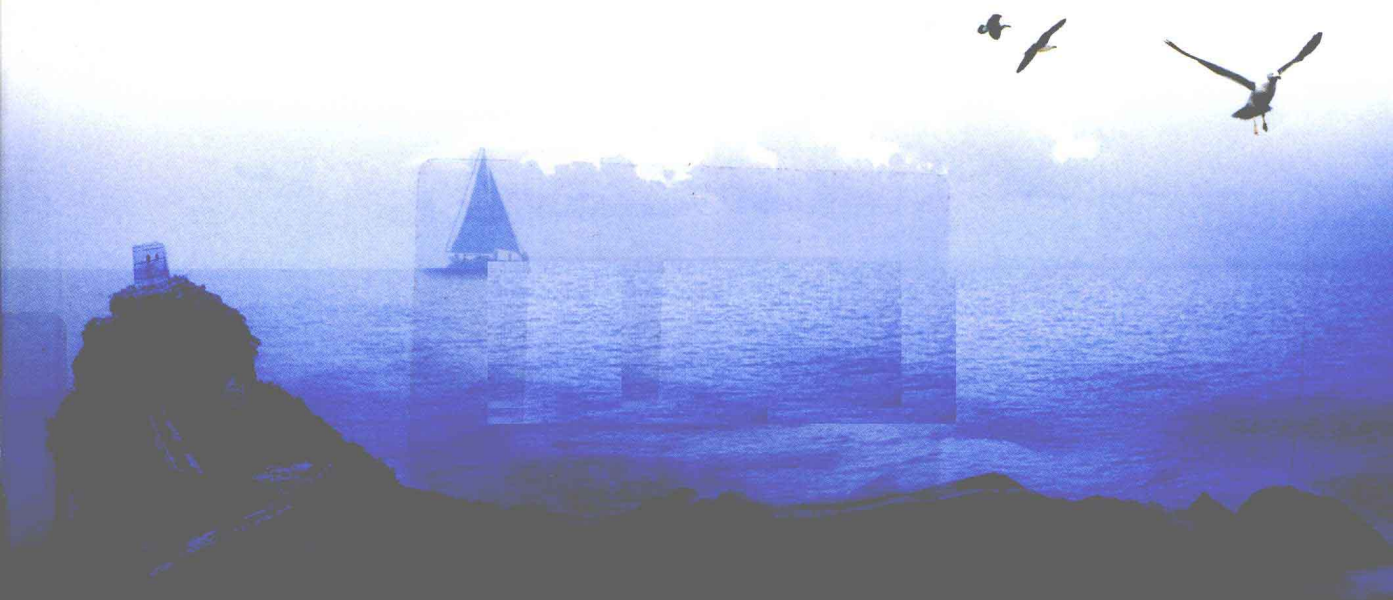
29个案例与项目分析 电子课件及源程序下载

C程序设计与项目实践

王一萍 梁伟 金梅 编著



突出项目实践：，由浅入深、不同类型、贴近实际

启发编程思想：同一个案例可能采用不同的方法，

引导多读源码：展现案例和项目的，在代码阅读中提升能力

力求实用通俗：案例选取多样，编排循序渐进，

清华大学出版社

C 程序设计与项目实践

王一萍　梁　伟　金　梅　编著

清华大学出版社

北　京

内 容 简 介

本书是程序设计类教材，在系统化介绍C语言语法知识的前提下，致力于培养学生利用所学知识进行程序设计和项目实践的能力。全书通过数十个精心设计、由浅入深、贴近实际的案例和小项目的分析讲解，帮助学生学以致用、轻松入门和快速提高。

为了便于初学者学习，本书列出了所有案例和项目的完整源代码以及运行效果图，供学习者阅读、分析、领悟和超越。另外，本书还为任课教师免费提供电子课件，包括教学用PPT、全部案例和项目的完整源代码，以方便教学者教学。

本书特别适合作为高等院校相关专业学生学习程序设计课程的教材，也可作为相关人员自学程序设计的教材和参考书。

图书在版编目（CIP）数据

C程序设计与项目实践/王一萍，梁伟，金梅编著. —北京：清华大学出版社，2011.10

ISBN 978-7-302-26922-9

I. ①C…　II. ①王…　②梁…　③金…　III. ①C语言-程序设计　IV. ①TP312

中国版本图书馆CIP数据核字（2011）第187898号

责任编辑：赵洛育
版式设计：文森时代
责任校对：张彩凤　张兴旺
责任印制：李红英
出版发行：清华大学出版社　　**地　　址**：北京清华大学学研大厦A座
http://www.tup.com.cn　　**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：北京密云胶印厂
经　　销：全国新华书店
开　　本：185×260　**印　张**：30.25　**字　数**：699千字
版　　次：2011年10月第1版　　**印　　次**：2011年10月第1次印刷
印　　数：1～5000
定　　价：49.80元

产品编号：044324-01

前　言

C 语言是近几十年来影响最为深远的程序设计语言，在它的基础上诞生了 C++、Java 和 C#等当前非常流行并且极具生产力的程序设计语言，许多计算机专业人员和计算机爱好者都把 C 语言作为学习程序设计语言的首选。

本书不是一本大而全的 C 语言语法教科书，而是一个能够让初学者轻松入门和快速提高的“言传身教”者。在编写过程中，我们始终坚持理论与实践并重，理论深入浅出，高屋建瓴，实践扎扎实实、脚踏实地。

本书在系统化介绍 C 语言语法知识的前提下，致力于培养学生利用所学知识进行程序设计和项目实践的能力。全书通过数十个精心设计、由浅入深、贴近实际的案例和小项目的分析讲解，帮助他们学以致用、轻松入门和快速提高。本书案例精心设计，来源于长期的教学实践，覆盖了尽可能多的类型（如数值计算类、逻辑推理类、小游戏类、事务处理类、算法学习类、设计方法类等），每一个案例都尽可能贴近实用、有意义，让初学者爱学好学、容易上手，并且在学习过程中不知不觉、循序渐进地就掌握了众多实用的案例，积累了丰富的小项目开发实践经验。

本书一开始就促使程序设计初学者把学习的焦点集中在如何利用程序设计语言进行程序设计上，而不是死抠语法，纸上谈兵。“读书百遍，不如抄书一遍”，程序设计更是“读书百遍，不如敲书一遍”。为了便于初学者学习，本书列出了所有案例和项目的完整源代码以及运行效果图，供学习者阅读、分析、领悟和超越。

本书作为程序设计类教材，着重强调超越语言的各种语言概念和指导程序设计实践的相关思想与理念。从一开始就强调注重代码的书写风格，便于初学者模仿，逐渐养成良好的程序设计习惯。侧重讲解了迭代式、增量式的模块化程序分析、设计方法。在案例和小项目实践中逐步地引入了测试、程序的调试、输入验证、持久存储等使程序更加健壮和实用化的考虑。同一个案例或小项目在不同的章节往往采用不同的数据结构、算法以及程序架构来进行重构，使它们尽可能地贴近实际，让学习者持久地思考和改进它们，为将来企业级的开发做足准备。

通过本教材，学习者可以在贴近实际的案例和项目中更容易地理解和掌握相关的程序设计语言的概念和语法知识，奠定扎实的基本功，同时积累丰富的小项目开发经验。

本书注重教材的可读性和实用性，在每章的开头都给出了本章的学习目标，重要的知识点都给出了完整的程序示例和运行效果，结尾部分给出了本章知识点的总结，并在习题与实践中给出了课后进一步提升的程序设计实践项目。

本书共分 9 章。第 1 章 C 语言程序设计导引，介绍什么是计算机、什么是程序、什么是程序设计语言、什么是程序设计、C 语言的相关背景知识（历史沿革及特点）、C 程序设计快速入门三要素（快速开发 HelloWorld 程序、掌握基本的输入/输出手段、C 程序的基本

组织结构），以及HelloWorld程序扩展的几个小案例（字符界面设计、纸张对折到月球、IPv6地址分配、计算机的计时系统、超市计费系统1.0版）；第2章数据类型、运算符与表达式，介绍C语言的基本数据类型、常量和变量、各种运算符和表达式、数据类型转换，以及两个小案例（计算机中的数学和3种方式交换两个整数）；第3章算法与流程控制，介绍什么是算法、算法的流程图表示、基本的流程控制语句（顺序、分支、循环），以及3个小案例（超市计费系统2.0版、模拟龟兔赛跑1.0版、猜数游戏1.0版）；第4章函数与模块化程序设计，介绍函数的概念、定义、声明、调用，参数传递的两种方式，函数的嵌套和递归调用、变量的存储类别、生存期和作用域，编译预处理（包含头文件、宏定义及宏展开、条件编译及特殊符号处理），模块化的编译和链接，以及5个利用模块化方法开发的小案例（模拟龟兔赛跑2.0版、猜数游戏2.0版、模拟银行ATM机、石头剪刀布游戏、小学生四则运算练习软件）；第5章数组，介绍数组的概念、定义及使用，一维数组和二维数组的应用，基本的排序和查找算法，字符串操作相关库函数，以及相关的案例（学籍管理系统、奇数阶魔方矩阵）；第6章指针，介绍地址与指针的概念，指针的基本运算，空指针与指向void的指针、多级指针的概念，指针与一维/二维数组的关系，指针与函数的关系（指向函数的指针、指针作为函数的参数、返回指针的函数、命令行参数与动态内存分配），以及相关的案例（寻找最长输入行、输入串模式匹配、输入验证处理、基本数据类型的存储表示）；第7章用户定制数据类型，介绍结构体、共用体的概念、定义及使用，结构体与数组指针和函数的关系，链表的概念、创建及基本操作，枚举类型，typedef自定义类型，位段与位操作，以及相关的案例（C关键词查找统计、表达式求值）；第8章文件，介绍文件的概念、作用、基本操作（打开与关闭、顺序读写、随机读写、错误检测等），以及相关的案例（英语单词测试系统、软件产权保护系统）；第9章综合案例分析，介绍利用不同数据结构进行迭代、增量开发的同学通讯录系统和骑士飞行棋游戏的完整开发过程。

本书为任课教师免费提供电子课件，包括教学用PPT、全部案例和项目的完整源代码，以方便教学者进行教学，需要者可登录清华大学出版社网站。

本书特别适合作为高等院校相关专业学生学习程序设计课程的教材，也可作为相关人员自学程序设计的教材和参考书。

本书由王一萍主编，梁伟、金梅参编。第1章、第4章、第6章、第9章由梁伟编写，第2章、第8章和附录由金梅编写，第3章、第5章、第7章由王一萍编写。全书由王一萍、梁伟统稿。

在本书的编写过程中，参考了其他大量的同类教材和网络上的相关资源，在此向其作者表示衷心的感谢。

由于编者水平有限，书中难免存在疏漏之处，恳请广大读者批评指正。

作　者

2011年7月

目　录

第 1 章

C 语言程序设计导引

【教学目标】

- 理解什么是计算机。
- 理解什么是程序。
- 理解程序设计语言。
- 掌握 C 语言的相关知识。
- 掌握 HelloWorld 程序的开发过程。
- 掌握 C 语言中基本的输入/输出手段。
- 掌握 C 程序的基本结构。

1.1 程序与程序设计语言

1.1.1 计算机与程序

人与世界的关系是人类永恒的主题，也是全部的主题。人类的一切活动皆是为了能在这个世界更好地生存与发展。在漫长的历史演化中，人类逐渐摸索出了两条行之有效的生存和发展之道：一是内部团结协作，使人类作为一个整体更为强大，从而更能适应这个世界，由此诞生和演化出了社会科学；二是创造各种各样的工具，用于模拟和扩展人的某方面的功能，使单个的人更为强大，甚至是某种意义上的“超人”，从而更能适应这个世界，由此诞生和演化出了自然科学。

世界有三大支柱（也称三大资源），即物质、能量和信息。同时，世界又有两大维度，即时间和空间。人类为了能在这个世界生存和发展，创造了各种各样的工具。这些工具主要用于模拟和扩展人的各方面的功能，如肢体、感官等。同时，工具也用于帮助人类突破时间和空间的限制，从而获得更多的自由。例如，交通工具和通信网络技术等已使地球成为了“地球村”；人类借助工具已经能够上天入地；能够感知和认识的世界也拓展到从几十亿光年的宇宙尺度范围到极其微小的各种粒子层次范围。

任何工具的顶级抽象都可归纳为一个数学函数：f(I)= O。其中，I 代表输入（Input），O 代表输出（Output），f 代表处理（function）。即任何工具都是对输入的对象进行某种处理，然后输出。不同工具间的区别就在于它们能够处理的输入对象和处理过程不同。有的工具输入和处理物质，如洗衣机将脏衣服输入，经过清洗、漂洗、甩干后输出干净的衣服，

抽水机将水从 A 处抽到 B 处等；有的工具输入和处理能量，如水电站将水的势能转变为电能，电池将化学能转变为电能，电灯将电能再转变为光能、热能等；有的工具输入和处理信息，如电话机、电报机、电视机、计算机等。

计算机就是这样一种工具，它用于模拟和扩展人的大脑的功能，帮助人类处理信息。人类对信息的处理都是为了更好地认识和改造这个世界。计算机提供了这样一种可能，将世界进行数字化建模，即构建数字化的世界，亦即人化世界。因此，从本质上看，计算机的功能，亦即程序的本质功能，就是为了模拟这个世界。模拟世界的目的就是为了更好地认识和改造这个世界，从而达到人类更好地在这个世界生存和发展的根本目标。

人类对信息处理的需求主要概括为：信息的获取、信息的存储、信息的传播、信息的智能处理，以及信息的表现。

信息的表现形式多种多样，有数值、文本、图形图像、音频视频、知识情感等。信息受它本身的特质、时间空间以及人类对信息处理的不同需求的制约，可以分为如下几类。

（1）简单信息和复杂信息。

（2）少量信息和大量信息。

（3）集中信息和分布信息。

（4）固定信息和移动信息。

（5）公开信息和机密信息。

（6）临时信息和永久信息。

针对不同的信息需求，人们开发了不同的信息处理技术，如采集技术、存储技术、通信技术、安全技术、网络技术、智能技术、表现技术等。

计算机的发展，本质是它能输入并处理的数据对象的发展。如今的计算机发展已经经历并将继续经历数值计算、文本处理、图形图像处理、音频视频处理、知识情感处理等。计算机发展的终极目标就是成为“人”，甚至是“超人”，如图 1-1 所示。

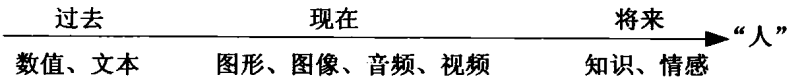


图 1-1　计算机的发展示意图

更进一步，人就是计算机，并且是高级的计算机。在人类和计算机的发展历史上，将逐渐经历人的机器化、机器的人化、人机融合。

计算机世界就是现实世界的映射，任何计算机技术均来源于现实世界。在学习计算机相关技术时，要特别注意“以人为本”的学习和思考方法。

计算机在存储程序（Stored Program）的控制下，对输入的原始信息（即数据（Data））进行处理，然后得到人们感兴趣的有用的信息（Information）并加以利用。

现代计算机几乎都采用冯·诺依曼的存储程序体系结构。它有 5 大部件：输入设备、输出设备、存储设备、运算器和控制器。其中运算器和控制器合称为中央处理单元，即 CPU。计算机在组织结构上非常类似于生产性的企业：输入设备相当于原材料的采购部门，输出设备相当于最后产品的销售部门，存储设备相当于企业的仓库，运算器相当于生产车间，

控制器相当于企业的经营管理部门。在工作流程上，计算机也和生产性的企业非常类似：生产性的企业在经营管理部门（控制器）的指导控制下，通过采购部门（输入设备）采购相应的原材料（待加工的数据），存储在仓库（存储器）中，生产开始时将原材料从仓库（存储器）中取出送至生产车间（运算器）进行加工生产，生产的半成品或最终成品又临时存储在仓库（存储器）中，最后将成品交由销售部门（输出设备）进行销售，从而完成整个生产流程。

计算机唯一的功能就是执行程序。什么是程序？如何执行程序？如何开发程序？整个计算机学科就围绕这 3 个问题展开。计算机领域的应用技术人员、工程师和科学家等都会在不同的时机、层次和侧面接触、认识和深入钻研这 3 个最为基本的问题。同时，对学习和研究计算机的不同层次的所有人来说，这 3 个问题也是他们最为根本的学习和研究指南。

任何程序都是用某种程序设计语言编写的用于解决某个实际问题的逻辑处理序列，它对特定的数据集进行处理。程序设计就是针对要解决的实际问题进行分析、设计并采用某种特定的程序设计语言编写程序来模拟和解决之。简而言之，程序就是对数据的输入、处理和输出的逻辑控制序列，它通过对某个特定问题的模拟来帮助人们认识和解决该问题。有科学家提出：程序=数据结构+算法。复杂的程序系统更涉及程序本身的组织结构，也称为系统架构。因此，程序设计存在三要素：（1）数据结构的分析和设计；（2）算法（即处理逻辑）的分析和设计；（3）系统架构的分析和设计。

1.1.2　程序设计语言

语言是一个符号系统，用于描述客观世界，并将真实世界的对象及其关系符号化，用于帮助人们更好地认识和改造世界并且便于人们之间的相互交流。在全球范围内，人类拥有数以千计的不同语言，如汉语、英语、俄语、法语、日语、韩语等。这些不同的语言，体现了不同的国家和民族对这个世界不同的认识方法、角度、深度和广度等。

计算机领域的程序设计语言用于将客观世界的对象及其关系（称为问题空间）通过逻辑等价映射为计算机世界的对象及其关系（称为解空间）。由问题空间到解空间的这个映射过程就称为程序设计，如图 1-2 所示。

问题空间 <·····> 程序设计 <·····> 解空间
（客观世界的对象及其关系）　（逻辑等价映射）　（计算机世界的对象及其关系）

图 1-2　程序设计的本质示意图

计算机中存在多种不同的程序设计语言，它们体现了在不同的抽象层次上对计算机这个客观世界的认识。

处于最底层的是机器语言，它用二进制 0、1 代码表示待处理的数据对象和处理的逻辑序列，是计算机硬件能够直接识别并执行的唯一语言。不同的机器平台拥有不同的机器语言。用机器语言编写的程序运行效率高，代码短小精悍，但学习、开发和维护极其困难。

在机器语言之上是汇编语言，它是对机器语言的简单抽象，是符号化的语言。它用地址、指令等概念来抽象待处理的数据对象和处理的逻辑序列，不能被计算机直接执行，必

须先经过汇编程序将符号转化为对应的0、1二进制代码（这个过程称为汇编），即机器语言，然后才能被执行。汇编语言的指令和机器语言的指令一一对应，不同的机器平台拥有不同的汇编语言。与机器语言相比，汇编语言提升了程序的开发效率和易维护性，执行效率和代码量上也非常接近机器语言，但仍然难于学习和掌握。

在汇编语言之上是高级语言，如C语言、Java语言等。它们是对机器语言和汇编语言更高一级的抽象，使用常量、变量、数据结构、运算符、表达式、语句、流程控制、函数、类与对象等概念来抽象待处理的数据对象和处理的逻辑序列。与汇编语言类似，用高级语言编写的程序也不能直接被计算机硬件识别和执行，必须经过编译或解释程序将它们转化为逻辑上等价的机器语言程序，然后才能被计算机硬件直接识别和执行（注意：有的编译器可以先编译成汇编语言程序，然后再通过汇编链接成机器语言程序）。高级语言更接近人类的自然语言，非常容易理解和掌握，也极大地提升了程序的开发效率和易维护性，但降低了程序的运行效率，增加了程序的代码量。

随着计算机技术的不断发展，计算机硬件的性能越来越高，存储容量越来越大，成本却越来越低。除非特别需要的情况，一般而言，程序的运行效率和程序耗费的存储容量已不再是主要考虑的问题，人们更关心程序开发效率的提升、程序的易维护性以及学习和掌握该语言的难易程度等。因此，各种高级语言应用越来越广，而汇编语言只应用在极少数对运行性能要求极高的场合。

程序设计语言的实现机制如表1-1所示。

表1-1　程序设计语言的实现机制

程序设计语言类别	语言实现机制
机器语言	由硬件直接识别并执行
汇编语言	经过汇编转化为机器语言执行
高级语言	经过编译或解释转化为机器语言执行

任何高级程序设计语言均会提供：

- 对数据及其结构的描述手段。涉及常量、变量、数据类型（数组、结构体、指针、共用体、文件）、类型转换、数据结构等重要概念。
- 对处理逻辑（核心是算法）的描述手段。涉及运算符、表达式、语句、流程控制、算法、函数或子程序、出错处理等重要概念。
- 对系统架构的描述手段，即程序系统的组织结构。涉及函数或子程序（面向过程的模块化程序设计）、类与对象（面向对象的程序设计）、设计模式及架构模式等重要概念。

高级语言编写的程序运行过程如下：

高级语言程序→编译或解释→汇编语言程序→汇编链接→机器语言程序→硬件运行。

软件是程序、程序操作的数据以及相关的文档资料的总称，主要用于解决实际问题，即模拟某个具体的小的客观世界。完整的软件开发流程如下：

捕获问题需求→分析、设计→代码实现（编写源程序）→编辑源程序→编译→汇编链接→测试、调试→运行、维护。

1.1.3　程序开发过程

程序用于解决客观世界的问题，其开发要经历捕获问题、分析设计、编码实现、测试调试、运行维护等几个主要阶段。

（1）捕获问题，也称为需求分析。此阶段的任务是深入掌握需要解决的问题是什么，有哪些要求，如性能上的、功能上的、安全性方面的要求等。问题如果比较复杂，正确认识问题本身并不是一件可以一蹴而就的事，需要反复地迭代，不断地加深对问题本身的认识。

（2）分析设计：明确需求后，就可以进行设计了，主要是确定程序所需的数据结构、核心的处理逻辑（即算法）、程序的整体架构（有哪些部分、各部分间的关联、整体的工作流程）。

（3）编码实现：就是用某种具体的程序设计语言，如 C 语言，来编程实现已经完成的设计。

（4）测试调试：包括两个方面，即测试和调试。当程序已经初步开发完成，可以运行时，为了找出其中可能出现的错误，使程序更加健壮，需要进行大量、反复的试运行，这一过程便称为测试。需要注意的是，测试只能发现尽可能多的错误，而不能发现所有的错误，但测试越早、越充分，以后付出的代价就越小。调试是指当发现程序虽然能够运行，但逻辑上并未完成预定的功能，而进行的让程序受控地运行，从而可以利用多种手段来定位错误并修正错误的过程。

（5）运行维护：当程序通过测试，达到各项设计指标的要求后，就可以获准投入运行。在运行的过程中，因可能出现的新的错误、新的需求变化（需要增加或更改程序的某些功能、需要增强程序某方面的性能等）而进行的补充开发和修正完善，称为维护。

程序开发的以上几个主要阶段，由软件团队中的不同角色来完成——项目管理者、需求分析人员、系统架构师、设计人员、编码人员、测试人员和运行维护人员等。学习程序设计者可以在上述各个阶段对应的角色中找到相应的工作机会。有趣的是，从事软件开发的人员由底层到高层的进步过程恰恰与程序开发过程相反：初始时往往从测试人员和运行维护人员做起，然后逐渐经历编码人员、设计人员，到系统架构师、需求分析人员，再到项目管理者的过程。

在软件开发中，如何组织和遵守上述程序开发的基本流程，称为软件开发模式。在实际的软件开发过程中，由于需求的不明确、需求的变化、设计的不完善、系统的复杂性、开发任务的时间、人力物力等资源限制，并不纯粹线性地遵守上述流程，而往往更多地采用迭代式、增量式的开发过程。在后面的章节中，我们将用实例演示迭代、增量式的开发过程。

1.2　C 语言相关知识概述

1.2.1　C 语言的历史沿革

C 语言于 20 世纪 70 年代初问世。它源于 UNIX 操作系统，最初只是用于改写用汇编语言编写的 UNIX 操作系统。为了将 UNIX 操作系统更大范围地进行推广，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》，这标志着 C 语言的正式诞生。

1978 年 Brian W.Kernighan 和 Dennis M.Ritchie 出版了经典的 C 语言教材 *The C Programming Language*，有人称之为《K&R》标准，从而使 C 语言逐渐成为目前世界上流行最广泛的高级程序设计语言。后来美国国家标准学会（American National Standards Institute, ANSI）在此基础上制定了一个 C 语言标准，于 1983 年发表，通常称之为 ANSI C 或标准 C。

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，使得这些 C 语言之间出现了一些不一致的地方。为了改变这种情况，ANSIC 语言制定了一套 ANSI 标准，并于 1989 年通过，1990 年正式颁布，称为 C89 或 C90 标准。

1999 年，最新的 C 语言标准颁布，称为 C99 标准。它是对 C89/C90 标准的进一步完善和发展，但到目前为止，很多 C 语言编译器并不完全支持 C99 标准的全部特性。

从诞生到现在，几十年过去了，C 语言的影响越来越深远。例如，当前处于统治地位的三大操作系统——Windows、Linux 和 UNIX 的绝大多数代码都是用 C/C++程序开发的；C 语言的应用领域极广，从上层应用程序到底层操作系统，再到各种嵌入式应用等，几乎无处不在；以 C 语言为基础，相继诞生了 C++、Java 和 C#语言，这 3 种语言都逐渐成为应用最多的前几种语言之一。这种趋势还在不断的演化中。

1.2.2 C 语言的特点

C 语言具有如下特点：

（1）语言简洁、灵活。

（2）运算符类别丰富。

（3）数据类型丰富，能够支持各种复杂的数据结构。

（4）具有结构化的流程控制语句，支持模块化的分析设计，适合编写各种不同层次的程序系统，如各种应用程序、各种操作系统、数据库管理系统等。

（5）语法限制不太严格，程序书写灵活方便。

（6）允许直接访问物理地址，能进行位操作，可直接对硬件进行操作，从而可实现汇编语言的大部分功能，兼有高级和低级语言的特点。

（7）目标代码质量高，程序执行效率高。经过编译器优化后生成的代码效率接近汇编语言代码。

（8）与汇编语言相比，程序可移植性好。

1.3 C 语言程序设计快速入门

1.3.1 HelloWorld 程序开发过程

下面将开发我们的第一个 C 语言程序。

【例 1-1】 第一个简单的 C 程序。

```
#include <stdio.h>

int main(int argc, char *argv[])
{
        printf("Hello,world!\n");

        return 0;
}
```

对于初学者来说，现在无须理解上面的代码，只要利用相应的开发工具软件将该程序输入到计算机并运行即可。通过后面的学习，一定会慢慢地理解和掌握它。

学习程序设计时最有效的方法不是对什么都刨根问底，把遇到的每一点都弄明白，而是应该先不求甚解，努力实践，把它做出来，然后再追究为什么这么做。这样的过程可能充满疑惑，甚至可以说是跌跌撞撞的，但这非常重要！正是在跌跌撞撞的过程中，你才能切身体会更深，发现更多疑问，激发你主动分析问题和解决问题的热情，从而能主动地自主学习，收获更多、更大。学习应该讲究水到渠成，而不要做崂山道士，费力不讨好，因为崂山道士式的学习会打击你学习的兴趣和积极性，导致你坚持不了多久，最终以失败收场。严格来说，程序设计并不完全是科学，它更应该是工程。工程最大的特点就是重复性，只要你积累足够的实践经验，就能掌握并且可以达到熟能生巧的境界。所以，学习程序设计一定要大量地实践。记住，程序设计“无他，惟手熟尔”!

在进入正式的开发步骤之前，应先做好相应的准备工作——安装和配置好相应的开发环境。本教材中选用了开源的 Dev-C++作为开发环境。它可以严格地支持 C99 标准的全部特性，支持中英文界面的选择，支持图形化菜单方式的开发调试，另外它还可以外挂各种工具程序。

Dev-C++的安装、配置非常简单，具体步骤如下。

（1）利用百度等搜索引擎直接搜索关键字 Dev-C++，找到相应的下载地址并下载。Dev-C++非常短小精悍，只有大约 10MB 左右。

（2）直接双击 Dev-C++的安装文件，开始安装，然后逐步往下即可（其中一步是选择中文或英文界面，推荐初学者选用中文界面），直至安装完毕。

（3）配置简单的外挂程序。为了在后面的学习开发中便于使用命令行的编译、运行命令，此处在“工具”菜单中添加了一个名为“DOS 窗口”的外挂工具选项。配置步骤如下：

① 启动 Dev-C++，选择“工具”→“配置工具”命令，如图 1-3 所示。

② 在弹出的“工具配置”对话框中单击 “添加”按钮，如图 1-4 所示。

③ 按照图 1-5 所示输入如下的内容：

- 标题：直接输入“DOS 窗口”。
- 程序：单击右侧的“浏览”按钮，在弹出的对话框中选择 C:\Windows\system32 目录下的 cmd.exe 文件，单击“确定”按钮；也可在文本框中直接输入“C:\Windows\system32\cmd.exe”。

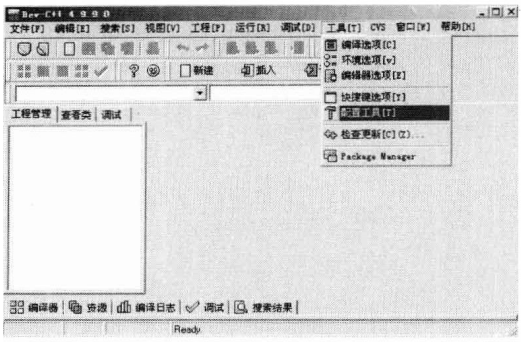


图 1-3　Dev-C++中配置 DOS 命令行步骤图示 1

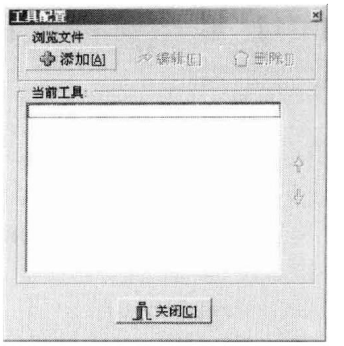


图 1-4　Dev-C++中配置 DOS 命令行步骤图示 2

- 工作路径：首先将光标定位在“工作路径”文本框中（如果有内容则先直接清除），然后在左下角的“可用的宏”列表框中选择 <PROJECTPATH>选项，单击“插入宏”按钮即可；也可直接输入“<PROJECTPATH>”。

提示：工作路径配置为<PROJECTPATH>，可以保证 DOS 命令行窗口在启动时自行进入到源程序文件所在的目录。

④ 最后单击“确定”按钮，即可完成配置，如图 1-5 所示。

以后如果需要用到 DOS 命令行，直接在 Dev-C++窗口中选择“工具”→“DOS 窗口”命令即可。

有了相应的开发环境后，就可以进入正式的程序开发阶段了。一般来说，程序编写好后，需要执行以下几步才能得到输出结果。

（1）输入源程序，保存为.c 的源程序文件。

（2）编译。

（3）链接。

提示：步骤（2）和（3）在集成开发环境下通常自动合成一步完成。

（4）运行程序。

具体操作步骤如下：

（1）启动 Dev-C++，选择 “文件”→“新建”→“工程”命令，如图 1-6 所示。

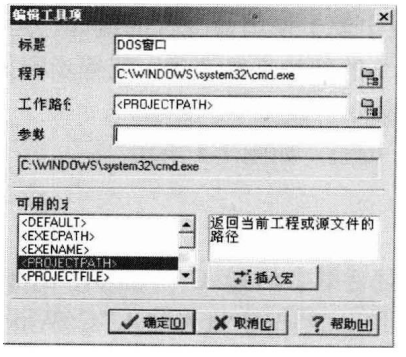


图 1-5　在 Dev-C++中配置 DOS 命令行步骤图示 3

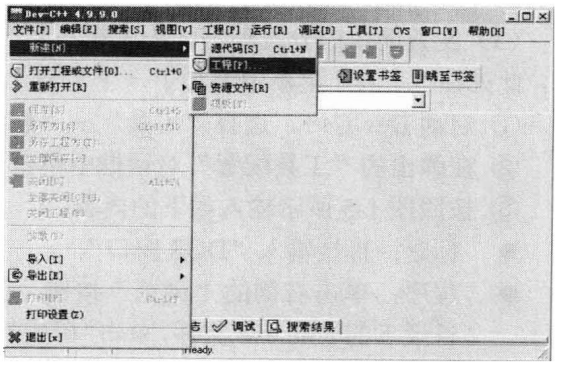


图 1-6　在 Dev-C++环境下启动新建工程示意图 1

（2）在弹出的“新工程”对话框中选中“C 工程”单选按钮，工程类型选择为 Console Application（控制台应用程序，即命令行字符界面程序），并输入工程名称“firstp”（工程名称一般为小写，可以自己命名，也可为汉字），然后单击“确定”按钮，如图 1-7 所示。

（3）弹出 Create new project 对话框，从中选择适当的保存位置（对于初学者，一般推荐保存在桌面上，这样容易查找），然后单击“保存”按钮，如图 1-8 所示。

图 1-7　在 Dev-C++环境下启动新建工程示意图 2

图 1-8　在 Dev-C++环境下保存工程示意图

（4）此时在 Dev-C++窗口中可以看到，系统自动生成了一个名为 firstp 的工程。单击工程名 firstp 前面的“+”图标，在展开的工程文件中双击 main.c，即可在右边的代码窗口中输入程序源码（Dev-C++自动帮助用户生成了程序源码框架，可以在此基础上改写代码，也可以将它们全部删除后再从零写起），如图 1-9 所示。

（5）输入完程序代码后，选择“运行”→“编译”命令，如图 1-10 所示。

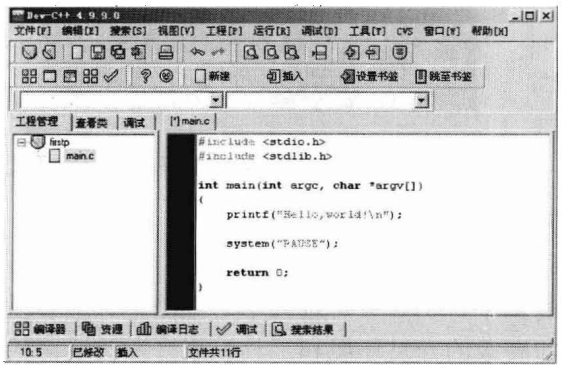


图 1-9　在工程中新建一个 main.c 文件示意图

图 1-10　Dev-C++环境下编译源程序示意图

（6）由于源程序文件 main.c 未保存，所以弹出“保存文件”对话框。在该对话框中将文件名改为 HelloWorld.c，然后单击“保存”按钮，如图 1-11 所示。

（7）编译成功后，单击“关闭”按钮返回，如图 1-12 所示。如果输入的代码有误，则会提示相应的出错信息。此时应该返回重新修改代码，然后再次编译，直到编译无误时为止。