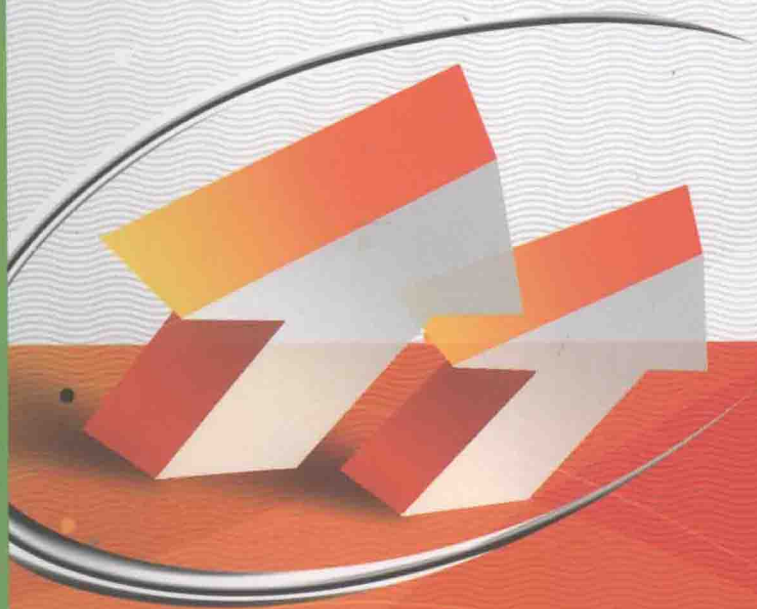
安博教育

安博教育集团职业教育标准教材

C语言基础

安博教育集团　编著



软件方向

電子工業出版社
PUBLISHING HOUSE OF ELECTRONICS INDUSTRY
http://www.phei.com.cn

安博教育集团职业教育标准教材

C语言基础

安博教育集团　编著

電子工業出版社
Publishing House of Electronics Industry
北京・BEIJING

内 容 简 介

本书深入浅出地介绍了C语言程序设计的基础知识，内容涉及C语言基础、算法基础、变量、数据类型、运算符、输入/输出相关函数、选择结构、循环结构、各种表达式、数组、字符串、指针、函数、结构体、ISO C99的扩展语法等。全书内容丰富，结构严谨，层次清晰，语言生动，论述精准而深刻，实例丰富而实用。

本书不仅适合用做计算机职业培训的首选教材，也可作为普通高校教材使用，更是C语言初学者和开发者的首选参考书。

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语言基础/安博教育集团编著. —北京：电子工业出版社，2012.2
安博教育集团职业教育标准教材

ISBN 978-7-121-15162-0

Ⅰ. ①C… Ⅱ. ①安… Ⅲ. ①C语言—程序设计—职业教育—教材 Ⅳ. ①TP312

中国版本图书馆CIP数据核字（2011）第236698号

策划编辑：关雅莉
责任编辑：郝黎明　　文字编辑：裴　杰
印　　刷：
装　　订：三河市鑫金马印装有限公司
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：787×1092　1/16　印张：13.25　字数：339.2千字　彩插：1
印　　次：2012年2月第1次印刷
定　　价：37.00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zlts@phei.com.cn，盗版侵权举报请发邮件至dbqq@phei.com.cn。

服务热线：（010）88258888。

序言

百年大计，教育为本。教育是民族振兴、社会进步的基石，是提高国民素质、促进人的全面发展的根本途径，寄托着亿万家庭对美好生活的期盼。2010 年 7 月，国务院颁发《国家中长期教育改革和发展规划纲要（2010—2020 年）》。这份《纲要》把“坚持能力为重”放在了战略主题的位置，指出教育要“优化知识结构，丰富社会实践，强化能力培养。着力提高学生的学习能力、实践能力、创新能力，教育学生学会知识技能，学会动手动脑，学会生存生活，学会做人做事，促进学生主动适应社会，开创美好未来”。这对学生的职前教育、职后培训都提出了更高的要求，需要建立和完善多层次、高质量的职业培养机制。

安博教育集团率先倡导“构建中国自己的开放式网络教育平台”，并最早实践、研创出教育部鉴定并符合国际标准的网络教育平台；同时，安博是全国信息技术标准化委员会教育技术分技术委员会的核心创建成员，是国际化软件工程高级人才培养体系、实训体系、园区型实训基地的倡导者和最早实践者。

当前，作为国内教育培训业内最大的整合者，安博教育的优势主要集中在三个方面。一是安博在通过信息化手段与教育的结合方面有着独特的理解和成功尝试。此外安博对中国教育市场有深刻理解，并且是国内少数拥有丰厚技术优势的教育公司。安博能充分整合来自国际厂商和企业人才的需求，并将传统教育资源、先进的行业技术资源与学习者的个性化需求进行有机的结合，实现了真正意义上的“教育是满足企业和行业发展需求”的终极目标。二是拥有遍布全国的实施网络和大型基地，以及大量具备企业项目实施经验和教育培训经验的优秀教师，通过高品质标准化的教育服务为其业务稳步发展起到了重要保障和促进作用。三是安博受到国家教育部及各地教育部门的大力支持和高度认可，安博是教育部 IT 实训推广工程的唯一实施单位。

安博教育服务业务以重点解决升学和就业两大关键需求为目标，为各个阶段学习者提供高效的个性化学习服务。目前，安博教育集团的业务涉及基础教育服务、职业教育服务、企业培训等领域，基地、学校、机构等已遍及全国数十个重点城市，形成了以区域教育服务中心和实训基地为依托，以师资、课程、服务流程、IT 支持、网络学习服务的标准化为载体的服务体系，通过标准品质的服务保障全国各地用户的个性化需求。

为了贯彻落实党中央、国务院关于大力发展高等职业教育、培养高等技术应用型人才的战略部署，解决高职高专院校缺乏实用性教材的问题，安博根据企业工作岗位要求和院校的教学需要，充分汲取高职高专院校在探索培养高等技术应用型人才方面取得的成功经验和教学成果，并依托安博丰厚的 IT 产业背景，坚持自主研发和强强合作的指导思想，组织编写了本套“职业教育标准教材”丛书。

在组织编写过程中，我们力求使这套教材具有以下特点：一是根据国内产业经济发展现状，加大课程体系、实训体系及自主知识产权软件产品的研发力度；二是积极引进国际先进的课程与技能资源，大力推动国际合作，实现安博教育体系与国际教育体系的接轨，实现课程无缝对接与学分互认；三是从职业（岗位）分析入手，围绕课程的教学目标，体现技能训练的针对性；四是突出教材的先进性，更多地将新技术融入其中，以期缩短学校教育与企业需要的距离，更好地满足企业用人的需要；五是贯彻以技能训练为主线、相关知识为支撑的编写思路，切实落实“管用、够用、适用”的教学指导思想。

此次出版的职业教育标准教材，是安博实训理念探索和实践的又一步，我们希望能为提升大学生的就业竞争力和就业质量尽自己的绵薄之力。

“红日初升，其道大光；河出伏流，一泻汪洋。”新的征程已经开始，安博职教将继续前行，争做中国最专业的大学生就业服务提供商！

安博职业教育运营集团　总裁
编审委员会　主席

前言

本套教材在保证知识体系完备、脉络清晰、论述精准深刻的同时，尤其注重培养读者的实际动手能力和企业岗位技能的应用能力，并结合大量的工程案例和项目来使读者更进一步灵活掌握及应用相关的技能。

- **本书内容**

下面简要地介绍本书的概貌，使读者对本书有一个提纲挈领的了解。

第 1 章，讲解 C 语言基础知识，算法的基本概念，简单的流程图绘制。

第 2 章，讲解变量、数据类型、运算符的相关知识和应用。

第 3 章，讲解 C 语言输入/输出相关函数的语法和使用。

第 4 章，讲解 if 结构、switch 结构、各种表达式的使用。

第 5 章，讲解 while 循环、do-while 循环、for 循环以及循环控制的使用。

第 6 章，讲解数组和字符串的定义和使用，字符串相关函数的使用，以及 C99 对数组的扩展。

第 7 章，讲解指针的定义和使用，如何使用指针处理字符串。

第 8 章，讲解函数的定义、参数、作用域，递归函数的实现，以及 C99 对函数的扩展。

第 9 章，讲解结构体的定义、嵌套结构、结构变量的声明和使用，以及 C99 对结构体的扩展。

- **配套教学资源**

本书提供了配套的立体化教学资源，包括教学大纲、电子教案、源代码、项目案例等配套文档，以及素材库等必需的文件，读者可以通过华信教育资源网（www.hxedu.com.cn）免费下载使用。

- **本书主编**

本书由马锋、刘伟主编。由于编者水平有限，错漏之处在所难免，请广大读者批评指正。

- **特别鸣谢**

感谢安博（大连）软件和服务外包人才实训基地、安博（昆山）服务外包人才实训基地、安博华南实训基地、安博广州金桥学校、安博大连希望学校、安博上海英豪学院、安博天津数字艺术产业基地、安博亚威科技（北京）、安博中程在线（北京）、安博长沙牛耳学校、安博河北实训基地、安博山东师创学院、安博西南实训基地的学术研究团队对本书所进行的认真的审校及建议。

主编

2011 年 9 月

目录

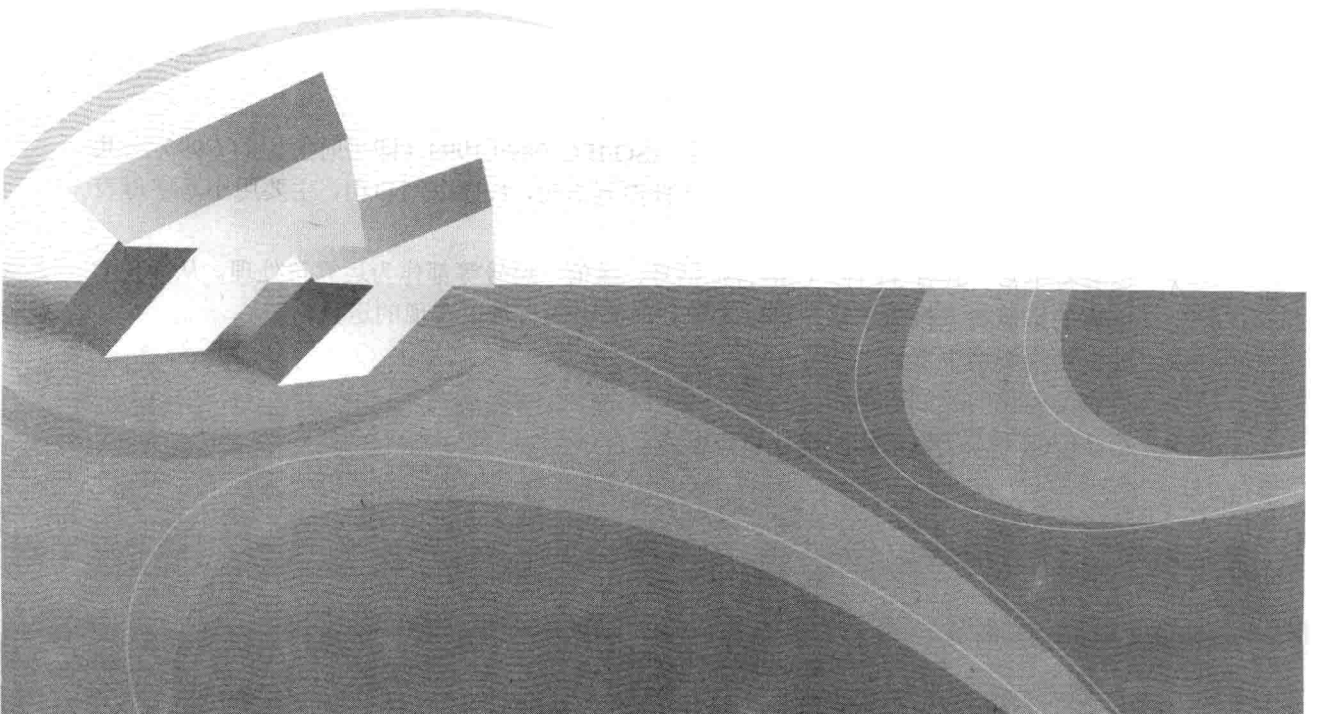
此为试读，需要完整PDF请访问：www.ertongbook.com

第1章

C 程序概述和算法

本章导读：

- 了解 C 语言的发展史
- 掌握 C 语言的特点
- 熟悉 C 程序的结构
- 熟悉 C 程序的编译、执行过程
- 理解算法，掌握算法的概念和特点
- 掌握流程图
- 理解伪代码



1.1 C 语言简介

C 语言是面向过程的高级语言，现在多用于开发操作系统内核、系统级应用、系统运行库、嵌入式设备应用，以及对运行速度要求高的应用。

1.1.1 C 语言发展历史

1946 年，第一台电子计算机问世，应用领域迅速扩大，计算机软、硬件飞速发展，程序设计语言相继问世。

1978 年由美国电话电报公司(AT&T)贝尔实验室正式发表了 C 语言。同时由 B.W.Kernighan 和 D.M.Ritchit 共同编写了著名的“THE C PROGRAMMING LANGUAGE”一书，这本书通常被称为《K&R》，也有人称之为《K&R》标准。但是，在《K&R》中并没有给出完整的 C 语言标准。后来由美国国家标准协会（American National Standards Institute）在这本书内容的基础上制定了一个 C 语言标准，于 1983 年发表，通常称为 ANSI C。ANSI C 到现在经历了若干的版本，在这些版本的推行过程中，各个编译器厂商（如微软、Borland、Intel、GNU 等）对 ANSI C 的支持也不尽相同，而且都在 ANSI C 基础之上增加了有特色的一些功能，在实际工作中如非必要则应尽量避免使用这些特色功能。目前兼容性最好、最通用的是 ISO C99，也就是通常所说的 C99。C99 是在 C89 的基础上衍生出来的，该版本在 C 语言的各个方面都进行了改进或增强，吸收了部分其他语言的特点，使 C 语言的应用领域更为广泛，开发也更为便利。C99 通常被认为在增加了很多特性之后变得更加复杂繁琐，但在初学时不用过于关心 C99 和 C89 的差异，对于 C99 中增加的实用特性或功能在本书中的对应章节会有专题描述。

1.1.2 C 语言特点

C 语言的特点如下。

- C 语言简洁、紧凑，使用方便、灵活。ISO/IEC 9899:1999（即平时所说的 C99）一共只有 37 个关键字（如表 1.1 所示）。9 种控制语句，程序书写自由，主要用小写字母表示，压缩了一切不必要的成分。
- 运算符丰富。共有 34 种。C 语言把括号、赋值、逗号等都作为运算符处理。从而使 C 语言的运算类型极为丰富，可以实现其他高级语言难以实现的运算。
- 数据结构类型丰富。
- 具有结构化的控制语句。
- 语法限制不太严格，程序设计自由度大。

- C 语言允许直接访问物理地址，能进行位（bit）操作，能实现汇编语言的大部分功能。因此有人把它称为中级语言。
- 生成目标代码质量高，程序执行效率高。
- 与汇编语言相比，用 C 语言编写的程序可移植性好。

但是，C 语言对程序员要求也高，程序员用 C 语言写程序会感到限制少、灵活性大，功能强，但较其他高级语言在学习上要困难一些。

表 1.1　C 语言关键字

auto	continue	enum	if	restrict	static	unsigned	_Complex
break	default	extern	inline	return	struct	void	_Imaginary
case	do	float	int	short	switch	volatile	
char	double	for	long	signed	typedef	while	
const	else	goto	register	sizeof	union	_Bool	

提示：

在 C99 之前的版本中有 32 个关键字，这些关键字都是小写的。

从学习角度看，C 语言还具有以下一些特点。

- 层次清晰。便于按模块化方式组织程序，易于调试和维护。
- 语句简洁，学习时入门相对容易，C 语言很好地总结了其他语言提出的程序库概念。
- 功能强大。既可用于系统软件的开发，也适合于应用软件的开发。
- 移植性好。只要对这种语言稍加修改，便可以适应不同型号机器或各类操作系统。

1.1.3　C 程序的基本结构

先来看一个简单的 C 语言程序例子。

```
#include <stdio.h>
int main(void)
{
    printf("Hello world\n");
}
```

第 1 行：“#include”为 C 语言的预处理语法，是包含的意思。“stdio.h”为头文件，头文件可以是 C 程序中现成的标准库文件，也可以是自定义的库文件。该行的作用是把标准库文件 stdio.h 包含到当前的程序中，成为当前程序的一部分。该行对于多数编译器可以忽略，同时很多编译器会提示警告。

第 2 行：“main”为固定名称，无论当前的程序代码有多长、多复杂，程序必然从这里开

始运行。

第 3 行：表示函数的开始，后面是函数的主体，也就是实现功能的程序代码。

第 4 行：在屏幕上打印出“Hello world”。注意在该行的末尾有一个“；”，分号多数时候表示一条语句的结束。在 C 语言中一行可以有多条语句，也可以一条语句跨越多行，但要在每条语句结束位置写一个“；”。

第 5 行：表示函数的结束，程序到这里执行完毕。

1.1.4 C 程序中的注释

```
/* ************************
作者：
创建日期：
描述：
……
************************ */
#include <stdio.h>
int main(void)
{
    ……
}
```

在程序中添加注释是一个好的编程习惯，可以增强程序的可读性，并且对于将来的二次开发、升级维护都有很大的帮助。原则上讲，注释应当有助于对程序的阅读理解，注释不需要很多也不能太少，注释中措辞必须准确、易懂、简洁。不同语言对注释量的要求不同，对于 C 语言程序通常要求注释量不少于 20%，特殊的程序（例如交换机内程序）会要求不少于 75%。

对于 C 语言程序，注释应该出现在三种位置，一是在文件头部，二是在函数头部，三是函数体内的和代码混在一起的注释，第三种比例最高。对于文件头部的注释要求至少列出：版权声明、版本号、文件创建日期、作者、内容/功能、与其他文件的关系、修改日志等。头文件（.h 后缀的文件）的注释中还应有函数功能简要说明。函数头部注释要求至少列出：函数功能、输入/输出参数、返回值、调用/被调用关系等。和代码混在一起的注释主要出现在函数内部，是对具体功能实现代码的说明。对于注释的具体写法，不同软件公司的编码规范会有细节差异，但主体是一致的。除了上面提到的要求之外，通常还会有以下要求。

提示：

如果是初学 C 语言，那么下面要求中出现的代码会看不懂，反而会引起困惑，建议把 C 语言基础学完之后再来看这部分内容。对于本书应该在第 5 章学完之后学习。

- 写代码和写注释应该同步进行，也就是一边写代码一边写注释，修改代码的同时也必须修改相应的注释，以保证注释与代码的一致性，废弃的代码和对应的注释要同时删除。

- 注释的内容及措辞应当清楚明了、语义精确，防止注释出现二义性。这里常见的一个错误是，代码修改后没有及时修改注释，或者言不达意。
- 在注释中应减少缩写的使用，尤其是不常见的缩写。对于任何缩写，包括约定俗成的缩写，在当前文件初次出现时应该给出完整拼写并予以说明。
- C 程序的执行本质上是顺序执行的，所以注释应该紧邻被描述的代码，注释可以在代码的上方或右侧（符合阅读顺序和执行顺序），不可以在代码的下方。对于放在代码上方的注释有时还会和上方其他代码用一个空行隔开。例如：

```
int id;                                 /* 学籍号 */
char name[1024];                        /* 学生姓名 */
struct student *stup = NULL;            /* 一名学生的信息 */

 /* 取得学籍号、姓名 */
printf("\033[5;1H 请输入学籍号：");
scanf("%d", &id);
getchar();
```

- 如果变量、常量、宏等标识符不是自注释的，在定义时必须有对应注释，说明其含义。自注释指标识符可以说明自身属性、功能，例如，变量“user_name”，不用注释也知道基本含义，但标识符“un”就很难看懂了。注意，在实际项目中，单个字符的变量和类似“un”这样的变量就算愿意写注释也是禁用的。例如：

```
/* 保存用户信息的链表 */
struct  userinfo  *userinfo_head  =  NULL;
```

- 对于某种数据结构的标识符(包括数组、结构、类、枚举等)，如果其命名不是自注释的，在定义时必须有对应注释。对数据结构的整体性注释应放在上方紧邻位置，不可放在下面。对结构中的每个域（成员）的注释放在该域（成员）的右方。例如：

```
/* 关于用户信息链表的结构 */
struct userinfo {
char name[20];                  /* 用户名 */
char password[20];              /* 密码 */
struct userinfo *next;          /* 下一个节点 */
};
```

- 对于全局变量应减少使用，必须使用时要有详细的注释。包括功能/作用、取值范围、调用者、注意事项等。例如：

```
/* 这是一个用于进程间同步的信号量
** 取值范围：  -1 = 失败；其他 = IPC 标识符
** 调用举例：sem_id = semget((key_t)3477, 1, 0666 | IPC_CREAT);
*/
static int sem_id;
```

- 注释和对应代码应该同步缩进，以便于阅读理解，避免歧义。
- 对于各种分支语句（条件分支、循环语句等）必须编写注释。这些注释对于测试、维护、开发相关人员理解程序功能帮助很大。
- 对于 switch-case 语句，如果一个 cese 结束之后不能调用 break，则必须详加注释以说明原因，防止意外遗漏 break 而导致逻辑错误。
- 不能在表达式内或一行代码内插入注释。这种注释实际上影响阅读理解，而且有的编译器会在编译时报错。
- 从业务（功能）的角度进行注释，使代码更容易理解和检查错误。例如：

```
/* 保存一名学生的信息 */
stup = (struct student *)malloc(sizeof(struct student));
if(NULL == stup)
{
    printf("\033[24;1H 分配内存失败！");
    return 1;
} /* end if */
stup->id = id;
strncpy(stup->name, name, 19);
```

- 在代码块结束位置（“}”所在位置）的右侧写注释，表明是某种类型的代码块结束或者某功能的结束。在多重嵌套中，这样注释可以提高代码的可读性。例如：

```
int main(void)
{
      printf("\033c");
      for(;;)
      {
            /* 提示主菜单 */
            key = print_menu();
            /* 功能选择 */
            switch(key)
            {
                  case 0: printf("\033c"); return 0;
                  case 1: input(); break;
                  case 2: list(); break;
                  case 3: delete(); break;
                  case 4: update(); break;
                  case 5: search(); break;
                  default:
                        printf("\033[24;1H 请输入正确选项！");
            }/* end switch */
            clear_input();
      } /* end for */
}/* end int main(); */
```