

◎ 计算机

“卓越工程师计划”应用型教材



Computer



C语言 程序设计教程

C Language Programming Tutorial

◎ 张宗杰 主编

◎ 向琳 徐展翼 黄刚 副主编



电子工业出版社

PUBLISHING HOUSE OF ELECTRONICS INDUSTRY <http://www.phei.com.cn>

计算机“卓越工程师计划”应用型教材

C 语言程序设计教程

张宗杰 主编

向琳 徐展翼 黄刚 副主编

電子工業出版社

Publishing House of Electronics Industry

北京·BEIJING

内 容 简 介

本书分基础篇、进阶篇和提高篇。基础篇包括第1章至第3章，主要介绍利用整型变量进行简单的控制结构程序设计；进阶篇包括第4章至第9章，主要阐述C语言程序设计的基础知识、结构化程序设计相关内容；提高篇包括第10章至第13章，主要介绍一些复杂的数据组织结构，为读者进一步编写复杂程序打下基础。本书每章都附有习题，并提供免费配套的电子教案。

本书适用于作为应用型本科院校计算机专业及理工科类非计算机专业的C语言程序设计教材，也可以作为计算机爱好者学习C语言程序设计的参考书。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

C语言程序设计教程 / 张宗杰主编. —北京：电子工业出版社，2013.8

计算机“卓越工程师计划”应用型教材

ISBN 978-7-121-20474-6

I. ①C… II. ①张… 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(2013)第105841号

责任编辑：刘海艳

印 刷：北京京师印务有限公司

装 订：北京京师印务有限公司

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：787×1 092 1/16 印张：17.75 字数：454.4千字

印 次：2013年8月第1次印刷

印 数：4 000册 定价：35.00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 zltz@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010) 88258888。

丛书序言

党的十八大提出要“努力办好人民满意的教育”，要“推动高等教育内涵式发展”，“全面实施素质教育，深化教育领域综合改革，着力提高教育质量，培养学生社会责任感、创新精神、实践能力。”这对高等教育提出了新的要求，明确了人才培养的目标和标准。

十八大明确指出“坚持走中国特色新型工业化、信息化、城镇化、农业现代化道路，推动信息化和工业化深度融合、工业化和城镇化良性互动、城镇化和农业现代化相互协调，促进工业化、信息化、城镇化、农业现代化同步发展。”“推动信息化和工业化深度融合”对高等工程教育改革发展提出了迫切要求。

遵照《国家中长期教育改革和发展规划纲要（2010—2020年）》和《国家中长期人才发展规划纲要（2010—2020年）》，为贯彻落实教育部“卓越工程师教育培养计划”，促进我国由工程教育大国迈向工程教育强国，培养造就一批创新能力强、适应经济社会发展需要的高质量计算机工程技术人才，电子工业出版社决定组织相关实施和计划实施卓越计划以及江浙两省实施软件服务外包人才培养试点的地方高校的相关教师，在以往实践校企合作人才培养的基础上编写一套适合地方高校的计算机“卓越工程师计划”人才培养系列教材。

我们将秉承“行业指导、校企合作、分类实施、形式多样”的“卓越工程师教育培养计划”四原则，坚持“学科规范、本科平台、行业应用”，以“具备较为扎实的专业基础知识、拥有良好的职业道德素质、具有创新的计算机应用能力”为目标，探索“校企一体化”产学研结合人才培养模式改革，强化“岗位目标、职业培养”，努力实现计算机工程型技术人才（应用型）培养目标：

（1）尝试以“知识保障、能力渐进、素质为本，重视技术应用能力培养为主线”，坚持以“素质教育，能力培养”为导向，体现本科平台、能力定位、应用背景构建课程体系。

（2）尝试“以学生工程意识、创新精神和工程实践能力培养”为核心，坚持以“培养学生的工程化开发能力和职业素质”为原则，校企合作构建实践教学体系。

本系列教材基于“以德为先、能力为重、全面发展”的人才培养观念，在内容选择、知识点覆盖、课程体系安排、实践环节构建、企业强化训练上按照能力培养和满足职业需求为本进行了有益的、初步的探索。

然而，由于社会对计算机人才的需求广泛而多样，各领域的人才规格和标准既有共性又有特殊性，同时各相关高校在计算机相关专业设置以及人才培养的探索上各有特点，我们编写的本套系列教材目前只能部分满足计算机相关专业人才培养的需要。我们力争建立一个体系，以模块构建的增量方式实现教材编写的滚动、增加和淘汰，逐步建设可供地方高校计算机不同专业、针对不同领域培养计算机工程技术人才选择的教材库：①所有专业的公共基础课相对统一，不同专业的专业基础课按模块划分、各自专业的专业课按领域整合、拓展课紧跟技术和行业发展；②公共基础课、专业基础课以

经典知识为主，专业课、拓展课与国际主流技术接轨；③实践环节或实践课程必须接纳企业文化、优选企业实际工程项目，体现校企合作、重视企业导师的参与。

“卓越工程师教育培养计划”的实施具有三个特点：一是行业企业深度参与培养过程；二是学校按通用标准和行业标准培养工程人才；三是强化培养学生的工程能力和创新能力。

本系列教材的编写得到了中软国际、苏州花桥国际商务区（及所属企业）、常州创意产业基地（及所属企业）等热心和关注计算机类人才培养的国家重点企业、园区的大力支持。我们曾以“目标明确、责任共担、实现共赢”为原则探索了多种人才培养合作途经：从师资培养到校企共建实训基地，到建立校内软件学院，再到学生进企业强化、顶岗实训……取得了一定的经验。在“卓越工程师教育培养计划”的实施中，企业和学校签订了全面合作协议，共同确定人才培养标准、制订人才培养方案、参与人才培养过程，提供企业学习课程和项目案例，确保学生在企业的学习时间。

同样，本系列教材的编写总结了参编高校和支撑企业在校企合作人才培养过程中共同取得的经验和教训，并涵盖了我们已经做的、想要做的实施卓越计划的理念和努力。这仅是初步的尝试，会存在许多不足和缺陷，但希望由此能起到抛砖引玉的作用。在卓越计划的实施探索中，我们衷心地希望能有更多的地方高校计算机院系、更多的行业企业加入团队，面对企业必须参与的国际化产业竞争，为培养优秀的、具有应用创新精神的计算机工程技术（包括软件）人才，企业和学校能深度合作、各尽职责；每一位教育工作者都能贡献自己的聪明才智，尽一份绵薄之力。

对给予本套丛书编审大力支持的江苏计算机学会、中国矿业大学计算机学院以及参与编写教材的高校、单位表示由衷的感谢！

计算机“卓越工程师计划”应用型教材编委会

前 言

众所周知，UNIX 系统及运行于其上的软件都是用 C 语言编写的。C 语言在很多方面继承和发展了以往许多高级程序设计语言的成功经验和特色，已经成为编写系统软件、应用软件重要编程语言，尤其在嵌入式系统、工业控制中更是得到广泛应用。从 C 语言产生至今，它已经成为最重要和最流行的编程语言之一。在各种流行编程语言中，都能看到 C 语言的影子，如 Java 的语法与 C 语言基本相同。学习、掌握 C 语言既可以增进对于计算机底层工作机制的了解，又为进一步学习其他高级语言打下了坚实的基础。是每一个计算机技术人员的基本功之一。

目前，很多高校都选用 C 语言作为程序设计的指定语言。由于 C 语言牵涉概念多、语法比较复杂、内容广泛，许多学生感到学习难度较大，难以入门。目前，许多 C 语言教材按照知识的体系结构组织内容，而忽略了教学过程的实用性。针对这种情况，我们在 2008 年编写了《C 语言程序设计实用教程》，该教程结合学生的认知特点，把相关知识分层次地讲解，效果不错。此次教材编写，面向卓越工程师计划，注重编程能力和语言知识的培养，对以前教材的不足之处进行改进，部分内容做了结构上的修改，尽量精简理论内容，兼顾后继课程需求，重点放在理论知识在计算机学科中的应用上。全书内容分为基础篇、进阶篇、提高篇。在基础篇中，在必要的准备知识后即开始简单的程序设计，使学生能很快上手，掌握程序设计的基本技能。在学生对 C 程序设计有了一定的基础之后，进入进阶篇，详细地介绍 C 语言的数据类型、运算符及表达式、输入/输出语句，对基础篇的内容进行拓展，学习 C 语言的精华部分函数和指针，讲述相对复杂的算法和程序，使学生的编程能力在循序渐进中得到提高。在提高篇中对函数和指针做了深入讲解，同时介绍一些构造数据类型、文件等内容，为学生编写复杂的 C 程序打下基础。

全书具体内容安排如下：

（1）基础篇

基础篇包括第 1~3 章。该部分在介绍整型变量及相关运算的基础上，通过学习简单的控制结构程序设计，使读者能尽快上手编写简单的 C 程序。

第 1 章 C 语言概述：主要介绍 C 语言的发展及特点、C 程序的基本结构及书写风格、Visual C++ 6.0 开发环境，并通过简单的实例引出常量与变量的概念及整型变量的输入输出格式。

第 2 章 简单程序设计之一：主要介绍 C 语言提供的三种控制流程及程序设计的基本方法，并对控制结构涉及的运算符和表达式做简要介绍。

第 3 章 一维数组：主要介绍一维数组的定义、引用、初始化及应用，算法的基本概念和用于描述算法的流程图，VC6.0 下 C 程序的调试方法。

（2）进阶篇

进阶篇包括第 4~9 章。该篇系统地介绍了 C 语言的基本数据类型，部分运算符和

表达式，函数、指针的概念和初步应用。

第4章 C语言基本语法：主要介绍C语言基本数据类型，不同类型数据的性质、表示形式、构造特点、类型转换、输入/输出等，还介绍了自增自减运算符、条件运算符、赋值运算符、逗号运算符及它们的表达式。

第5章 简单程序设计之二：主要介绍一些和实际应用关系密切的实例，如数学计算问题、枚举问题、查找问题、排序问题、菜单设计。通过实例进一步加强控制结构程序设计的练习。

第6章 二维数组和字符数组：主要介绍二维数组的定义、引用、初始化和应用。作为数组的特例，介绍字符数组的特点、输入/输出、字符串处理函数及应用。

第7章 函数：主要介绍函数的基本概念及利用函数进行模块化程序设计的基本方法，变量的作用域和生存期及内外部函数。

第8章 指针：介绍指针变量的概念、特点、定义和使用方法，数组与指针的关系，指针变量作函数参数，指向数组的指针作函数的参数等。

第9章 编译预处理和位运算：主要介绍带参数及不带参数的宏定义、文件包含、条件编译，位运算符和位运算的特点和应用。

(3) 提高篇

提高篇包括第10~13章。该篇主要介绍指针的高级应用，结构体等构造类型的使用，用文件组织数据的方法等，为读者进一步编写复杂程序打下基础。

第10章 指针的高级应用：主要介绍指针型函数、指向函数的指针、指向二维数组的指针、指针数组和指向指针的指针等内容。

第11章 结构体、共同体和枚举类型：介绍几种构造数据类型。主要介绍结构体类型的变量、数组、指针及指针数组的定义、初始化、引用、赋值、输入、输出方法。

第12章 文件：主要介绍文件及文件指针的概念；文件的打开和关闭操作及对打开和关闭操作的错误处理；基于字节、数据项的读/写操作；文件的随机读/写及错误检测等。

第13章 C程序应用举例：通过几个综合实例的完整设计过程，使读者对C语言程序设计有更全面的认识。

本教材面向卓越工程师计划，注重编程能力和语言知识的培养，结合实例，引导学生掌握程序设计的基本方法，适用于应用型本科院校各专业的学生学习使用。该书作为教程编写，注重教学过程，有些知识分散在不同的章节逐步讲解，必要时读者可将它们组织到一起来看。

为了便于考查学习效果，每章后附有习题。习题分为选择题、填空题和编程题。选择题侧重基础知识的巩固；填空题侧重提高学生的阅读程序的能力；编程题则侧重所学知识的运用及编程能力的提高。本书中所有的例题都是在 Visual C++6.0 环境下调试的。

在本书的写作过程中，任建平教授提出很多中肯的建议，并在百忙中审阅了全部书稿，在此表示深深的谢意。衷心感谢苏州科技学院C语言教学研究组的全体教师。几年来，这些教师不断探索新的教学方法，总结出一套效果显著、学生容易接受的教学方

法，即将全书内容分为基础篇、进阶篇、提高篇 3 部分，将数据类型及运算符和表达式的相关知识穿插在程序设计中介绍。几个学期的实践证明该教法效果非常好。该教法体现了 C 课程组全体教师的集体智慧。本教程即是该教法的体现。

本书由张宗杰主编，参加编写的还有向琳、徐展翼、黄刚、杨文硕、任建平、张明亮、严迪新、陆卫忠、倪启东、杨永生、赵梅。

由于作者水平有限，加上编著、出版时间仓促，难免存在错误和不妥之处，恳请专家和广大读者批评指正。

编 者

目 录

第一篇 基 础 篇

第 1 章 C 语言概述	1
1.1 C 语言的发展及特点	1
1.2 简单的 C 程序设计	2
1.2.1 认识 C 语言程序	2
1.2.2 数据类型、常量与变量	4
1.2.3 数据输入/输出在 C 语言中的实现	5
1.3 C 程序的基本结构和书写风格	6
1.3.1 C 程序的基本结构	6
1.3.2 C 程序的书写风格	8
1.4 C 程序的上机步骤和 Visual C++ 6.0 的基本用法	8
1.4.1 C 程序的上机步骤	8
1.4.2 Microsoft Visual C++ 6.0 的基本用法	9
小结	12
习题 1	12
第 2 章 简单程序设计之一	14
2.1 顺序结构程序设计	14
2.1.1 算术运算符及其表达式	14
2.1.2 顺序结构程序设计举例	15
2.2 选择结构程序设计	16
2.2.1 关系运算符及其表达式	16
2.2.2 逻辑运算符及其表达式	17
2.2.3 if 分支结构	18
2.2.4 switch 多分支结构	23
2.3 简单循环结构程序设计	26
2.3.1 为什么使用循环结构	26
2.3.2 while 语句和 do-while 语句	27
2.3.3 for 语句	30
2.3.4 几种循环的比较	32
2.4 break 和 continue 语句在循环中的应用	32
小结	34
习题 2	34
第 3 章 一维数组	38
3.1 为什么要使用数组	38
3.2 一维数组	39
3.2.1 一维数组的定义	39
3.2.2 一维数组元素的引用	40

3.2.3 一维数组的初始化	40
3.2.4 一维数组应用举例	41
3.3 程序调试	42
3.3.1 用输出语句调试程序	42
3.3.2 利用 VC 调试工具调试 C 程序	43
3.4 算法及算法描述	50
3.4.1 算法的概念及特征	50
3.4.2 算法的描述	52
3.4.3 用计算机语言表示算法	54
小结	54
习题 3	55

第二篇 进 阶 篇

第 4 章 C 语言基本语法	57
4.1 C 语言的数据类型	57
4.2 整型数据	58
4.2.1 整型常量	58
4.2.2 整型变量	59
4.3 实型数据	60
4.3.1 实型常量	60
4.3.2 实型变量	60
4.4 字符型数据	61
4.4.1 字符型常量	61
4.4.2 字符型变量	62
4.4.3 字符数据的输入/输出	64
4.5 格式化输入/输出	65
4.5.1 格式化输出函数 printf()	65
4.5.2 格式化输入函数 scanf()	67
4.6 数据类型转换	68
4.6.1 自动类型转换	69
4.6.2 强制类型转换	70
4.7 运算符与表达式	70
小结	73
习题 4	74
第 5 章 简单程序设计之二	76
5.1 嵌套循环结构程序设计	76
5.2 数学计算问题	78
5.3 枚举问题	78
5.4 查找问题	80
5.5 排序问题	81
5.6 菜单设计	84

小结	86
习题 5	86
第 6 章 二维数组和字符数组	90
6.1 二维数组	90
6.1.1 二维数组的定义	90
6.1.2 二维数组元素的引用	91
6.1.3 二维数组的初始化	91
6.1.4 程序举例	92
6.2 字符数组	95
6.2.1 字符数组的定义及初始化	95
6.2.2 字符串的输入/输出	96
6.2.3 常用的字符串处理函数	97
6.2.4 应用举例	100
小结	102
习题 6	103
第 7 章 函数	107
7.1 认识函数	107
7.1.1 模块化程序设计	107
7.1.2 函数的概念和分类	109
7.2 函数的定义及调用	110
7.2.1 函数的定义	110
7.2.2 函数的调用	111
7.3 函数参数和返回值	112
7.3.1 形式参数和实际参数	112
7.3.2 函数的返回值	114
7.4 函数的声明	116
7.5 函数的嵌套与递归	118
7.5.1 函数的嵌套	118
7.5.2 函数的递归调用	119
7.6 数组名作为函数的参数	121
7.6.1 实参到形参的数据传递方式	121
7.6.2 数组作为函数的参数	121
7.7 变量的作用域及存储特性	123
7.7.1 变量的作用域	123
7.7.2 变量的存储特性	125
7.8 内部函数和外部函数	128
7.8.1 内部函数	128
7.8.2 外部函数	128
小结	130
习题 7	130

第 8 章 指针	136
8.1 指针类型	136
8.1.1 变量的地址和变量的值	136
8.1.2 变量的直接访问和间接访问	137
8.1.3 指针变量的定义	137
8.1.4 指针变量的引用	138
8.1.5 空指针和 void 型指针	139
8.2 指针运算	140
8.2.1 指针的赋值	140
8.2.2 指针的加减运算	141
8.2.3 指针的比较	142
8.3 指针变量作为函数参数	142
8.3.1 在函数中处理主调函数中的变量	143
8.3.2 利用指针将函数处理结果带回主调函数	145
8.4 用指针访问一维数组	146
8.4.1 指针与一维数组的关系	146
8.4.2 指向数组的指针变量的定义	146
8.4.3 利用指针访问数组	147
8.4.4 动态申请存储空间	149
8.5 数组名作为函数的参数	151
8.6 字符串与指针	153
小结	155
习题 8	156
第 9 章 编译预处理和位运算	161
9.1 宏定义	161
9.1.1 不带参数的宏定义	161
9.1.2 带参数的宏定义	164
9.2 文件包含	168
9.3 条件编译	170
9.4 位运算	173
9.4.1 位运算和位运算符	173
9.4.2 位段	173
小结	175
习题 9	176

第三篇 提 高 篇

第 10 章 指针的高级应用	178
10.1 指针与函数	178
10.1.1 指针型函数	178
10.1.2 指向函数的指针	179
10.2 用指针访问二维数组	181

10.2.1 二维数组元素的地址	182
10.2.2 指向二维数组元素的指针	183
10.2.3 指向二维数组中一行元素的指针	183
10.2.4 指针数组	185
10.2.5 二维数组名作函数参数	186
10.3 二级指针	187
10.3.1 多级指针的概念	187
10.3.2 二级指针的定义及使用	188
10.4 指针数组作 main 函数的形参	189
小结	190
习题 10	190
第 11 章 结构体、共同体和枚举类型	193
11.1 结构体类型	193
11.1.1 结构体类型的认识	193
11.1.2 结构体类型的定义	194
11.2 结构体变量	195
11.2.1 结构体变量的定义	195
11.2.2 结构体变量的存储模式	196
11.2.3 结构体变量的引用	196
11.2.4 结构体变量的基本操作	197
11.3 结构体数组	199
11.3.1 结构体数组的定义和引用	199
11.3.2 结构体数组的初始化	200
11.4 结构体类型指针	202
11.4.1 结构体指针的定义	203
11.4.2 通过结构体指针引用结构体变量	203
11.4.3 结构体指针作函数参数	205
11.5 链表	206
11.5.1 链表概述	206
11.5.2 链表的基本操作	207
11.5.3 链表综合应用举例	212
11.6 共用体	214
11.6.1 共用体的概念	214
11.6.2 共用体类型的应用	215
11.7 枚举	217
11.7.1 枚举类型及枚举变量的定义	217
11.7.2 枚举类型的应用	218
11.8 类型定义	219
小结	220
习题 11	220

第 12 章 文件	224
12.1 文件概述	224
12.1.1 文本文件和二进制文件	224
12.1.2 缓冲文件系统和非缓冲文件系统	225
12.2 文件类型指针	225
12.2.1 文件类型指针的定义	225
12.2.2 设备文件	226
12.3 文件的打开和关闭	226
12.3.1 文件的打开 fopen()	226
12.3.2 文件的关闭 fclose()	227
12.4 文件读/写操作	228
12.4.1 字符读/写函数: fgetc()和 fputc()	228
12.4.2 字符串读/写函数: fgets()和 fputs()	230
12.4.3 数据块读/写函数: fread()和 fwrite()	231
12.4.4 格式化读/写函数 fscanf()和 fprintf()	234
12.5 文件定位函数	235
12.5.1 文件指针的重置函数 rewind()	235
12.5.2 取文件的当前位置函数 ftell()	236
12.5.3 文件读/写指针定位函数 fseek()	236
12.6 出错检测函数	237
12.6.1 ferror()函数	237
12.6.2 clearerr()函数	238
小结	238
习题 12	238
第 13 章 C 程序应用举例	241
13.1 Josephus 问题	241
13.2 字符串处理问题	245
13.3 学生信息管理应用程序	248
习题 13	256
附录 A 常用字符与 ASCII 代码对照表	258
附录 B 运算符与结合性	260
附录 C C 语言中的关键字	261
附录 D C 语言库函数	262
参考文献	265

第一篇 基础篇

第 1 章

C 语言概述

本章要点

- ◆ C 程序的基本结构和书写风格
- ◆ 使用开发环境 Visual C++编写运行简单 C 程序
- ◆ 常量和变量的概念
- ◆ 基本输入/输出函数的初步使用

1.1 C 语言的发展及特点

C 语言是国际上广泛流行的计算机高级语言，既可以用来编写系统软件，也可以用来编写应用软件。由于 C 语言强大的功能及接近低级语言的效率，尽管面向对象语言的应用已十分广泛，C 语言在计算机程序设计领域中仍发挥着重要的作用。

进入 20 世纪八九十年代，计算机语言从面向过程转化为面向对象程序设计，可视化多媒体程序设计模式的概念得到了推广、普及。C 语言在发展的同时，也朝着支持面向对象、可视化程序设计语言方向发展，如 Borland C++、Visual C++、C++ Builder 等先后问世。C 语言已经成为编写系统软件、应用软件和进行程序设计、教学的重要编程语言，甚至许多硬件开发系统也使用 C 语言。C 语言已经成为世界上应用最广泛的计算机语言之一。

计算机的每一种语言都有极大的相通之处，又都有各自的独到之处。

C 语言既有高级语言的优点，又在很多方面保留了低级语言速度快，具有可进行很多直接映射硬件结构的操作的长处。例如，我们无时不用的 Windows 等操作系统，就是用 C 语言和汇编语言写的。正因为这个优点，事实上很多人称 C 语言为中级语言。用 C 语言编写的程序非常简洁，运算符和数据结构非常丰富，具有结构化的控制语句，允许直接访问物理内存，能实现对位的操作，生成的目标代码质量高，只比汇编语言生成的代码效率低 20% 左右。

C 语言有其优秀的实现工具。微软公司的 Microsoft Visual Studio 6.0 中的 Microsoft Visual C++ 6.0 (VC6.0) 和 Borland 公司的 Turbo C 都是相当优秀的集成开发环境 (IDE)。

1.2 简单的 C 程序设计

1.2.1 认识 C 语言程序

程序一词来源于生活。在现实生活中，我们做任何事情都有一定的步骤，一步步地去做，最终达到我们想要的结果。

计算机程序是一组指令的集合，这些指令被逐条执行，直到完成特定的操作。

我们通过下面的实例认识一下 C 语言程序。

【例 1.1】编写程序在显示器输出一行信息：Hello, World!

〔程序代码〕

```
#include <stdio.h>          /*文件包含，编译预处理命令*/
int main()
//在屏幕上显示 Hello, World!
{
    printf("Hello, World!\n");    /*显示文本 Hello, World!*/
    return 0;
}
```

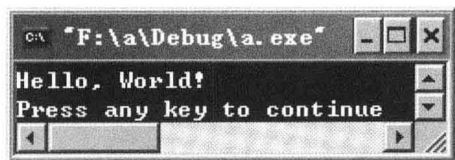


图 1.1 【例 1.1】的运行结果

程序运行结果如图 1.1 所示。

这是一个很简单的 C 程序，它在屏幕上显示一行文本“Hello, World!”。

结合这个程序，给出 C 语言程序的几点说明如下：

(1) 为了方便程序的阅读，C 语言允许我们在程序中写上一些说明性文字，这就是代码的注释。注释并不被编译器处理，它的书写只是为了增加程序的可读性。在 C 语言中，常用的注释有两种：行注释 (//) 和块注释 (/*.....*/)。例如：

```
/* This is a main function */
//这是变量声明
```

(2) 第 2 行到第 7 行定义了一个名为 main 的函数。

```
int main()
{
    .....
    return 0;
}
```

如同盖房子要先搭一个架子，然后才能添砖加瓦，C 程序也有自己的架子。C 程序可以由多个函数构成。所有的 C 程序都有且只有一个 main() 函数。C 程序从 main() 函数处开始执行，当 main() 函数结束时，程序也就结束了。正如我们的房子，不论有多少个房间都要从门进入一样，main 函数是 C 程序的入口也是出口，没有 main 函数，计算机就不知道从哪里执行程序。

main() 函数的最后一条语句是

```
return 0;
```

它结束 `main()` 函数的运行，并向系统返回一个整数 0，作为程序结束状态，这条语句可以省略。实际上，主函数 `main()` 还可以定义成没有返回值的形式：

```
void main()
{
    .....
}
```

(3) 花括号括起来的是函数体，其内部就是我们让计算机执行的指令。该程序只完成一个操作，那就是在显示器显示“Hello, World!”。`printf()` 是 C 语言自带的函数，使用它可以向控制台输出信息。程序中，我们只要把需要输出的话用英文引号引起来就可以了。双引号中的“`\n`”表示输出一个换行。

(4) C 语言的每一条语句都以分号“`;`”结束。

C 语言是一门高级语言，用它来与特定对象（计算机）进行沟通，那么 C 语言如何表达数据并描述相应的算法呢？让我们看下面的实例：

【例 1.2】 求长方形的周长和面积。

〔程序分析〕

(1) 问题所涉及的数据：输入数据为长、宽。输出数据为周长、面积。

(2) 问题所涉及的算法：

周长=(长+宽)×2 面积=长×宽

〔计算机解决办法〕

(1) 变量定义。长、宽两个变量的名称用 `a` 和 `b`，周长和面积两个变量的名称用 `l` 和 `s`，语句如下：

```
int a,b,l,s;
```

可以每行定义一个变量，也可以一行定义若干个变量，变量间用逗号“`,`”分隔。

(2) 算法实现。周长计算： $l = (a+b) \times 2$ 面积计算： $s = a \times b$

〔程序代码〕

```
#include <stdio.h>
int main ()
{
    int a, b, l, s;                /*定义变量*/
    printf("请输入长方形的长和宽: ");
    scanf("%d%d",&a,&b);
    l=2*(a+b);                    /*求长方形的周长*/
    s=a*b;                        /*求长方形的面积*/
    printf("周长=%d, 面积=%d\n", l,s); /*输出结果*/
    return(0);                    /*返回值*/
}
```

程序运行结果如图 1.2 所示。

该程序涉及这样一些问题：

(1) 整型变量定义及引用；

(2) 输入函数 `scanf()` 和输出函数 `printf()` 的使用。

下面将逐一介绍。



图 1.2 【例 1.2】的运行结果