普通高等教育“十三五”规划教材

C语言程序设计实用教程

（第2版）

主编　张桂珠　杨开莜
方　伟　韩亦强

CodeBlocks + Visual Studio 2017 + Visual C++ 6.0

北京邮电大学出版社
www.buptpress.com

普通高等教育“十三五”规划教材

CodeBlocks＋Visual Studio 2017＋Visual C++ 6.0

C 语言程序设计实用教程

（第 2 版）

主编　张桂珠　杨开菠　方　伟　韩亦强

北京邮电大学出版社
www.buptpress.com

内 容 简 介

本书详细介绍了C语言的语法知识和使用，结合实际工程应用中的大量实例，讲解了如何使用C语言解决实际问题的理论、方法和过程，全书内容也兼顾到全国计算机二级等级考试C语言的大纲要求。针对初学者和自学者的特点，在讲解过程中，力求语言简洁、抓住重点、精选例子。结合作者多年的教学经验和项目开发经验组织教材，做到深入浅出、难点分散，力争在解决问题的过程中使学习者能融会贯通地掌握C语言。在C语言程序的上机环境安装和使用上，分别介绍了全国计算机二级等级考试的机考环境 Visual C++ 6.0、最新的 Visual Studio 2017 集成开发环境以及 CCF CSP 的机考环境 CodeBlocks。

本书可作为学习C语言程序设计课程的教材，也可作为全国计算机二级等级考试C语言的学习主导教材，还可作为C语言的自学者或短训班人员的学习教材。为方便读者学习，作者还编写了与本书配套的《C语言程序设计实用教程习题解答与实验(第2版)》。本书还配有一套教学电子资源，包括教学课件、例子源代码、习题解答源代码和实验答案源代码。

图书在版编目(CIP)数据

C语言程序设计实用教程/张桂珠，等主编. --2版. --北京：北京邮电大学出版社，2018.8

ISBN 978-7-5635-5573-4

Ⅰ. ①C… Ⅱ. ①张… Ⅲ. ①C语言—程序设计—教材 Ⅳ. ①TP312.8

中国版本图书馆 CIP 数据核字(2018)第 183861 号

书　　名：C语言程序设计实用教程(第2版)
著作责任者：张桂珠　杨开莅　方　伟　韩亦强　主编
责 任 编 辑：张珊珊
出 版 发 行：北京邮电大学出版社
社　　址：北京市海淀区西土城路10号(邮编：100876)
发　行　部：电话：010-62282185　传真：010-62283578
E-mail：publish@bupt.edu.cn
经　　销：各地新华书店
印　　刷：
开　　本：787 mm×1 092 mm　1/16
印　　张：15
字　　数：384千字
版　　次：2012年8月第1版　2018年8月第2版　2018年8月第1次印刷

ISBN 978-7-5635-5573-4　　定　价：36.00元

· 如有印装质量问题，请与北京邮电大学出版社发行部联系 ·

此为试读，需要完整PDF请访问：www.ertongbook.com

前　　言

在程序设计语言中，C语言是国内外编程人员使用最广泛的语言。由于C语言本身功能丰富、使用灵活、可移植性好，既具有高级语言的优点，又具有低级语言的特点，既可用于编写系统软件如操作系统、编译程序、设备驱动程序，又可用于编写应用软件，在嵌入式系统领域，C语言也得到了广泛使用。因此C语言程序设计是计算机应用人员应掌握的基本功。

《C语言程序设计实用教程(第2版)》和与之配套的《C语言程序设计实用教程习题解答与实验(第2版)》，是作者结合多年的教学实践和项目实践编写的。全书内容也兼顾到全国计算机二级等级考试C语言的大纲要求。在讲解过程中力求语言简洁、抓住重点、精选例子，组织教材做到深入浅出、难点分散，力争在解决问题的过程中使学习者能融会贯通地掌握C语言。

在C语言程序的上机环境安装和使用上，首先介绍软件CodeBlocks 16.01，它是一款开源、免费、跨平台的C/C++集成开发环境，在Windows、Linux等OS平台上都能便捷安装，且界面友好，调试功能强大，运行高效，它也是中国计算机学会组织的计算机软件能力认证CSP的C/C++机考环境。接着介绍软件Visual C++ 6.0，它是全国计算机二级等级考试的C语言机考环境，但其版本已经很旧，且在Windows 7以上版本的操作系统上，需要安装较高版本的Visual Studio，C程序才能正常执行。最后介绍了目前最新版本Visual Studio 2017的下载、安装和使用。对于初学者而言，最好选择CodeBlocks 16.01下载和使用比较方便。

全书共有9章，内容概要如下：

第1章，程序设计和C语言概述。介绍程序设计相关的基本概念，C语言的特点，简单C程序例子；介绍CodeBlocks 16.01、Visual Studio 2017和VC++ 6.0三种环境软件的下载、安装和使用，结合开发环境介绍如何输入、编译、连接和运行C程序的过程。

第2章，顺序结构程序设计。从计算机系统角度介绍数据的机内表示和存储，介绍C程序的组成结构，基本数据类型、变量、常量、指针变量、表达式、赋值语句，以及基本输入和输出语句等。最后讨论了顺序结构程序设计应用实例。

第3章，选择结构程序设计。介绍算法的基本知识和结构化程序设计的方法，在理解程序的三种基本控制结构基础上，详细介绍与选择结构相关的程序设计，包括：关系表达式、逻辑表达式、if语句和switch语句。最后讨论了选择结构的程序设计实例。

第4章，循环结构程序设计。介绍三种循环语句while语句、do-while语句和for语句的使用格式，以及如何使用这些语句表达循环结构。最后讨论了循环结构的程序设计实例。

第5章，函数。介绍了函数的定义和调用，函数参数的传递方式，与函数相关的指针应用，给出了利用函数进行模块化程序设计的大量实例。最后讨论了变量的作用域、C程序的多文件结构、编译预处理常用命令、系统的库函数及其应用实例。

第6章，数组、字符串与动态内存分配。介绍了一维数组和多维数组的声明和使用，介绍了通过下标变量和指针访问一维数组元素和二维数组元素的方法，讨论了数组应用的一组常

用算法。介绍了存放字符串的字符型数组声明、访问和输入/输出,字符串处理库函数及其应用实例。最后介绍了动态内存的申请或释放。

第 7 章,用户自定义类型。用户自定义数据类型包括:结构体、联合体、枚举型,重点介绍了每种数据类型的定义和应用实例。介绍了 typedef 的定义和使用。

第 8 章,位操作程序设计。介绍了二进制位运算,包括:位与、或、异或、取反、左移和位右移。介绍了使用结构体表示二进位的数据结构——位域。讨论了位操作程序设计的综合举例。

第 9 章,文件的输入和输出处理。介绍了文件的命名、文件的打开与关闭、文件的读取与写入。详细介绍了与文件读写操作相关的一组库函数,并结合应用实例给出了文件的顺序读写和随机读写的方法。

第 10 章,调试程序。分别介绍了在 CodeBlocks 16.01 和 Visual C++ 6.0 两种软件环境下,如何调试运行 C 程序。

本书还配有一套完整的电子教学资源,包括教学课件、例子源代码、习题解答源程序和实验解答源程序等。读者可在北京邮电大学出版社网站自行下载。

全书由张桂珠、杨开莅、方伟、韩亦强主编,参编人员有徐华、韩振、蒋敏、李婷、姚健等。本书在编写过程中,得到了江南大学同仁们的协助与支持,在此一并致谢。

感谢读者选择使用本书,欢迎您对本书提出批评和修改建议,我们将不胜感激,并在再版时予以考虑。作者的邮箱地址如下:zhangguizhu@163.com。

作　者

目　　录

第 1 章　程序设计和 C 语言概述 …… 1

1.1　程序设计基本概念 …… 1
1.1.1　什么叫程序设计 …… 1
1.1.2　什么叫程序设计语言 …… 2
1.2　C 语言的特点和应用 …… 3
1.2.1　C 语言的特点 …… 3
1.2.2　C 与 C＋＋、Java、C♯ …… 4
1.3　简单 C 语言程序入门 …… 4
1.4　C 语言程序运行环境的安装和使用 …… 6
1.4.1　在 CodeBlocks 集成开发环境下执行 C 语言程序 …… 6
1.4.2　在 Visual C＋＋ 6.0 集成开发环境下执行 C 语言程序 …… 11
1.4.3　在 Visual Studio 2017 集成开发环境下执行 C 语言程序 …… 14
1.5　本章小结 …… 18
习题 …… 18

第 2 章　顺序结构程序设计 …… 20

2.1　C 语言程序结构 …… 20
2.1.1　字符集 …… 21
2.1.2　词法记号 …… 21
2.2　数据的机内表示和存储 …… 22
2.2.1　二进制、八进制、十六进制 …… 22
2.2.2　原码和反码 …… 24
2.2.3　补码——有符号整数的机内表示形式 …… 24
2.2.4　浮点数的机内表示形式 …… 25
2.3　数据类型 …… 26
2.4　常量 …… 27
2.4.1　整型常量 …… 27
2.4.2　浮点型常量 …… 27
2.4.3　字符常量 …… 27
2.4.4　字符串常量 …… 28
2.4.5　逻辑型常量 …… 29
2.5　变量 …… 29

2.5.1 变量的声明 …… 29
2.5.2 变量的初始化 …… 33
2.6 操作数存储空间的大小 sizeof …… 33
2.7 常变量与符号常量 …… 34
2.8 算术运算符与算术表达式 …… 35
2.9 赋值运算符与赋值表达式 …… 38
2.10 逗号运算符与逗号表达式 …… 40
2.11 运算符的优先级与结合性 …… 40
2.12 混合运算时数据类型的转换 …… 41
2.13 语句和块 …… 43
2.14 指针与指针变量 …… 43
2.14.1 指针的概念 …… 43
2.14.2 取地址运算符 …… 43
2.14.3 指针变量的声明与初始化 …… 44
2.14.4 取内容运算符 …… 45
2.14.5 指针变量的赋值与使用规则 …… 46
2.15 数据的输入与输出 …… 47
2.15.1 用 printf 函数输出数据 …… 47
2.15.2 用 scanf 函数输入数据 …… 50
2.15.3 用 getchar 和 putchar 函数输入/输出单个字符 …… 54
2.16 顺序结构程序设计综合举例 …… 55
2.17 本章小结 …… 58
习题 …… 58

第 3 章 选择结构程序设计 …… 62

3.1 算法的基本概念和表示方法 …… 62
3.1.1 算法的基本概念 …… 62
3.1.2 算法的表示 …… 62
3.1.3 结构化程序设计 …… 64
3.2 关系运算符与关系表达式 …… 65
3.3 逻辑运算符与逻辑表达式 …… 66
3.4 用 if 语句实现选择结构 …… 68
3.4.1 实现单分支的 if 语句 …… 68
3.4.2 实现双分支的 if 语句 …… 68
3.4.3 实现多分支的 if 语句嵌套 …… 70
3.4.4 条件运算符与条件表达式 …… 72
3.5 用 switch 语句实现选择结构 …… 72
3.6 选择结构程序设计综合举例 …… 75
3.7 本章小结 …… 77
习题 …… 78

第 4 章　循环结构程序设计 …… 80

4.1　用 while 语句实现循环 …… 80
4.2　用 do-while 语句实现循环 …… 82
4.3　用 for 语句实现循环 …… 84
4.4　循环的嵌套 …… 86
4.5　跳转语句：break 语句、continue 语句和 goto 语句 …… 89
4.5.1　continue 语句 …… 89
4.5.2　break 语句 …… 89
4.5.3　无条件转移 goto 语句 …… 90
4.6　循环结构程序设计综合举例 …… 91
4.7　本章小结 …… 98
习题 …… 99

第 5 章　函数 …… 103

5.1　函数的定义与调用 …… 103
5.1.1　函数的定义 …… 103
5.1.2　函数的调用 …… 104
5.2　函数的参数传递 …… 108
5.2.1　按值传递 …… 108
5.2.2　按地址传递 …… 109
5.2.3　按引用传递 …… 111
5.3　函数嵌套与递归 …… 112
5.4　返回指针类型的函数 …… 115
5.5　指向函数的指针 …… 115
5.6　变量的作用域与可见性 …… 117
5.6.1　变量的作用域 …… 117
5.6.2　变量的可见性 …… 119
5.7　变量的存储类型和生存期 …… 120
5.7.1　变量的生存期 …… 120
5.7.2　变量的存储类型 …… 120
5.8　C 程序的多文件结构 …… 121
5.9　编译预处理指令 …… 123
5.9.1　宏定义指令 #define …… 123
5.9.2　文件包含指令 #include …… 124
5.10　C 系统函数 …… 124
5.11　本章小结 …… 126
习题 …… 126

第 6 章　数组、字符串与动态内存分配 …… 130

6.1　数组概念 …… 130
6.2　一维数组 …… 130
6.2.1　一维数组的声明 …… 130
6.2.2　一维数组的初始化 …… 131
6.2.3　一维数组元素的表示方法 …… 132
6.2.4　用指针访问一维数组 …… 133
6.2.5　函数参数为访问一维数组的指针 …… 135
6.2.6　一维数组综合程序设计举例 …… 136
6.3　多维数组 …… 140
6.3.1　二维数组的声明 …… 140
6.3.2　二维数组的初始化 …… 140
6.3.3　二维数组元素的表示方法 …… 141
6.3.4　声明二级指针 …… 144
6.3.5　用指针访问二维数组 …… 144
6.3.6　函数参数为访问二维数组的指针 …… 147
6.3.7　二维数组综合程序设计举例 …… 148
6.4　字符数组 …… 149
6.4.1　字符数组的定义 …… 149
6.4.2　字符数组的初始化 …… 149
6.4.3　字符数组元素的表示方法 …… 150
6.4.4　字符数组的输入与输出 …… 151
6.4.5　使用字符串函数处理字符串 …… 153
6.4.6　用指针访问字符串 …… 157
6.4.7　字符串的综合程序设计举例 …… 159
6.5　指针数组与 main 函数的参数 …… 161
6.5.1　指针数组的定义 …… 161
6.5.2　main 函数的参数 …… 163
6.6　内存的动态分配与释放 …… 165
6.6.1　void 指针类型 …… 165
6.6.2　动态内存的申请 …… 165
6.6.3　动态内存的释放 …… 166
6.7　本章小结 …… 167
习题 …… 167

第 7 章　用户自定义类型 …… 171

7.1　结构体类型 …… 171
7.1.1　定义结构体类型 …… 171
7.1.2　定义结构体变量 …… 172

7.1.3　结构体变量的使用 …… 173
7.1.4　结构体变量的初始化 …… 174
7.2　结构体数组的使用 …… 175
7.3　结构体指针变量的使用 …… 177
7.4　用 typedef 声明新类型名 …… 179
7.5　单向链表的建立与基本操作 …… 181
7.5.1　什么叫链表? …… 181
7.5.2　如何定义结点的数据类型 …… 181
7.5.3　创建动态链表 …… 183
7.6　联合体类型 …… 185
7.6.1　定义联合体类型 …… 185
7.6.2　定义联合体变量 …… 186
7.6.3　联合体变量的使用 …… 187
7.7　枚举类型 …… 190
7.8　本章小结 …… 192
习题 …… 192

第 8 章　位操作程序设计 …… 195

8.1　位运算符 …… 195
8.2　位域 …… 197
8.3　位操作程序设计综合举例 …… 199
8.4　本章小结 …… 200
习题 …… 200

第 9 章　文件的输入和输出处理 …… 202

9.1　文件的基本概念 …… 202
9.1.1　文件的分类 …… 202
9.1.2　文件名 …… 203
9.1.3　文件类型指针 …… 203
9.2　文件的打开与关闭 …… 203
9.2.1　文件的打开 …… 203
9.2.2　文件的关闭 …… 205
9.3　文件的顺序读写 …… 205
9.3.1　向文件读写字符 …… 205
9.3.2　向文件读写字符串 …… 208
9.3.3　以二进制方式向文件读写数据块 …… 209
9.3.4　向文件格式化读写数据 …… 212
9.4　文件的随机读写 …… 214
9.4.1　文件的定位 …… 214
9.4.2　随机的读写 …… 215

此为试读,需要完整PDF请访问: www.ertongbook.com

9.5 本章小结 …… 218
习题 …… 218

第 10 章 调试程序 …… 220

10.1 CodeBlocks 环境下如何调试程序 …… 220
10.2 Visual C++ 6.0 环境下如何调试程序 …… 222

附录 A 常用字符与 ASCII 代码对照表 …… 223

附录 B C 语言常用的库函数 …… 225

参考文献 …… 230

第 1 章　程序设计和 C 语言概述

本章首先介绍了程序设计相关的基本概念，如程序、程序设计语言、程序设计的开发过程、语言的分类（包括机器语言、汇编语言和高级语言及其翻译程序）。接着介绍了 C 语言的特点和应用场合，详细介绍了几个简单的 C 语言程序例子，介绍了集成开发环境 CodeBlocks 16.01 和Visual C++ 6.0、Visual Studio 2017 的安装过程，以及它们如何输入、编译、连接和运行 C 语言的程序，如何纠正程序中的语法错误等。本章为读者了解和使用 C 语言编程、进一步学习后面的章节打下了很好的基础。

1.1　程序设计基本概念

1.1.1　什么叫程序设计

程序是计算机能执行的一组指令，用于完成特定的任务。程序设计的任务就是如何将一个问题转换成计算机能自动执行的一组指令。程序设计的开发过程一般由四个步骤组成：

（1）分析问题：分析要得到哪些输出结果，有哪些输入，以及问题的处理过程。

（2）设计算法：算法是对一个问题求解步骤的一种描述，是求解问题的方法。可用自然语言、伪代码或流程图表达算法（第 3 章将详细讨论算法）。

（3）编制程序：将流程图或伪代码转换成程序设计语言表达的程序。

（4）测试程序：通过在计算机运行程序，输入各种类型的数据，观察输出结果是否达到预期要求，不断修正程序，使程序的运行结果正确。

例如：求任意三个整型数 a，b，c 的最大值。

此问题的求解过程，有三个步骤：

（a）输入三个整数送给 a，b，c；

（b）求 a、b 的较大者 larger，再用 larger 与 c 比较，求出的较大者就是最大者 max；

（c）输出最大者 max。

程序设计是对一个给定问题的求解过程，它会分析问题有哪些输入数据，有哪些输出结果，分析由输入数据如何映射为输出结果的处理过程，如图 1-1 所示。处理过程往往由 N 步骤组成，有简单的处理过程，也有复杂的处理过程（如天气预报预测），由要处理问题的规模大小决定。

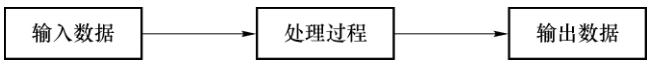

图 1-1　程序设计的求解过程

1.1.2 什么叫程序设计语言

程序设计语言是计算机能够识别和执行的语言,它是由一套语法规则和语义组成的系统,用于对要解决的问题进行描述。程序设计语言按级别分为机器语言、汇编语言和高级语言。语言的发展经历了由低级向高级的发展过程。图 1-2 展示了两数加法操作的例子在不同级别语言上的代码表示。

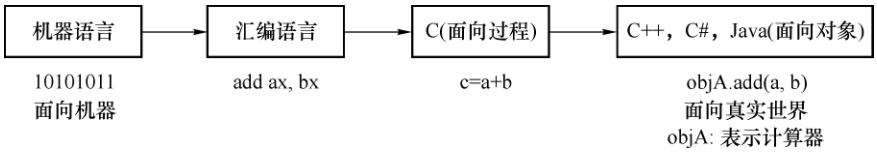

图 1-2 程序设计语言的发展过程

1. 机器语言

机器语言是由二进制序列组成的机器指令集合。用机器语言编制的程序可被计算机直接识别和执行。对于计算机本身来说,它只能接受和处理由 0 和 1 代码构成的二进制指令或数据,由于这种形式的指令是面向机器的,因此也就称为“机器语言”。例如: 10101011 表示一条加法的机器指令。计算机发展的初期,程序设计人员使用机器语言来编制程序,它非常难于记忆,很不方便,但用机器语言编制的程序运行效率高。

2. 汇编语言与汇编程序

汇编语言是机器语言符号化的语言。例如: 机器指令 10101011 符号化后对应的汇编指令为: add ax,bx。一台机器的所有汇编指令集合就组成了汇编语言。汇编指令与机器指令一一对应,汇编语言和机器语言都属于低级语言。

用汇编语言编制的程序,计算机并不能直接识别和执行,必须经过汇编程序的翻译,然后才能运行。汇编程序的任务是将用汇编语言写的源程序(Source) 转换成用机器语言写的目标程序(Object) 。人们用汇编语言编程比用机器语言编程前进了一步,但使用起来仍然不方便。

3. 高级语言与编译程序

高级语言是接近于人类自然语言的、允许用英文和数学式子来表达的语言。它允许用英文编写解题的计算程序,程序中所使用的运算符号和运算式子,与我们日常用的数学式子差不多。高级语言容易学习,通用性强,书写出的程序比较短,便于推广和交流,是一种很理想的程序设计语言。目前常用的高级语言有 C、C + +、Java、. Net、Python 、COBOL、Basic、Fortran、Pascal 等,其中 C、COBOL、Basic、Fortran 和 Pascal 是面向过程的语言,而 C + +、Java、. Net和 Python 是面向对象的语言。图 1-2 中两数相加的例子,用面向过程的 C 语言表示为: c = a + b; 用面向对象的语言表示为: objA. add(a,b) ,objA 代表计算器对象。

高级语言编写的源程序不能被计算机直接识别和执行,必须经过翻译程序转换才能被执行。编译程序就是这样一种翻译程序,它将高级语言写的源程序翻译成功能等价的用机器语言写的目标代码。图 1-3 给出的用高级语言 C 编制的程序在上机运行的过程中由四个步骤组成:

(1) 上机输入和编辑源程序,生成源文件(后缀名为.c);

(2) 对源程序进行编译,生成目标文件(后缀名为.obj);

(3) 进行连接处理,将一个或多个目标文件连接生成可执行文件(后缀名为.exe);

(4) 运行可执行程序,得到运行结果。

在 1.4 节中将介绍使用集成开发环境对 C 程序进行编辑、编译、连接和执行的方法和步骤。

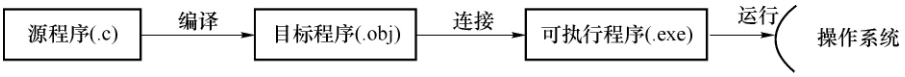

图 1-3　C 程序的上机运行过程

4. 高级语言程序的执行方式

高级语言程序的执行方式有两种:编译执行方式和解释执行方式。

编译执行方式是指:先把源程序整个翻译成可执行文件,这个过程称为翻译阶段;然后运行可执行文件,这个过程称为运行阶段。程序第一次执行时,需要翻译和运行两个阶段,而以后程序的多次执行,就不需要翻译阶段,只需要运行阶段,即直接运行可执行文件,所以程序的执行效率高。C、C++程序的执行,只能采用编译执行方式。

解释执行方式是对源程序采取边翻译边运行的过程,这个过程由解释器来完成,这个过程中并不生成可执行文件。解释方式下,程序每次执行需要翻译、执行,其执行效率比编译方式要低。Basic、Java 等程序的执行,可以采用解释执行方式。

1.2　C 语言的特点和应用

C 语言是一种通用的程序设计语言,既可用于编写计算机的系统软件,如操作系统、编译程序、设备驱动程序;又可用于编写一般的应用程序。在嵌入式系统,由于 C 语言既具有高级语言的特点,又能直接对机器硬件进行操作,因而得到了广泛的应用。

1.2.1　C 语言的特点

总的说来,C 语言具有以下主要特点:

(1) 语言表达简洁、使用方便灵活。

(2) 运算符丰富,包括算术运算、关系运算、逻辑运算和位运算等。

(3) 数据类型丰富,包括整型、实型、字符型、数组、指针、结构体、共用体等。

(4) 是面向过程的结构化程序设计语言,反映结构化的控制语句有:if…else、switch、while、do…while、for 等。

(5) 具有面向低级语言的特性,可直接对硬件编程,这样生成的目标代码质量高,程序执行效率高。

(6) 语法限制不太严格,语法表达灵活、多样,使程序设计人员在编程上有较大的自由发挥空间。

(7) 使用 C 语言编写的程序,很容易在不同的计算机之间进行移植,具有很好的移植性。

C 语言的以上特点,等学完本书内容后,读者一定会有深刻的理解,能应用 C 语言灵活高效地解决各种实际问题。

1.2.2 C 与 C++、Java、C#

C++是从 C 语言发展而来,并扩充了面向对象程序设计的成分,由于它兼容 C 语言,这就使得许多 C 代码不经修改就可被 C++编译通过。C++是兼容面向过程和面向对象的语言,语法规则完备且复杂。Java 对 C++进行了精简,增加了与 Internet 网络相关的成分,是完全面向对象的语言。C#吸收了 Java 大量的优点,是 Java 和 C++的杂合体。所以,我们学好 C 语言程序设计,可以为进一步学习 C++、Java 和 C#打下坚实的语言基础。

1.3 简单 C 语言程序入门

下面介绍几个简单的 C 语言程序。

例 1-1 在屏幕显示一行字符串。

```
#include <stdio.h>
int main( )
{
    printf("Programming is fun.\n");
    return 0;
}
```

程序运行结果如下:

```
Programming is fun.
```

在书写 C 语言源程序时,要注意:大写字母和小写字母具有不同的含义;语句之间的缩进,表明了语句之间的控制关系。另外为了增强程序的可读性,在单词之间可加一个或多个空格分隔。

程序中的语句说明如下。

(1) 语句的第一行:#include <stdio.h>是包含语句,指明将标准库函数包含到本程序中,库文件名 stdio.h 是用于提供输入/输出函数的,如输出函数 printf( ),输入函数 scanf( )。要注意#include 语句要放在程序的开始处。

(2) 从第二行开始往后的部分,定义了主函数 main,它是程序执行的入口点,在 C 语言程序中,必须有一个且只能有一个 main 函数。

int main( ) 是 main 函数的头,其中 int 是函数返回值的类型,返回整型 int,与下面的返回语句 return 0 相对应。main 是函数名,一对圆括号( ) 是定义函数的标志。

一对大括号{ }和其括住的两条语句是函数体,函数体定义了 main 函数要完成的功能。其中语句:

```
printf("Programming is fun. \n");
```

将在屏幕显示一行字符串“Programming is fun.”。“\n”是换行符,表示将光标移到下一行的起始处。语句:

```
return 0;
```

是返回语句,其功能是从 main 函数返回,并返回值 0, main 函数执行结束。

(3) 语句结束符号“;”。C 语言中的每一条语句都由分号(;)结束。一条语句可跨越多行。例如,跨越两行的语句:

```
total = a + b + c +
        d + e + f + g;
```

与语句:

```
total = a + b + c + d + e + f + g;
```

是等价的。

例 1-2　计算两个数的和。

```
//程序求两个数之和
#include <stdio.h>
int main( ) {
    int a,b,c;
    a = 10;      //设置变量 a 的值为 10
    b = 20;      //设置变量 b 的值为 20
    c = a + b;   //取变量 a 的值 10,与变量 b 的值 20 进行加法,将结果 30 赋给变量 c
    printf("Sum is %d\n",c);
    return 0;
}
```

程序运行结果如下:

```
Sum is 30
```

程序中的语句说明如下。

(1) 注释行

在本程序中第二行“//”后的内容为注释,进行编译时,这一行的所有内容会被忽略。注释只是帮助阅读理解程序的,并不影响程序的执行结果,编译器将忽略注释。C 语言的注释语句有两种形式:

(a) 单行注解。从“//”开始一直到当前行尾均为注释。例如:

```
//程序求两个数和
```

(b) 多行注释。可用于跨越多行的注释,“/* ”是注释的开始,“* /”表示注释结束。例如:

```
/*  一行或
多行的注释 * /
```

(2) 变量声明语句

变量声明语句:

```
int a,b,c;
```

将声明三个整型变量,变量名分别为 a,b,c。给整型变量设置的值只能为整数。

(3) 赋值语句

赋值语句:

```
a = 10;
b = 20;
```

完成给变量 a 和变量 b 分别赋予(或设置)值 10 和 20。

赋值语句:

```
c = a + b;
```

完成取变量 a 的值 10,与变量 b 的值 20 进行加法,将相加结果 30 赋给变量 c。

(4) 打印语句

打印语句:

```
printf("Sum is %d\n",c);
```

这里的 printf 函数带有两个参数,两个参数之间以逗号“,”分隔。第一个参数是要打印的格式字符串“Sum is %d\n”,它由固定的文本和格式说明符组成,其中固定的文本直接由 printf 输出,例如,“Sum is ”(表示输出 Sum is),“\n”(表示输出一个换行)。而以“% ”开头的格式说明符是占位符,在输出时会被一个值替换。例如“%d”会被替换成一个十进制整型值。第二个参数是变量 c,是用变量 c 的值 30 去替换前面的格式说明符%d,即得到输出结果“Sum is 30”。

1.4 C 语言程序运行环境的安装和使用

集成开发环境(Integrated Development Environment,IDE)一般都提供可视化界面用于程序的编辑、编译、调试、运行、项目管理等一体化功能。C 语言的集成开发环境有很多种类,常用的有 CodeBlocks、Dev-C++、Visual C++、Turbo C、Borland C++等。其中 Turbo C 是早期 DOS 环境下使用的 IDE。Visual C++是微软推出的 Windows 平台上的一款C/C++开发环境,它被包含在 Visual Studio 集成开发包中,但随着 Visual Studio 版本的不断更新(目前最新版本 2017),下载和安装它对于初学者比较麻烦。而 Visual C++ 6.0 是微软在 1997 年推出的一款经典的 C/C++编译器,它界面友好,调试功能强大,安装便捷,是全国计算机二级等级考试 C 语言的机考环境。CodeBlocks 和 Dev-C++是两款免费开放源代码的软件,是中国计算机学会组织的计算机软件能力认证 CCF CSP 的机考环境。

本节将分别介绍 CodeBlocks 和 Visual C++开发环境的下载、安装和使用,比较适合于 C 语言的初学者,读者在学习 C 语言时,只需要选择其中一种开发环境安装在自己的机器上即可。

1.4.1 在 CodeBlocks 集成开发环境下执行 C 语言程序

1. CodeBlocks 下载和安装

CodeBlocks 是一个免费的跨平台 C/C++集成开发环境,跨平台是指它在 Windows、Linux 等 OS 平台上都能安装,而且运行高效。从官网 http://www.codeblocks.org/可以下载不同操作系统下的软件版本。本书推荐下载的是 Windows 下的版本“codeblocks-16.01mingw-setup.

此为试读,需要完整PDF请访问: www.ertongbook.com