …… 33
2.4.3　字符常量 …… 33
2.4.4　字符串常量 …… 35
2.4.5　符号常量 …… 35
2.4.6　枚举常量 …… 36
2.5　变量 …… 37
2.5.1　变量的声明与初始化 …… 37
2.5.2　const 类型修饰符 …… 38
2.5.3　变量的类型 …… 39
课后习题 2 …… 42

第 3 章　运算符和表达式、输入输出 … 45
3.1　算术运算符 …… 45
3.2　赋值运算符 …… 47
3.3　增 1、减 1 运算符 …… 48
3.4　关系运算符 …… 49
3.5　逻辑运算符 …… 50
3.6　条件运算符 …… 52
3.7　强制类型转换运算符 …… 52
3.8　逗号运算符 …… 53
3.9　位运算符 …… 54
3.10　sizeof 运算符 …… 56
3.11　类型转换 …… 57
3.12　运算符的优先级和结合性 …… 59
3.13　基本输入输出函数 …… 60
3.13.1　字符输入输出函数 …… 61
3.13.2　格式化输入输出函数 …… 63
课后习题 3 …… 74

第 4 章　控制结构 …… 78
4.1　算法及其描述方法 …… 78
4.1.1　算法的概念 …… 78
4.1.2　算法的描述方法 …… 79
4.2　顺序结构 …… 81
4.3　选择结构 …… 83
4.3.1　if 语句 …… 84
4.3.2　switch 语句 …… 93
4.4　循环结构 …… 102

4.4.1 while 语句……103
4.4.2 do…while 语句……105
4.4.3 for 语句……108
4.4.4 三种循环控制语句的应用举例……111
4.4.5 循环的嵌套……116
4.4.6 提前结束循环……120
4.5 综合应用举例……123
课后习题 4……130

第 5 章 数组……136
5.1 一维数组……136
5.1.1 一维数组的定义……136
5.1.2 一维数组的引用……137
5.1.3 一维数组的初始化……137
5.1.4 一维数组程序举例……139
5.2 二维数组……148
5.2.1 二维数组的定义……148
5.2.2 二维数组的引用……149
5.2.3 二维数组的初始化……150
5.2.4 二维数组程序举例……151
5.3 字符数组与字符串……155
5.3.1 字符数组的初始化……155
5.3.2 字符数组的输入/输出……157
5.3.3 字符串处理函数……158
5.3.4 字符数组和字符串程序举例……162
课后习题 5……168

第 6 章 函数……172
6.1 函数的定义……175
6.1.1 函数的分类……176
6.1.2 函数的定义……176
6.2 函数的调用、参数和返回值……177
6.3 函数的声明……179
6.4 函数的嵌套调用……183
*6.5 函数的递归调用……185
6.5.1 递归问题的提出……185
6.5.2 递归函数……186
6.6 数组作为函数参数……190
6.6.1 一维数组作为函数参数……190
6.6.2 二维数组作为函数参数……192
6.7 变量的作用域与生存期……194
6.7.1 局部变量……194
6.7.2 全局变量……195
6.7.3 变量的存储类别……196
6.7.4 小结……201
6.8 内部函数和外部函数……202
6.8.1 内部函数……202
6.8.2 外部函数……202
6.9 预处理命令……204
6.9.1 宏定义……205
6.9.2 文件包含……209
6.9.3 条件编译……210
6.10 综合应用举例……212
课后习题 6……219

第 7 章 指针……225
7.1 内存、地址和内容……225
7.2 指针与指针变量……226
7.2.1 指针变量的定义……226
7.2.2 指针变量的引用……227
7.2.3 指针变量作为函数参数……230
7.3 指针与数组……233
7.3.1 指向一维数组的指针……233
7.3.2 有关指针的运算……236
7.3.3 一维数组的指针作为函数参数……237
7.3.4 指向二维数组的指针……242
7.3.5 二维数组的指针作为函数参数……245
7.4 指针与字符串……248
7.4.1 指向字符串的指针变量……248
7.4.2 指向字符串的指针作为函数参数……249

7.4.3 字符数组与字符串指针变量的区别……252
7.5 指针与函数……253
7.5.1 返回指针值的函数……253
*7.5.2 指向函数的指针……255
7.6 指针数组……256
*7.7 指向指针的指针……259
*7.8 带参数的函数 main()……261
7.9 动态内存分配……263
7.9.1 动态内存分配函数……263
*7.9.2 动态内存分配与变长数组…267
*7.10 ANSI C 的类型限定词 const……268
课后习题 7……270
第 8 章 结构体与共用体……277
8.1 问题的引出……277
8.2 结构体类型和结构体类型变量……279
8.2.1 结构体类型的声明……279
8.2.2 结构体类型变量的定义……280
8.2.3 结构体的嵌套……282
8.3 结构体类型变量的引用和初始化……283
8.4 结构体数组……285
8.5 结构体指针……288
8.5.1 指向结构体类型变量的指针……288
8.5.2 指向结构体数组的指针……289
8.6 结构体与函数……292
8.7 结构体综合应用实例……295
8.8 共用体……305
8.8.1 问题的引出……305
8.8.2 声明共用体类型和定义共用体类型的变量……305
8.8.3 共用体成员的引用……307
8.9 枚举类型……308
8.10 typedef……311
*8.11 链表……312
8.11.1 问题的引出……312
8.11.2 链表的定义和特点……313
8.11.3 链表的创建……313
8.11.4 链表的删除操作……318
8.11.5 链表的插入操作……320
课后习题 8……324
第 9 章 文件……333
9.1 文件概述……333
9.1.1 什么是文件……333
9.1.2 文件名……334
9.1.3 文件的分类……334
9.1.4 文件缓冲区……335
9.1.5 文件指针……336
9.2 文件的打开与关闭……336
9.2.1 用 fopen 函数打开文件……336
9.2.2 用 fclose 函数关闭文件……338
9.3 文件的读写……339
9.3.1 读/写字符函数……339
9.3.2 读/写字符串函数……341
9.3.3 格式化读/写函数……343
9.3.4 读/写数据块函数……345
9.4 文件的定位……353
9.4.1 移动文件指针……353
9.4.2 获取文件读写位置……355
9.5 出错检测……356
课后习题 9……357
第 10 章 综合应用案例——学生学籍管理系统……360
10.1 需求分析……360
10.2 总体设计……361
10.2.1 系统总体设计……361
10.2.2 数据结构……361
10.3 详细设计……362
10.3.1 系统包含的函数……362

10.3.2　各个功能模块的软件功能 …… 362
10.3.3　各个功能模块的程序流程图和算法描述 …… 363
10.4　编码实现 …… 368
10.5　运行结果 …… 376
课后习题 10 …… 379
附录 A　C 关键字 …… 382
附录 B　C 运算符的优先级和结合性 …… 383
附录 C　ASCII 码字符表 …… 384
附录 D　常用的 ANSI C 标准库函数 …… 388
参考文献 …… 395

第1章　C语言概述

计算机是20世纪最重要的发明之一，它深刻地影响了我们的生活，改变了我们的社会。例如，它控制着我们的交通、通信及其他系统。那么，什么是计算机，计算机又是如何工作的呢？

计算机(Computer)是基于一串指令进行数据操作的机器，而这串指令被称为程序(Program)。比起人的大脑，计算机具有更快的信息处理速度和数值计算能力。而安装在计算机内的程序告诉计算机什么时候取信息，如何处理信息并实施计算，产生什么样的结果。程序指令由中央处理器(Central Processing Unit，CPU)来执行，CPU通常也被称为微处理器(Microprocessor)。目前常见的台式计算机(Desktop Computer)、笔记本式计算机(Notebook Computer)或平板计算机(Tablet Computer)，都是通用计算机，用来为不同的任务运行不同的程序。例如，运行在计算机上的游戏程序能够处理用户的输入并根据程序指令触发动作；运行在计算机上的Internet网页浏览器可以根据用户输入的网页地址，经一系列的处理后与运行在该网页地址上的主机的另一个程序进行通信而获得信息，最终将结果显示在网页浏览器上。

嵌入式系统(Embedded System)，是为了完成一个或多个特殊功能的专用计算机系统。例如，一个机器人通常含有一个单板嵌入式计算机系统。根据机器人程序，机器人能够做出智能决策并根据一些外部传感信息(如位置、受力、视觉等)采取特定的动作。一台计算机或一个机器人仅服从写入其内的程序命令，计算机或机器人之所以表现得如此聪明，是因为有聪明的人编写了聪明的程序。

C语言是用来编写能够跟硬件交互的亿万嵌入式系统程序的首选。C和它的扩展C++也是编写游戏、网页浏览器、文字处理和运行在计算机上的大部分程序的首选语言。本教材将教你如何编写程序，让计算机来帮你完成想做的事。

1.1　计算机编程语言

计算机编程语言可以分为低级语言和高级语言。其中低级语言包括机器语言和汇编语言。高级语言比低级语言更容易使用，也更容易移植。下面将介绍各种语言的工作原理。

1.1.1　机器语言

机器语言(Machine Language)是由二进制编码指令构成的唯一可被计算机直接识别的计算机语言。每种处理器都有自己专用的机器指令集合。处理器的设计者用不同的二进制

编码表示不同的机器指令，每条机器指令只能完成非常低级的处理任务。例如，下列程序是用某种处理器的机器语言编写的，该程序的功能是在屏幕上显示字符串“Hello”。

```
11100000  01001000 ; 输出字符 H
11100000  01100101 ; 输出字符 e
11100000  01101100 ; 输出字符 l
11100000  01101100 ; 输出字符 l
11100000  01101111 ; 输出字符 o
00000000
```

其中，二进制串 11100000 表示屏幕字符输出指令，该指令带一个操作数，表示输出字符的 ASCII 码。二进制串 00000000 表示停止指令，该指令没有操作数。由此可见，一条机器指令是由一个操作码(Operator)和 0 到多个操作数(Operand)构成的。其中，操作码规定了指令的功能，操作数指明了操作的对象。

尽管用机器语言编写的程序能够被计算机直接理解和执行，但用二进制编码进行编程不仅效率极低，而且所编写的程序含义不直观，难以理解和记忆，错误也难以查找。因此，使用机器语言根本无法高效地编写出高质量的复杂程序，现在已经没有人再用机器语言编写程序了。

1.1.2　汇编语言

为了减轻使用机器语言编程的痛苦，人们进行了一些改进，即用一些简洁的英文字母、符号串来替代一个特定指令的二进制编码。例如，用 ADD 代表加法，MOV 代表数据传递等。这样，人们很容易读懂并理解程序的含义，查找和纠正错误也变得更加方便，这种编程语言就是汇编语言(Assembly Language)。汇编语言为每条机器指令分配了一个助记符号，人们可以使用这些助记符号代替二进制串来编写程序。例如，在屏幕上输出“Hello”的程序，用某种机器的汇编语言可以写为：

```
Write H
Write e
Write l
Write l
Write o
Stop
```

在上述程序中，用 Write 代替了二进制的操作码，用字符代替了其对应的二进制的 ASCII 码。可见，汇编语言是用助记符号表示机器指令的计算机语言。汇编语言指令与机器指令基本上具有一一对应关系。采用汇编语言编程，程序的可理解性、编写效率以及质量都有所提高，但是计算机不能直接理解和执行汇编语言程序，必须将其翻译成机器语言，程序才能被机器理解和执行，这个翻译过程称为“汇编”。目前汇编语言主要用于资源受限的嵌入式系统和对实时性要求非常高的系统编程中。尽管汇编语言程序非常简练、有效，但

试读结束，需要全本PDF请购买 www.ertongbook.com

它是机器依赖的，而且非常繁琐、可读性差、难以修改。

1.1.3　高级语言

从最初与计算机交流的痛苦经历中，人们意识到应该设计一种既接近于数学语言或人的自然语言又不依赖于计算机硬件，编出的程序能在所有机器上通用的语言，这样的语言被称为高级语言(High-Level Language)。高级语言与计算机类型无关，在某一机器上完成的程序可以在另一台机器上运行，而且它们易读、易写、易维护。例如，下面是一条用 C 语言书写的在屏幕上输出“Hello”的语句：

```
printf("Hello");
```

由上例可以看出，与汇编语言相比，高级语言将许多相关的机器指令合成为单条指令，因此，高级语言指令能完成较复杂的任务。由于屏蔽了与硬件操作有关的细节，编程人员不需要掌握太多的计算机硬件的专业知识，可以集中精力于确定问题求解的算法上。编码相对简单，所编写的程序也更加容易理解和维护。

工程人员和科学研究人员通常使用的高级语言有 FORTRAN、C、C++、Java 和 C#。表 1-1 给出了各种语言的应用领域及其发明者。

表 1-1　高级语言

语言	应用领域	发明者
FORTRAN	数值和科学计算编程	John W. Backus
C	系统编程和嵌入式系统	Dennis M. Ritchie
C++	面向对象系统编程	Bjarne Stroustrup
Java	网络与系统编程	James Gosling
C#	网络与系统编程	Anders Hejlsberg

FORTRAN 是首个通用高级语言，由 IBM 的 John W. Backus 于 20 世纪 50 年代后期开发。它的主要目的是进行公式转换计算。尽管很多人认为它相当落伍，但它至今还是一种有效进行高性能数值计算的主要语言。

C 语言是 20 世纪 70 年代为了写 UNIX 操作系统以及相关应用程序由 AT&T 公司的 Dennis M. Ritchie 开发的。之所以命名为 C，是因为它从早期 B 语言的基础上发展而来，B 语言是 BCPL 语言的简化版。Brian Kernighan 和 Dennis M. Ritchie 在 1978 年出版的《C 编程语言》在相当长的时间里都是 C 语言的默认标准。1983 年美国国家标准协会(ANSI)成立了 X3J11 委员会，为 C 制定了标准规范。该标准在 1989 年被认可，通常称之为 ANSI C 或 C89 标准。1990 年，ANSI C 标准做了一些小的修改后，被国际标准化组织(ISO)吸纳为国际标准，有时这个版本被称为 C90。C89 和 C90 指的是同一语言版本。最先的 C 标准起始于 1991 年，在 1999 年完成，2000 年被认可，该标准称为 C99 标准。

C 被广泛用来进行系统编程，如编写操作系统、应用程序、编译器、解释器以及函数库。由于 C 语言能够像汇编语言一样精确控制机器，所以 C 语言又通常被认为是一种中级语言(Mid-Level Language)。C 语言也是嵌入式系统开发的首选语言，同时广泛用于开发终端用户的应用程序。

1979 年贝尔实验室的 Bjarne Stroustrup 为了增加 C 的面向对象编程能力开发了 C++。该语言最初命名为具有类功能的 C(C With Classes)。1983 年，它才被命名为 C++(++是 C 和 C++中的增量操作)。面向对象编程是一种用对象及其交互接口来设计应用程序的编程方法，它适用于大规模的软件工程。C++比 C 复杂得多，如果没有较好的 C 基础，要掌握 C++中面向对象特性是不太可能的。C++一直以来是 C 的一个超集，直到最新标准 C99 出现，C++不再是 C 的超集。C++不支持 C99 中的可变长数组等特性。然而在 C++的下一个标准中很有可能包含 C99 的这些新特性。

就像 C++一样，很多所谓的现代计算机语言，如 Sun 公司开发的 Java、微软的 C#等都是基于 C 开发的，它们是面向对象的编程语言。Java 还能够用来方便地开发具有图形用户界面的网络计算应用。

计算机的 CPU 唯一能够执行的代码是机器码。编译器(Compiler)就是一个用来把高级语言程序翻译成低级语言或汇编语言或机器码的计算机程序。源文本称为源代码(Source Code)或源文件(Source File)，而通过编译器输出的称为目标码(Object Code)或目标文件(Object File)。一般情况下，编译器将源代码直接翻译成机器码。如一个 C 程序，编译器能够自动将一些源代码格式的头文件(Header File)包含到另一个源文件中。链接器(Linker)是一种能够将一个或多个由编译器生成的目标文件装配成单个可执行程序的应用程序。链接器还可以接纳库(Library)，库是一组编译后的目标文件的集合。图 1-1 给出了 Windows 下用 C 语言开发一个可执行程序的处理过程。首先利用 Windows 中的文本编辑器，如 Notepad，编辑好文件名为 hello.c 的 C 源代码。该源代码用编译器程序 compile.exe 来编译，源文件中的一些头文件在编译过程中会自动包含进来。编译器将源文件编译成一个目标文件 hello.obj。然后，链接器程序 link.exe 将目标代码和一些库链接生成一个可执行程序 hello.exe 并存放在外存上。如果不运行该文件，什么都不会发生。该程序可以通过在命令行界面中输入该程序名运行，或者通过 GUI 启动它，将它所需的实时运行库装载到贮存中。程序 hello.exe 运行过程如图 1-2 所示，根据程序设计的要求，可能要求用户输入一些数据，然后才输出一些结果。

图 1-1　利用编译器和链接器为 C 程序产生一个可执行程序的过程

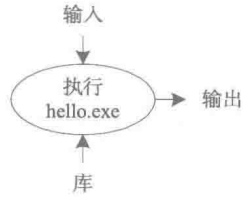


图 1-2　执行一个可执行程序的过程

由于几乎任何计算机平台中都有 C 编译器，所以一些高级语言首先可能被翻译成 C 语言，然后通过 C 编译器编译 C 代码。C++早期实现就是利用这种模式，但现在 C++程序已经可以跳过中间代码过程直接编译成机器码。

有些高级语言，其源代码被编译成一种称作字节码(Bytecode)的中间代码，这种字节码是可以在程序虚拟机(Virtual Machine)上运行的指令集，Java 就是利用这种运行模式。为了提高执行效率，如今的 Java 也可以直接编译成机器码。

用高级语言编写的程序也可以通过一个解释器直接读取并执行，不需要编译和链接的过程。

1.2　第一个 C 程序

我们从一个简单的程序开始，该程序运行后将在屏幕上输出如下信息：

```
Hello World
```

【程序 1-1】第一个 C 程序如下所示，该程序运行后的输出结果如上。

```
/* 文件：hello.c
程序功能：在屏幕上打印输出信息"Hello World" */
#include <stdio.h>
int main()
{
    printf("Hello World\n");
    return 0;
}
```

在这里，源代码保存在一个名为 hello.c 的文件中，然后由编译器来处理。编译器能处理 C 源代码并生成一个可执行程序。可执行程序运行时，就可以产生基于源代码的期望结果。C 程序可以通过一个集成开发环境来编辑执行，这样用户就可以在同一个界面上完成程序的编辑运行。下一节将介绍如何利用集成开发环境来编辑运行 C 程序。

下面详细介绍程序 1-1 中每一行的内容和含义。

(1) 用/*符号起始并用*/符号结尾的部分是注释(Comment)。当编译器处理注释时，这些注释内容将被忽略不产生任何动作。注释用来注解程序使得代码易读易懂。程序 1-1 中的头两行就是注释：

```
/* 文件: hello.c
程序功能：在屏幕上打印输出信息"Hello World" */
```

它们注解该程序文件名称为 hello.c，程序的功能是在屏幕上打印输出信息“Hello World”。C 程序文件通常都是以.c(称为扩展名)结尾。

注释内容可以跨越多行。但是，两个注释定界符号不允许嵌套出现。例如：

```
/*注释开始/*嵌套在一起的注释是不正确的*/注释结束*/
```

在 C99 标准中加入了有符号//的注释方式。这种方式下，将符号//后出现的文本视为注释内容，该注释到当前行末为止。例如：

```
printf("Hello World\n");          //这部分是注释
```

(2) 程序 1-1 中的行：

```
#include <stdio.h>
```

是预处理命令。用#开始的行称为预处理命令，这些预处理命令通常被 C 编译器的预处理程序处理。include 命令通知编译器要在该程序中包含文件 stdio.h 的所有内容。通过#include 预处理命令而包含的文件被称为头文件。头文件 stdio.h 包含了与标准输入输出库相关的函数声明等信息。头文件通常以.h 作为其扩展名。如 stdio.h 是标准 C 头文件，通常都是编译器自带的，用户不需要关心这些标准头文件所包含的内容。它们的实现与编译器相关，不同的编译器提供的标准头文件的内容会有所不同。每一个编译器都会有各自的头文件集。程序 1-1 中之所以要包含头文件 stdio.h，是因为后面要用到标准输出函数 printf()。

(3) 函数(Function)是 C 程序中基本的可执行模块。一个 C 程序含有一个或多个函数，其中必须包含的函数是 main()，它将返回一个类型为 int 的值，关于 int 类型的详述请参见第 2 章。下面这一行：

```
int main()
```

是每个 C 程序必须包含的部分。符号 main 后面的圆括号表示它是一个函数。C 程序都是从函数 main()开始执行的。

一组类似的、由 C 语言预先定义、具有特定含义的标识符被称为关键字。附录 A 给出了 C 语言所有完整的关键字列表，符号 int 就是其中的一个。int 用来声明函数 main()的返回值类型为整型，这意味着 main()函数返回值的类型是整数。返回到哪里呢？返回给操作系统。

如果浏览老版本的 C 代码，你将发现程序常常以：

```
main()
```

这种形式开始。C90 标准勉强允许这种形式，但是 C99 标准不允许。

你还将看到另一种形式：

```
void main()
```

有些编译器允许这种形式，但是还没有任何标准考虑接受它。因而，编译器不必接受这种形式，并且许多编译器也不这样做。如果坚持使用标准形式，当把程序从一个编译器移到另一个编译器时也不会有问题。

一个函数通常包含有很多语句。函数中的所有语句用一对大括号表示起止。左大括号{是函数体的开始，与之匹配的右大括号}表示函数定义的结束。这对大括号及其内的所有语句被称为程序块(Block)。

程序 1-1 中的函数 main()包含两条语句。

```
printf("Hello World\n");
```

调用了函数 printf()来输出“Hello World”，函数 printf()是输入/输出库中的一个标准函数。在本例中函数 printf()的参数是一个字符串，printf()函数可以有多个参数。在这里只有一个字符串数据类型的参数传入。当一个字符串被传给函数 printf()时，通常将出现在程序中的字符串完整地显示在计算机屏幕上。

引号中有字符\n，但并没有输出它们，发生了什么事情呢？\n 字符的意思是开始新的一行。\n 组合代表一个称为“换行符”的字符，它意味着“在下一行的最左边开始新的一行”。换句话说，打印换行字符的效果和在普通键盘上按下“Enter”键一样。换行符是转义字符(Escape Sequence)的一个例子。转义字符通常用于代表难于表达的或是无法键入的字符。完整的转义字符列表请参考第 2 章。

(4) 程序中的每一条语句必须用分号结尾。为了软件的可读性和可维护性，在函数中的语句要进行恰当的缩进(Indent)，缩进时采用固定长度的空格，建议采用如程序 1-1 的四个空格的缩进法。像 int 一样，符号 return 也是 C 语言的关键字。

```
return 0;
```

出现在函数 main()的最后。一旦该语句被执行，表示程序成功结束并返回值 0。根据程序被执行时的方式，该程序返回值将会传递给执行环境。对于 main()函数来说，如果漏掉了 return 语句，则大多数编译器将对你的疏忽提出警告，但仍将编译该程序。此时，你可以暂时把 main()中的 return 语句看作是保持逻辑连贯性所需的内容。但对于某些操作系统(包括 DOS 和 UNIX)而言，它有实际的用途。

1.3　C 程序的上机步骤

开发 C 程序可以在一个集成环境中进行所有的工作，Visual C++ 6.0 简称 VC6，是常用的开发 C 程序的工具之一，其界面如图 1-3 所示。在 VC6 中，应用程序向导 AppWizard 可以帮助程序员创建一些常用的应用程序类型框架，此处只介绍 Win32 控制台应用程序(Win32 Console Application)的创建、编译和执行。

图 1-3　VC6 的界面

1.3.1　单文件的 C 程序的上机步骤

下面以程序 1-1 为例，介绍使用 VC6 开发控制台应用程序的步骤，具体操作过程如下。

(1) 在 VC6 环境下，选择菜单“文件”→“新建”命令，弹出如图 1-4 所示的窗口。在该窗口中，选择“工程”选项卡中的“Win32 Console Application”选项，在“工程名称”文本框中输入项目名称，如 prj，然后在“位置”文本框中输入文件的存放位置，最后单击“确定”按钮。

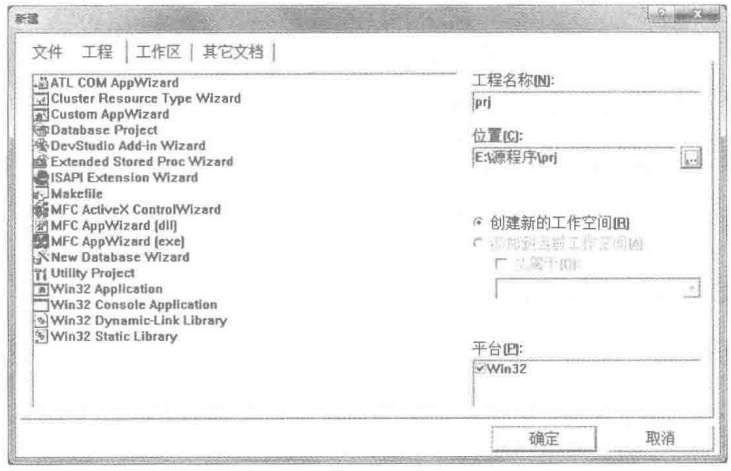


图 1-4　VC6 新建工程窗口

(2) 在弹出的如图 1-5 所示的询问项目类型的窗口中，选中“一个空工程”单选按钮，单击“完成”按钮。

(3) 系统将弹出如图 1-6 所示的窗口，即新建工程信息，单击“确定”按钮。

(4) 至此就已经完成了一个项目的框架。再次选择菜单“文件”→“新建”命令，选择“文件”选项卡中的“C++ Source File”选项，在“文件名”文本框中输入程序文件名

试读结束，需要全本PDF请购买 www.ertongbook.com