

## <<C语言程序设计与实践>>

### 图书基本信息

书名：<<C语言程序设计与实践>>

13位ISBN编号：9787111310075

10位ISBN编号：7111310071

出版时间：2010-9

出版时间：机械工业

作者：凌云//吴海燕//谢满德

页数：292

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

## &lt;&lt;C语言程序设计与实践&gt;&gt;

## 前言

“C语言程序设计”是一门理论与工程实践密切相关的专业基础课程，在计算机学科的教学具有十分重要的作用。

大力加强该课程的建设，提高该课程的教学质量，有利于教学改革和教育创新，有利于创新人才的培养。

学生通过本课程的学习，可以培养良好的编程风格，掌握常见的算法思路，真正提高运用C语言编程解决实际问题的综合能力，为后续课程的实践环节的教学打好基础。

目前国内关于C语言的教材较多，有些教材语法知识介绍细致，较适合作为非计算机专业的等级考试类教学用书；有些教材起点较高，内容深奥，不适用于初学者。

为了帮助广大学生更好地掌握C语言编程技术，我们组织浙江工商大学C语言程序设计课程组教师进行了深入的讨论和研究，针对学生学科竞赛和课时压缩的背景，将该课程的建设与其他信息类专业的课程体系改革相结合，发挥学院在计算机、电子商务和信息管理等专业上的办学优势，编写了本书。

全书以程序设计为主线，采用了渐进式的体系结构，在详细阐述程序设计基本概念、原理和方法的基础上，结合实践教学和学科竞赛的实际情况，通过大量经典实例讲解和实训，使学习者掌握利用C语言进行结构化程序设计的技术和方法，培养和他们的实践动手能力和创新协作精神。

本书的框架结构分为三个部分。

第一部分包括第1~11章，介绍C语言的基础语法知识，这部分内容按C语言的知识点循序渐进地介绍，同时，针对C语言中的重点和难点，如指针，精心设计了丰富的实例，用了大量的篇幅从不同方面对其进行讲解，帮助读者理解并掌握这些重点和难点。

第二部分包括第12~13章，为项目实训和常用算法指导部分，这部分通过项目开发全过程的全方位指导，从需求分析、算法设计到程序编写和过程调试，以项目实训的形式引导和帮助学生解决实际问题，提高学生解决具体问题的能力，并对程序设计竞赛中的常见算法及其应用进行了介绍。

第三部分即第14章，介绍编程风格与程序调试方法。

## <<C语言程序设计与实践>>

### 内容概要

《C语言程序设计与实践》由浅入深地讲授了C语言程序设计的技术与技巧。首先，介绍了c语言的基础语法知识；其次，通过项目开发全过程的全方位指导，从需求分析、算法设计到程序编写和过程调试，以项目实训的形式引导和帮助学生解决实际问题，提高学生解决具体问题的能力，并对程序设计竞赛中的常见算法及其应用进行了介绍；最后，介绍了编程风格与程序调试方法。

《C语言程序设计与实践》内容齐备、自成一体，可作为计算机及相关专业的本科或专科教材，也可以作为信息类或其他相关专业的辅助教材。

## &lt;&lt;C语言程序设计与实践&gt;&gt;

## 书籍目录

|                    |                          |                       |                        |                         |
|--------------------|--------------------------|-----------------------|------------------------|-------------------------|
| 出版者的话              | 前言                       | 教学建议                  | 第1章 C语言与程序设计概述         | 1.1 初见C语言程序             |
| 1.2 计算机与程序设计       | 1.2.1 指令与程序              | 1.2.2 程序与程序设计         | 1.2.3 程序设计和程序设计语言      | 1.2.4 程序设计和程序设计语言       |
| 1.2.4 程序设计与程序设计语言  | 1.3 C语言学习与自然语言学习的关系      | 1.4 C语言的发展历史和现状       | 1.4.1 C语言的发展历史和现状      | 1.4.2 C语言的特点            |
| 1.4.1 C语言的发展历史和现状  | 1.4.2 C语言的特点             | 习题                    | 第2章 例子驱动的C语言语法元素概览     | 2.1 变量与表达式              |
| 2.1 变量与表达式         | 2.2 分支语句                 | 2.2.1 if语句            | 2.2.2 switch语句         | 2.3 循环语句                |
| 2.3.1 while循环语句    | 2.3.2 for循环语句            | 2.4 符号常量              | 2.5 输入输出               | 2.6 数组                  |
| 2.7 函数             | 2.8 算法                   | 2.8.1 算法的概念           | 2.8.2 流程图与算法描述         | 习题                      |
| 第3章 基本数据类型和表达式     | 3.1 基本语法单位               | 3.1.1 基本符号            | 3.1.2 关键字              | 3.1.3 标识符               |
| 3.2 数据类型           | 3.3 常量与变量                | 3.3.1 常量              | 3.3.2 变量               | 3.3.3 变量的初始化            |
| 3.4 表达式和运算符        | 3.4.1 算术运算符              | 3.4.2 赋值运算符           | 3.4.3 关系运算符            | 3.4.4 逻辑运算符             |
| 3.4.5 位运算符         | 3.4.6 逗号运算符              | 3.4.7 条件运算符           | 3.5 各类数值型数据间的混合运算      | 习题                      |
| 第4章 输入输出语句         | 4.1 putchar函数            | 4.2 printf函数          | 4.2.1 printf函数的格式      | 4.2.2 格式说明字符            |
| 4.3 getchar函数      | 4.4 scanf函数              | 4.4.1 一般形式            | 4.4.2 格式说明             | 4.4.3 scanf函数的执行中应注意的问题 |
| 4.5 程序举例           | 习题                       | 第5章 C程序结构             | 5.1 C语句                | 5.2 程序设计基础              |
| 5.3 结构化程序设计的三种基本结构 | 5.3.1 顺序结构               | 5.3.2 选择结构            | 5.3.3 循环结构             | 5.4 if分支语句              |
| 5.4.1 第一种if语句形式    | 5.4.2 第二种if语句形式          | 5.4.3 第三种if语句形式       | 5.4.4 if语句的嵌套          | 5.4.5 程序举例              |
| 5.5 switch分支语句     | 5.6 while循环语句            | 5.7 do?while循环语句      | 5.8 for循环语句            | 5.9 break语句和continue语句  |
| 5.9.1 break语句      | 5.9.2 continue语句         | 5.10 多重循环的嵌套          | 5.11 程序举例              | 习题                      |
| 第6章 数组             | 6.1 一维数组                 | 6.1.1 一维数组的定义         | 6.1.2 一维数组元素的引用        | 6.1.3 一维数组元素的初始化        |
| 6.2 二维数组           | 6.2.1 双下标变量              | 6.2.2 二维数组及其定义        | 6.2.3 二维数组的初始化         | 6.2.4 二维数组应用举例          |
| 6.3 综合应用举例         | 6.4 字符数组                 | 6.4.1 字符串和字符串结束标志     | 6.4.2 字符数组的输入输出        | 6.4.3 字符串函数             |
| 6.4.4 二维字符数组       | 6.4.5 字符数组应用举例           | 习题                    | 第7章 函数                 | 7.1 函数的定义               |
| 7.1.1 函数定义的格式      | 7.1.2 形式参数和实际参数          | 7.1.3 函数的返回值          | 7.2 函数的一般调用            | 7.2.1 函数调用的形式           |
| 7.2.2 函数调用的方式      | 7.2.3 主调函数和被调函数的相对位置关系   | 7.2.4 函数调用时值的单向传递性    | 7.2.5 函数调用示例           | 7.3 函数的嵌套调用             |
| 7.4 函数的递归调用        | 7.4.1 概述                 | 7.4.2 函数的递归调用应用举例     | 7.5 数组作为函数的参数          | 7.5.1 数组元素作为函数的实参       |
| 7.5.2 数组名作为函数的参数   | 7.5.3 多维数组作为函数的参数        | 7.6 变量的作用域——局部变量和全局变量 | 7.6.1 局部变量             | 7.6.2 全局变量              |
| 7.7 变量的存储类别和生存期    | 7.7.1 变量的存储类别            | 7.7.2 动态变量            | 7.7.3 静态变量             | 7.7.4 外部变量              |
| 7.8 内部函数和外部函数      | 7.8.1 内部函数               | 7.8.2 外部函数            | 习题                     | 第8章 编译预处理               |
| 8.1 宏定义            | 8.1.1 不带参数的宏定义           | 8.1.2 带参数的宏定义         | 8.2 文件包含               | 8.3 条件编译                |
| 8.3.1 条件编译语句1      | 8.3.2 条件编译语句2            | 8.3.3 条件编译语句3         | 习题                     | 第9章 指针                  |
| 9.1 地址和指针的概念       | 9.2 指针变量和地址运算符           | 9.2.1 指针变量的定义         | 9.2.2 指针变量的使用          | 9.3 指针和数组               |
| 9.3.1 通过指针存取数组元素   | 9.3.2 字符串和指针             | 9.4 指针和函数             | 9.4.1 指针作为函数的参数        | 9.4.2 指针作为函数的返回值        |
| 9.4.3 指向函数的指针      | 9.5 多级指针                 | 9.5.1 多级指针的概念和使用      | 9.5.2 多级指针和多维数组        | 9.5.3 命令行参数             |
| 9.6 指针和动态存储管理      | 9.6.1 概述                 | 9.6.2 malloc函数和free函数 | 9.6.3 动态存储管理的应用        | 9.7 指针和指针运算小结           |
| 习题                 | 第10章 结构与联合               | 10.1 结构体类型变量的定义和引用    | 10.1.1 结构体类型变量的定义      | 10.1.2 结构体类型变量的引用       |
| 10.1.3 结构体类型变量的初始化 | 10.2 结构体数组的定义和引用         | 10.3 结构体指针的定义和引用      | 10.3.1 指向结构体类型变量的指针的使用 | 10.3.2 指向结构体类型数组的指针的使用  |
| 10.4 链表的定义和操作      | 10.4.1 概述                | 10.4.2 链表的建立          | 10.4.3 输出链表元素          | 10.4.4 删除链表元素           |
| 10.4.5 插入链表元素      | 10.4.6 查询链表元素            | 10.5 联合               | 10.5.1 联合的定义           | 10.5.2 联合成员的使用          |
| 10.5.3 应用举例        | 10.5.4 数组、结构和联合三种数据类型的比较 | 10.6 枚举类型             | 10.7 用typedef定义类型名     | 习题                      |

<<C语言程序设计与实践>>

|           |              |           |             |                               |             |           |             |                   |              |              |              |                     |                  |              |              |                      |              |    |           |                      |             |             |               |                        |                |                  |             |                |               |                    |                |                      |             |             |                   |                    |                |                           |             |             |                    |                    |           |                 |                 |                       |             |                |             |                      |             |                |             |                          |             |                |             |                        |             |                |             |                        |             |                |             |                     |             |                |             |                        |             |                |             |                   |             |                |             |              |                  |                        |                |               |           |                |             |              |                    |                   |                |                |               |                     |             |             |            |    |               |      |
|-----------|--------------|-----------|-------------|-------------------------------|-------------|-----------|-------------|-------------------|--------------|--------------|--------------|---------------------|------------------|--------------|--------------|----------------------|--------------|----|-----------|----------------------|-------------|-------------|---------------|------------------------|----------------|------------------|-------------|----------------|---------------|--------------------|----------------|----------------------|-------------|-------------|-------------------|--------------------|----------------|---------------------------|-------------|-------------|--------------------|--------------------|-----------|-----------------|-----------------|-----------------------|-------------|----------------|-------------|----------------------|-------------|----------------|-------------|--------------------------|-------------|----------------|-------------|------------------------|-------------|----------------|-------------|------------------------|-------------|----------------|-------------|---------------------|-------------|----------------|-------------|------------------------|-------------|----------------|-------------|-------------------|-------------|----------------|-------------|--------------|------------------|------------------------|----------------|---------------|-----------|----------------|-------------|--------------|--------------------|-------------------|----------------|----------------|---------------|---------------------|-------------|-------------|------------|----|---------------|------|
| 第11章 文件操作 | 11.1 文件的基本概念 | 11.1.1 概述 | 11.1.2 文件分类 | 11.1.3 缓冲文件系<br>统和非缓冲文件系<br>统 | 11.1.4 流式文件 | 11.2 标准文件 | 11.3 文件类型指针 | 11.4 文件的打开<br>与关闭 | 11.4.1 文件的打开 | 11.4.2 文件的关闭 | 11.5 文件的顺序读写 | 11.6 常用文件顺序读<br>写函数 | 11.7 文件顺序读写的应用举例 | 11.8 文件的随机读写 | 11.8.1 文件的定位 | 11.8.2 文<br>件操作的出错检测 | 11.9 非缓冲文件系统 | 习题 | 第12章 综合实训 | 12.1 综合实训一<br>：24点程序 | 12.1.1 问题描述 | 12.1.2 问题分析 | 12.1.3 数据结构分析 | 12.1.4 程序执行流程和<br>设计分析 | 12.1.5 程序运行和测试 | 12.2 综合实训二：五子棋游戏 | 12.2.1 问题描述 | 12.2.2<br>问题分析 | 12.2.3 数据结构分析 | 12.2.4 程序执行流程和设计分析 | 12.2.5 程序运行和测试 | 12.3 综合实训三：通用的管理信息系统 | 12.3.1 问题描述 | 12.3.2 问题分析 | 12.3.3 数据结构<br>分析 | 12.3.4 程序执行流程和设计分析 | 12.3.5 程序运行和测试 | 12.4 综合实训四：BASIC程序<br>解释器 | 12.4.1 问题描述 | 12.4.2 问题分析 | 12.4.3 数据结构和程序设计分析 | 12.4.4 程序运行<br>和测试 | 12.4.5 讨论 | 第13章 初涉ACM/ICPC | 13.1 ACM/ICPC概述 | 13.2 小数近似值问<br>题与枚举算法 | 13.2.1 问题描述 | 13.2.2 问题分析与求解 | 13.2.3 问题小结 | 13.3 迷宫问题与<br>深度优先搜索 | 13.3.1 问题描述 | 13.3.2 问题分析与求解 | 13.3.3 问题小结 | 13.4 经典01背包问<br>题与动态规划算法 | 13.4.1 问题描述 | 13.4.2 问题分析与求解 | 13.4.3 问题小结 | 13.5 经典部<br>分背包问题与贪心算法 | 13.5.1 问题描述 | 13.5.2 问题分析与求解 | 13.5.3 问题小结 | 13.6 关<br>系序列问题与图的拓扑排序 | 13.6.1 问题描述 | 13.6.2 问题分析与求解 | 13.6.3 问题小结 | 13.7<br>公路建设图与最小生成树 | 13.7.1 问题描述 | 13.7.2 问题分析与求解 | 13.7.3 问题小结 | 13.8<br>青蛙跳石头与图的单源最短路径 | 13.8.1 问题描述 | 13.8.2 问题分析与求解 | 13.8.3 问题小结 | 13.9 青蛙约会之解与模线性方程 | 13.9.1 问题描述 | 13.9.2 问题分析与求解 | 13.9.3 问题小结 | 13.10 练习参考网址 | 第14章 程序设计风格与程序调试 | 14.1 程序设计风格和程序设计方<br>法 | 14.2 C语言程序设计风格 | 14.2.1 标识符的命名 | 14.2.2 注释 | 14.2.3 清晰简洁的表达 | 14.2.4 书写格式 | 14.3 结构化程序设计 | 14.3.1 自顶向下的程序设计方法 | 14.3.2 程序的模块<br>化 | 14.4 健全程序的风格标准 | 14.5 程序错误类型和调试 | 14.5.1 程序错误类型 | 14.5.2 程序<br>错误分析方法 | 14.6 程序调试方法 | 14.7 常见错误分析 | 14.8 帮助的使用 | 习题 | 附录<br>ASCII码表 | 参考文献 |
|-----------|--------------|-----------|-------------|-------------------------------|-------------|-----------|-------------|-------------------|--------------|--------------|--------------|---------------------|------------------|--------------|--------------|----------------------|--------------|----|-----------|----------------------|-------------|-------------|---------------|------------------------|----------------|------------------|-------------|----------------|---------------|--------------------|----------------|----------------------|-------------|-------------|-------------------|--------------------|----------------|---------------------------|-------------|-------------|--------------------|--------------------|-----------|-----------------|-----------------|-----------------------|-------------|----------------|-------------|----------------------|-------------|----------------|-------------|--------------------------|-------------|----------------|-------------|------------------------|-------------|----------------|-------------|------------------------|-------------|----------------|-------------|---------------------|-------------|----------------|-------------|------------------------|-------------|----------------|-------------|-------------------|-------------|----------------|-------------|--------------|------------------|------------------------|----------------|---------------|-----------|----------------|-------------|--------------|--------------------|-------------------|----------------|----------------|---------------|---------------------|-------------|-------------|------------|----|---------------|------|

## &lt;&lt;C语言程序设计与实践&gt;&gt;

## 章节摘录

程序设计是按指定要求，编排计算机能识别的特定指令组合的过程，而程序设计语言是为方便人进行程序设计而提供的一种手段，是人与计算机交流的语言，而且这种程序设计语言也在随着计算机技术的发展而不断地发展。

计算机能直接识别的是由“0”和“1”组成的二进制数，二进制是计算机语言的基础。一开始，人们只能降贵纡尊，用计算机能直接理解的语言去命令计算机工作，通过写出一串串由“0”和“1”组成的指令序列交给计算机执行。

这种语言称为机器语言。

使用机器语言编写程序是一件十分痛苦的工作，特别是在程序有错需要修改时，更是如此。而且，由于每台计算机的指令系统往往各不相同，所以，在一台计算机上执行的程序，要想在另一台计算机上执行，必须重新修改程序，造成了重复工作。

所以，现在已经很少有人用机器语言直接写程序。

为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、有一定含义的符号串来替代一个特定指令的二进制串，比如，用“ADD”表示加法，“suB”表示减法，“Mov”表示数据传递等，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，即第二代计算机语言。

然而对于计算机而言，它只认识“0”和“1”组成的指令，并不认识这些符号，这就需要有一个专门的程序，来负责将这些符号翻译成计算机能直接识别和理解的二进制数的机器语言，完成这种工作的程序被称为汇编程序，它充当的就是一个翻译者的角色。

汇编语言同样十分依赖于机器硬件，移植性不好，但效率十分高。

现代的桌面计算机，性能已经非常强大，效率已经不是首要关注目标。

所以，通常只有在资源受限的嵌入式环境或与硬件相关的程序设计时（如驱动程序），汇编语言才会作为一种首选的软件开发语言。

虽然机器语言发展到汇编语言已经有了很大的进步，但是由于每条指令完成的工作非常有限，因此编程过程仍然繁琐，语义表达仍然比较费力。

于是，人们期望有更加方便、功能更加强大的高级编程语言的出现。

这种高级语言应该接近于数学语言或人的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。

c语言就是一种能满足这种要求的语言，它由于既有高级语言的通用性又有底层语言的高效性而展示出了强大的生命力，几十年来一直被广泛应用。

许多高校也基本上将c语言当作计算机专业和相关专业的重要必修课，作为高校学生接触的第一门编程语言。

同样，计算机本身并不认识c语言程序，因此我们需要将c语言程序先翻译成汇编程序，再将汇编程序翻译成机器语言，这个过程往往由编译程序帮我们完成，而不需要我们自己来做。



## <<C语言程序设计与实践>>

### 编辑推荐

《C语言程序设计与实践》以程序设计为主线，在详细阐述程序设计基本概念、原理和方法的基础上，结合实践教学和学科竞赛的实际情况，通过经典实例讲解和实训，使学生掌握利用c语言进行结构化程序设计的技术和方法。

注重培养良好的编程风格，掌握常见的算法思路，真正提高学生运用C语言编程以解决实际问题的综合能力，为后续课程实践环节的教学打下良好基础。

《C语言程序设计与实践》特色 实例丰富。

在介绍理论知识的同时，书中还给出了大量的实例和实训内容，以提高学生的动手实践能力。

重点突出。

针对C语言语法庞杂、有些语句可以相互替代、有些语法不常使用的情况，《C语言程序设计与实践》重点介绍了基本的、常用的C语言语法知识，使学生容易理解和掌握。

融会贯通。

着重介绍程序设计语言的共性，培养学生自学其他程序设计语言的能力。

## <<C语言程序设计与实践>>

### 版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>