

高等教育 立体化精品 系列规划教材

# C语言程序设计

## 立体化教程（附微课视频）

◎ 李刚 唐炜 主编

◎ 章万静 邢海霞 副主编



网站支持: <http://www.ptpedu.com.cn>  
<http://222.184.16.210/c>

### 教学资源包

25 个重难点微课教学视频

13 章精美详尽 PPT 课件

15 个项目应用实例及源程序

### 教学辅助包

教材每章配套模拟试题库

教材每章配套习题答案及源程序

教材每章所有例题源程序

### 教学拓展包

50 个知识拓展训练动画资源

每年定期提供拓展案例资源包

人民邮电出版社  
POSTS & TELECOM PRESS



高等教育立体化精品系列规划教材

# C语言程序设计

## 立体化教程（附微课视频）

◎ 李刚 唐炜 主编

◎ 章万静 邢海霞 副主编

人民邮电出版社

北京

## 图书在版编目(CIP)数据

C语言程序设计立体化教程：附微课视频 / 李刚，唐炜 主编. — 北京：人民邮电出版社，2015.2  
高等教育立体化精品系列规划教材  
ISBN 978-7-115-37521-6

I. ①C… II. ①李… ②唐… 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(2014)第275578号

## 内 容 提 要

本书主要分为四篇：语法基础篇、程序设计结构篇、初级应用篇和高级应用篇。其中第一篇语法基础部分介绍了C语言概述和C语言数据与运算；第二篇程序设计结构部分介绍了算法与流程图、顺序结构程序设计、选择结构程序设计和循环结构程序设计；第三篇初级应用部分介绍了数组、函数和编译预处理；第四篇高级应用部分介绍了指针、构造类型、位运算和文件。

作者针对高等教育应用型本科院校和高职高专学校的教学方法和教学理念编写本书，在写作上由浅入深、循序渐进，主要采用实例化的编写方法，算法设计逐步讲解，重点知识配有微课视频并加以实践项目练习，进而提高学生算法设计和程序设计水平。

本书可作为高等教育应用型本科院校和高职高专学校计算机专业的教材，也可作为各类计算机培训课程的教材。

---

◆ 主 编 李 刚 唐 炜  
副 主 编 章万静 邢海霞  
责任编辑 王 威  
责任印制 杨林杰

◆ 人民邮电出版社出版发行 北京市丰台区成寿寺路11号  
邮编 100164 电子邮件 315@ptpress.com.cn  
网址 <http://www.ptpress.com.cn>  
三河市中晟雅豪印务有限公司印刷

◆ 开本：787×1092 1/16  
印张：15.25 2015年2月第1版  
字数：397千字 2015年2月河北第1次印刷

---

定价：35.00元

读者服务热线：(010)81055256 印装质量热线：(010)81055316  
反盗版热线：(010)81055315

# 前 言

本课程是一门电子信息类专业基础课程，也是其他研究计算机控制以及嵌入式技术等相关专业的一门基础和工具课程。本书是为 C 语言程序设计课程编写的教材，其内容选取符合职业能力培养、教学职场化、教材实践化的特点，适合高等教育应用型本科院校和高职高专学校使用。

本书主编具有 12 年的 C 语言程序设计课程一线教学经验，对应用型本科和高职高专学生学习有自己的教学方法和教学理念。本书采用实例化的编写方法，算法设计逐步讲解，重点知识加以实践项目练习，进而提高学生算法设计和程序设计水平。

本书的每一章都采用理论讲解、实例演示、解析说明、调试技巧、实践应用、知识提高的结构。

理论讲解	简要介绍章节涉及的知识点以及技能目标，深入浅出讲解重要知识点
实例演示	详细介绍了实例的具体算法设计过程和具体操作步骤，并用 C 语言实现程序设计
解析说明	归纳和总结了实例中的知识点，并对实例中需要注意的知识点进行强调补充
调试技巧	举例介绍了程序设计过程中常见的编译错误和警告，有利于学生进行程序的调试运行
实践应用	将章节知识点进行系统实践应用，有利于重点知识的全面掌握和应用能力的提高
知识提高	结合实践项目中的内容给出难度适中的拓展题目，通过练习，达到强化、巩固所学知识的目的

本书主要分为四篇：语法基础篇、程序设计结构篇、初级应用篇和高级应用篇。第 1~2 章为语法基础部分，第 3~6 章为程序设计结构部分，第 7~9 章为初级应用部分，第 10~13 章为高级应用部分。语法基础部分介绍了 C 语言概述和 C 语言数据与运算；程序设计结构部分介绍了算法与流程图、顺序结构程序设计、选择结构程序设计和循环结构程序设计；初级应用部分介绍了数组、函数和编译预处理；高级应用部分介绍了指针、构造类型、位运算和文件。

全书由李刚、唐炜任主编，章万静、邢海霞任副主编。本书编写分工如下：李刚编写第 1 章、第 4 章、第 6 章、第 11 章、第 13 章，唐炜编写第 3 章、第 5 章、第 7 章、第 8 章，章万静编写第 2 章、第 10 章，邢海霞编写第 9 章、第 12 章，全书由李刚完成统稿。本书在编写过程中还得到了刘万辉和朱旦晨的帮助和支持，在此表示衷心感谢。

本书中的实例源程序、电子测试题库，各章节讲解的微视频，多媒体教学系统等其他授课时需要的教学资源请登录人民邮电出版社教学服务与资源网（[www.ptpedu.com.cn](http://www.ptpedu.com.cn)）及 <http://222.184.16.210/c> 下载使用。

由于编者水平有限，书中难免存在错误或不妥之处，恳请广大读者批评指正。

编 者  
2014 年冬

# 目 录 CONTENTS

## 第一篇 语法基础 1

### 第1章 C语言概述 2

1.1 C语言的发展史及特点	2	1.2.2 C语言的基本结构	5
1.1.1 C语言的发展史	2	1.3 C语言程序的开发过程	5
1.1.2 C语言的特点	3	1.4 Visual C++ 6.0 开发环境	6
1.2 简单的C语言程序	3	1.5 常见编译错误与调试	10
1.2.1 第一个C语言程序	3	习题	12

### 第2章 C语言数据与运算 14

2.1 C语言数据类型	14	2.4.1 算术运算符与算术表达式	23
2.2 常量与变量	15	2.4.2 赋值运算符与赋值表达式	25
2.2.1 常量和符号常量	15	2.4.3 逗号运算符与逗号表达式	26
2.2.2 变量	18	2.5 常见编译错误与调试	27
2.3 数据类型转换	21	习题	29
2.4 运算符与表达式	22		

## 第二篇 程序设计结构 31

### 第3章 算法与流程图 32

3.1 算法定义及特征	32	3.2 流程图表示法	33
3.1.1 算法的定义	32	3.3 程序设计结构	34
3.1.2 算法的特征	33	习题	36

### 第4章 顺序结构程序设计 37

4.1 简单语句分析	37	4.4 常见编译错误与调试	42
4.2 格式化输入与输出	38	4.5 实践应用	44
4.2.1 scanf函数	38	4.5.1 应用项目1: 各类数据输出 格式控制	44
4.2.2 printf函数	40	4.5.2 应用项目2: 百位数分裂	45
4.3 字符输入与输出	41	习题	47
4.3.1 getchar函数	41		
4.3.2 putchar函数	42		

### 第5章 选择结构程序设计 49

5.1 条件判断表达式	49	5.4 多分支结构	57
5.1.1 关系表达式设计	49	5.4.1 if...else if语句	57
5.1.2 逻辑表达式设计	50	5.4.2 if...else 嵌套	59
5.2 单分支结构	52	5.4.3 switch语句	60
5.3 双分支结构	55	5.5 常见编译错误与调试	62

5.6 实践应用	64	5.6.2 应用项目 2: ATM 机操作模拟	66
5.6.1 应用项目 1: 健康状况检查	64	习题	68

## 第 6 章 循环结构程序设计 71

6.1 while 和 do while 循环结构	71	6.5 常见编译错误与调试	79
6.1.1 while 语句	71	6.6 实践应用	81
6.1.2 do while 语句	73	6.6.1 应用项目 1: 小白兔吃萝卜	
6.2 for 循环结构	74	智力问答	81
6.3 break 和 continue 语句	76	6.6.2 应用项目 2: ATM 机密码	
6.3.1 break 语句	76	输入控制	82
6.3.2 continue 语句	77	习题	84
6.4 循环嵌套	78		

## 第三篇 初级应用 87

### 第 7 章 数组 88

7.1 一维数组	88	7.3 二维数组	102
7.1.1 一维数组的定义	88	7.3.1 二维数组的定义	102
7.1.2 一维数组的初始化	89	7.3.2 二维数组的初始化	102
7.1.3 一维数组中元素的引用	90	7.3.3 二维数组中元素的引用	103
7.1.4 一维数组程序举例	90	7.3.4 二维数组程序举例	103
7.2 字符数组	93	7.4 常见编译错误与调试	106
7.2.1 字符数组的定义	93	7.5 实践应用	108
7.2.2 字符数组的初始化	93	7.5.1 应用项目 1: 冒泡排序法	
7.2.3 字符数组中元素的引用	93	简单实例	108
7.2.4 字符串和字符串结束的标志	94	7.5.2 应用项目 2: 统计字符串中	
7.2.5 字符串处理函数	94	各类别字符的数量	109
7.2.6 字符数组程序举例	100	习题	111

### 第 8 章 函数 114

8.1 函数概述	114	8.6.1 局部变量和全局变量	124
8.2 函数定义	115	8.6.2 动态变量和静态变量	127
8.3 函数的一般调用方式	118	8.6.3 变量的其他存储类别	128
8.3.1 形式参数和实际参数	118	8.7 内部函数与外部函数	130
8.3.2 函数返回值	118	8.7.1 内部函数	130
8.3.3 函数调用方式	119	8.7.2 外部函数	131
8.4 函数的特殊调用方式	120	8.8 常见编译错误与调试	132
8.4.1 函数的嵌套调用	120	8.9 实践应用	134
8.4.2 函数的递归调用	122	应用项目: 模拟 ATM 机存取款操作	134
8.5 数组作为函数参数	123	习题	138
8.6 变量存储类别及作用域	124		

## 第9章 编译预处理 141

9.1 宏定义	141	9.5 实践应用	147
9.2 文件包含	143	应用项目：教务管理系统登录模块设计	147
9.3 条件编译	144	习题	149
9.4 常见编译错误与调试	146		

## 第四篇 高级应用 151

### 第10章 指针 152

10.1 指针概念及引用	152	10.4.1 字符串的表示	163
10.1.1 指针的概念	152	10.4.2 字符指针变量与 字符数组的区别	164
10.1.2 指针变量	153	10.4.3 指向字符串的指针变量 程序举例	165
10.1.3 指针变量的指向	154	10.5 指针作为函数参数	167
10.2 变量的指针和指向变量的 指针变量	154	10.5.1 指向变量的指针 作为函数的参数	167
10.2.1 定义一个指针变量	154	10.5.2 指向数组的指针 作为函数的参数	168
10.2.2 确定指针变量的指向	154	10.5.3 指向字符串的指针 作为函数的参数	169
10.2.3 指针变量的引用	155	10.6 常见编译错误与调试	170
10.2.4 指向变量的指针变量程序举例	155	10.7 实践应用	173
10.3 数组指针和指向数组的 指针变量	156	10.7.1 应用项目1：求一名同学所有 课程的平均成绩	173
10.3.1 定义一个指针变量指向一维数组	157	10.7.2 应用项目2：大小写字母转换	174
10.3.2 一维数组元素的表示法	158	习题	176
10.3.3 指针变量的运算	158		
10.3.4 指向一维数组的指针变量 程序举例	161		
10.4 字符串指针和指向字符串的 指针变量	162		

### 第11章 构造类型 179

11.1 结构体	179	11.3.2 枚举变量的引用	189
11.1.1 结构体类型的定义	179	11.4 类型定义符 typedef	190
11.1.2 结构体变量的定义	181	11.5 链表	191
11.1.3 结构体变量的引用	183	11.5.1 动态存储分配	191
11.1.4 结构体数组	184	11.5.2 链表概述与建立	191
11.2 共用体	186	11.6 常见编译错误与调试	193
11.2.1 共用体类型的定义	186	11.7 实践应用	194
11.2.2 共用体变量的定义	187	11.7.1 应用项目：学生奖学金评定系统设计	194
11.2.3 共用体变量的引用	187	习题	198
11.3 枚举类型	188		
11.3.1 枚举类型的定义	188		

## 第 12 章 位运算 200

<p>12.1 位运算概述 200</p> <p>12.2 位运算符及其表达式 200</p> <p style="padding-left: 20px;">12.2.1 “按位与”运算符 (&amp;)</p> <p style="padding-left: 40px;">——And 201</p> <p style="padding-left: 20px;">12.2.2 “按位或”运算符 ( )</p> <p style="padding-left: 40px;">——Or 201</p> <p style="padding-left: 20px;">12.2.3 “按位异或”运算符 (^)</p> <p style="padding-left: 40px;">——Xor 201</p> <p style="padding-left: 20px;">12.2.4 “按位取反”运算符 (~)</p> <p style="padding-left: 40px;">——Not 202</p>	<p>12.2.5 “左移”运算符 (&lt;&lt;)</p> <p style="padding-left: 40px;">——Lsh 202</p> <p>12.2.6 “右移”运算符 (&gt;&gt;)</p> <p style="padding-left: 40px;">——Rsh 202</p> <p>12.3 位运算应用 202</p> <p>12.4 常见编译错误与调试 204</p> <p>12.5 实践应用 205</p> <p style="padding-left: 20px;">应用项目：数据右循环移位操作 205</p> <p>习题 207</p>
--	--

## 第 13 章 文件 208

<p>13.1 文件概述 208</p> <p>13.2 文件指针 209</p> <p>13.3 文件的打开与关闭 209</p> <p style="padding-left: 20px;">13.3.1 文件的打开 fopen() 209</p> <p style="padding-left: 20px;">13.3.2 文件的关闭函数 fclose() 211</p> <p>13.4 文件的读写操作 211</p> <p style="padding-left: 20px;">13.4.1 字符读写函数 fgetc()和 fputc() 211</p> <p style="padding-left: 20px;">13.4.2 字符串读写函数 fgets()和 fputs() 214</p>	<p>13.4.3 数据块读写函数 fread()和 fwrite() 215</p> <p>13.4.4 格式化读写函数 fscanf()和 fprintf() 217</p> <p>13.5 常见编译错误与调试 218</p> <p>13.6 实践应用 220</p> <p style="padding-left: 20px;">应用项目：学生成绩管理系统设计 220</p> <p>习题 226</p>
--	---

## 附录 228

附录 A 常用字符与 ASCII 码对照表 228	附录 C C 语言常用函数表 231
附录 B 运算符优先级和结合性 229	

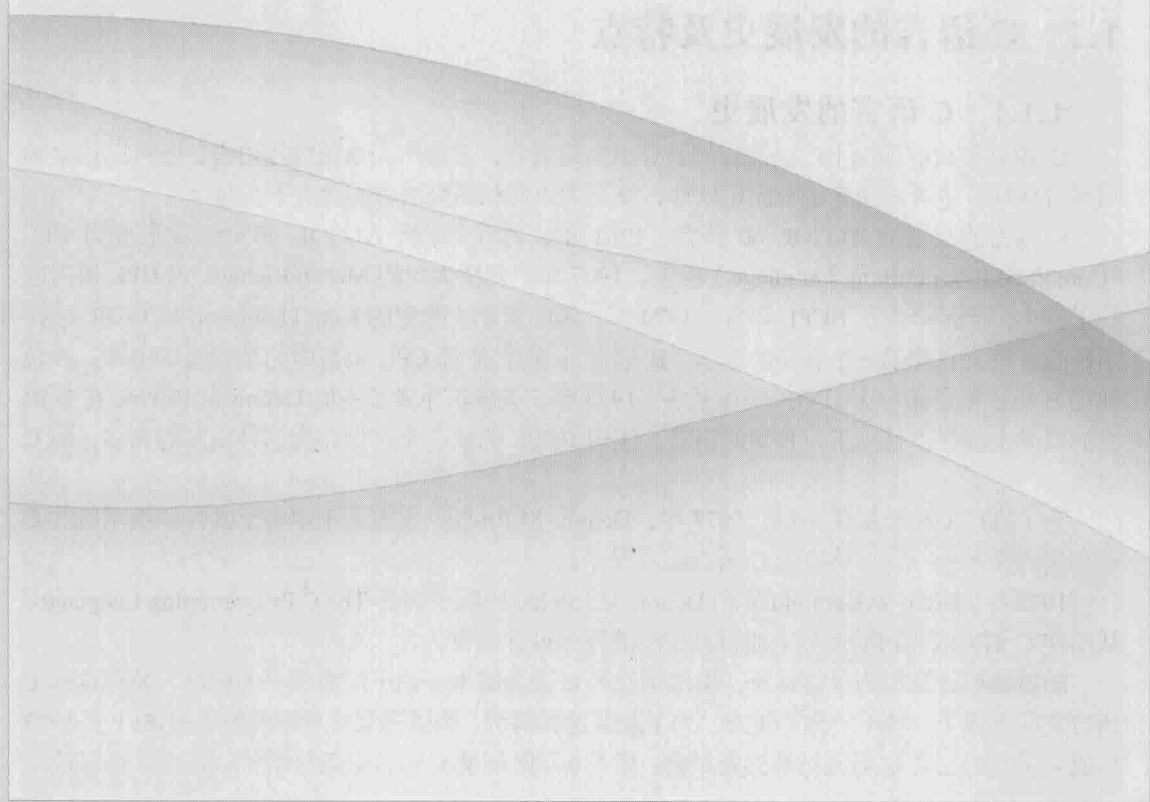
## 参考文献 235



第一章  
汉语入门

# 第一篇

## 语法基础



## 学习目标

1. 了解 C 语言的发展史及特点。
2. 掌握简单 C 语言程序的基本构成。
3. 掌握 Visual C++6.0 开发环境及应用。
4. 掌握编写 C 语言程序的基本步骤和调试过程。

C 语言是国际上广泛流行的高级程序设计语言之一，它具有语言简洁、使用方便灵活、移植性好、能直接对系统硬件和外围接口进行控制等特点。本章将简要介绍 C 语言的发展和特点、C 程序结构及 C 程序的运行环境。

## 1.1 C 语言的发展史及特点

### 1.1.1 C 语言的发展史

C 语言是目前国际上广泛流行的计算机高级语言，适合作为系统描述语言，既可以用来编写系统软件，也可以用来编写应用软件，集汇编语言和高级语言的优点于一身。

C 语言的原型是 ALGOL 60 语言。1963 年，剑桥大学将 ALGOL 60 语言发展成为 CPL (Combined Programming Language) 语言。1967 年，剑桥大学的 Martin Richard 对 CPL 语言进行了简化，于是产生了 BCPL 语言。1970 年，美国贝尔实验室的 Ken Thompson 将 BCPL 进行了修改，并为它起了一个有趣的名字“B 语言”，意思是将 CPL 语言中的精华提炼出来，并且他用 B 语言写了第一个 UNIX 操作系统。1973 年，美国贝尔实验室的 Dennis M.Ritchie 在 B 语言的基础上最终设计出了一种新的语言，他用 BCPL 的第二个字母作为这种语言的名字，即 C 语言。

为了推广 UNIX 操作系统，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》。

1978 年，Brian W.Kernighian 和 Dennis M.Ritchie 出版了名著 The C Programming Language，从而使 C 语言成为目前世界上流行最广泛的程序设计语言。

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，美国国家标准研究所 (ANSI) 于 1983 年成立了专门定义 C 语言标准的委员会，花了 6 年时间使 C 语言迈向标准化。随着 C 语言被广

泛关注与应用, ANSI C 标准于 1989 年被采用。该标准一般被称为 ANSI/ISO Standard C, 成为现行的 C 语言标准, 而且成为最受欢迎的语言之一。许多著名的系统软件都是由 C 语言编写的。

到了 1995 年, 在 ANSI C 的基础上增加了一些库函数, 出现了初步的 C++ 语言。C++ 进一步扩充和完善了 C 语言, 成为一种面向对象的程序设计语言。C++ 目前流行的最新版本是 Microsoft Visual C++ 6.0 (简称 VC++)。VC++ 提出了一些更为深入的概念, 它所支持的面向对象概念很容易将问题空间直接映射到程序空间, 为程序员提供了一种与传统结构程序设计不同的思维方式和编程方法, 但同时也增加了整个语言的复杂性, 掌握起来有一定难度。

C 语言是 C++ 语言的基础, C++ 语言和 C 语言在很多方面是兼容的。因此, 掌握了 C 语言, 再进一步学习 C++ 语言就能以一种熟悉的语法来学习面向对象的语言, 从而达到事半功倍的目的。

### 1.1.2 C 语言的特点

C 语言是一种极具生命力的语言, 它具有很多方面的特点。一般可归纳如下。

(1) C 语言具有结构语言的特点, 程序之间很容易实现段的共享。它具有结构化的流程控制语句实现选择结构、循环结构, 允许采用缩进的书写形式编程。因此, 用 C 语言编写的程序层次结构清晰。

(2) C 语言的主要结构成分是函数。函数作为 C 程序的模块单位, 便于实现程序的模块化, 而且便于模块间相互调用及数据传递。

(3) 运算符丰富。有 34 种运算符和 15 个等级的运算优先顺序, 使表达式类型多样化, 可以实现在其他语言中难以实现的运算。

(4) 数据类型丰富。数据类型有整型、实型、字符型、数组类型、指针类型、结构体类型、共用体类型及枚举类型, 能用来实现各种复杂的数据结构运算。

(5) 比较接近硬件。允许直接访问物理地址, 能进行位操作, 能实现汇编语言的大部分功能, 可以直接对硬件进行操作。

(6) 语法限制少和程序设计自由度大。允许程序编写者有较大的自由度, 放宽了以往高级语言严格的语法检查, 较好地处理了“限制”与“灵活”这一对矛盾。

(7) 生成目标代码质量高、程序执行效率高。只比汇编程序生成的目标代码效率低 10%~20%。

(8) 可移植性好。基本上不做修改就能用于各种型号的计算机和各种操作系统。

以上只介绍了 C 语言最容易理解的一般特点, 至于 C 语言内部的其他特点将在以后的章节陆续介绍。

## 1.2 简单的 C 语言程序

### 1.2.1 第一个 C 语言程序

首先通过 C 语言程序的简单实例来说明 C 语言源程序结构的特点和书写方式。

【例 1.1】输出某位同学的学号、姓名、性别、联系方式。

代码清单 1.1:

```
main()
{
    printf("学号:35013101\n");
```

```

printf("姓名:王 迪\n");
printf("性别:女\n");
printf("联系方式:13861595511\n");
}

```

程序运行后，结果如图 1.1 所示。

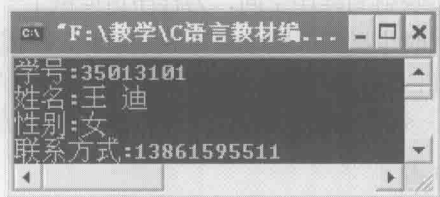


图 1.1 例 1.1 程序运行结果

程序说明如下。

- (1) 每一个 C 语言源程序都必须有且只能有一个主函数（main 函数）。
- (2) 一个函数由两部分组成：函数的首部和函数体。
- (3) 函数体由大括号“{}”括起来。
- (4) 函数调用语句，printf 函数的功能是将要输出的内容送到显示器去显示。
- (5) 双引号内的字符串按原样输出，但“\n”是转义字符，代表换行。

下面再来看一个相对复杂的 C 语言程序。

【例 1.2】已知两个整数，求它们的和，并输出。

代码清单 1.2:

```

#include "stdio.h"           //include 为文件包含命令
main()                       //主函数
{
    int x,y,sum;             //定义三个变量
    x=2;y=3;                //变量赋值
    sum=x+y;                //计算和
    printf("sum=%d\n",sum); //输出结果
}

```

程序运行后，结果为 sum=5。

程序说明如下。

(1) #include 称为文件包含命令，扩展名为.h 的文件称为头文件，利用<>或者双引号""括起来，表明将该文件包含到程序中来，成为程序的一部分。

(2) //表明注释部分，也可以写成/\*……\*/，区别是//只能用在一行，/\*……\*/可用于多行同时注释，注释只起说明作用，在编译时不进行编译，当然也不被执行。注释可以放在程序的任何位置，内容也可以是任意字符。

(3) 每一个说明、每一条语句都必须以分号结尾。但预处理命令、函数头和花括号“{}”之后不能加分号。

(4) 一行内可以书写一条或多条语句。例如“x=2;y=3;”。

## 1.2.2 C 语言的基本结构

概括地说,一个 C 语言源程序可由如下 5 个部分组成:

- 预处理部分(文件包含属于预处理内容)
- 变量说明部分
- 函数原型声明部分
- 主函数部分
- 自定义函数部分

注意!

(1)并非所有的 C 语言源程序都必须包含上述的 5 个部分,一个最简单的 C 语言程序可以只有包含文件的部分和主函数部分。

(2)每个 C 语言源程序都必须有且只能有一个主函数,主函数的组成形式如下所示。

```
main()  
{  
    变量说明部分  
    程序语句部分  
}
```

(3)每个 C 语言源程序可以有零个或多个自定义的非主函数,自定义非主函数的形式与主函数形式相同,只是它的名称不能是 main。

函数名([参数列表])

```
{  
    变量说明部分  
    程序语句部分  
}
```

(4)每个 C 语言源程序的语句必须用分号“;”结束。对于其中各部分的作用、使用方法和采用什么语句来完成,可以在后续章节中通过对基本表达式、结构控制语句的学习进一步掌握,并通过了解模块化设计等方面的内容,来掌握 C 语言程序设计的思想。

(5)当 C 语言源程序由多个函数组成时,主函数(main)可以定义在程序中的任何位置(不能在其他的函数体内定义),不论主函数处于源程序的任何位置,程序总是从主函数(main)开始执行,且总是在主函数中结束执行的。

## 1.3 C 语言程序的开发过程

用 C 语言编写的程序称为源程序,是不能直接运行的。一般 C 程序开发要经历 4 个基本步骤:编辑、编译、连接和运行,其操作过程如图 1.2 所示。

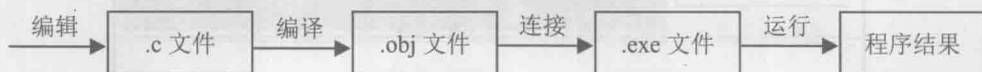


图 1.2 C 程序的编辑、编译、连接和运行流程图

### 1. 编辑源程序

一般来说,编辑是指 C 语言源程序的录入和修改。使用字处理软件或编辑工具(Visual C++)

将源程序以文本文件形式保存到磁盘，建议采用 Visual C++ 6.0，易于编辑和程序结构布局，源程序文件名由用户自己选定，但扩展名必须为“.c”。

## 2. 编译源程序

编译就像某语种的翻译一样，编译的功能就是调用“编译程序”，将已编辑好的源程序翻译成二进制的目标代码。如果源程序没有语法错误将产生一个与源程序同名，以“.obj”为扩展名的目标程序。如果发现错误，则不能生成目标程序，需要回到编辑状态修改源程序，直到没有错误为止。

## 3. 连接目标程序

编译后产生的目标程序往往形成多个模块，还要和库函数进行连接才能运行，连接过程是使用系统提供的“连接程序”运行的。连接后，产生以“.exe”为扩展名的可执行程序。

## 4. 运行


可执行程序生成后，就可以在操作系统的支持下运行。若执行结果达到预期的目的，则开发工作到此完成；否则，就要进一步检查修改源程序，重复上述步骤，直到取得最终的正确结果为止。

# 1.4 Visual C++ 6.0 开发环境

## 微课视频：VC++6.0 开发环境介绍

C 语言的开发环境有 Turbo C 环境与 Visual C++ 6.0 环境，本书以 Visual C++ 6.0 为操作平台，介绍 C 语言的应用。本小节简单介绍在 Visual C++ 6.0（以下简称 VC++）集成环境中，如何建立 C 语言程序，以及如何编辑、编译、连接和运行 C 语言程序。

### 1. 启动 Visual C++ 6.0

选择“开始”菜单“程序”项或者桌面上快捷方式中的 Microsoft Visual C++ 6.0 ()，启动 VC++ 6.0 编译系统。VC++ 6.0 主窗体如图 1.3 所示。

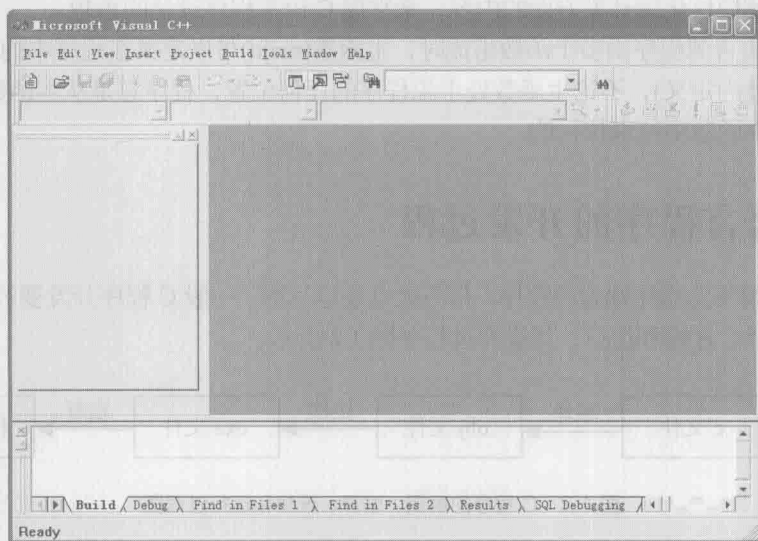


图 1.3 VC++ 6.0 主窗体

## 2. 新建项目

选择“File”菜单中的“New”项，如图 1.4 所示，在出现的窗口中选择“Files Projects”选项卡中的 Win32 Console Application，输入工程名，设定保存位置，如图 1.5 所示，单击“OK”按钮确定后弹出新窗口，选择默认选项，单击“Finish”按钮。

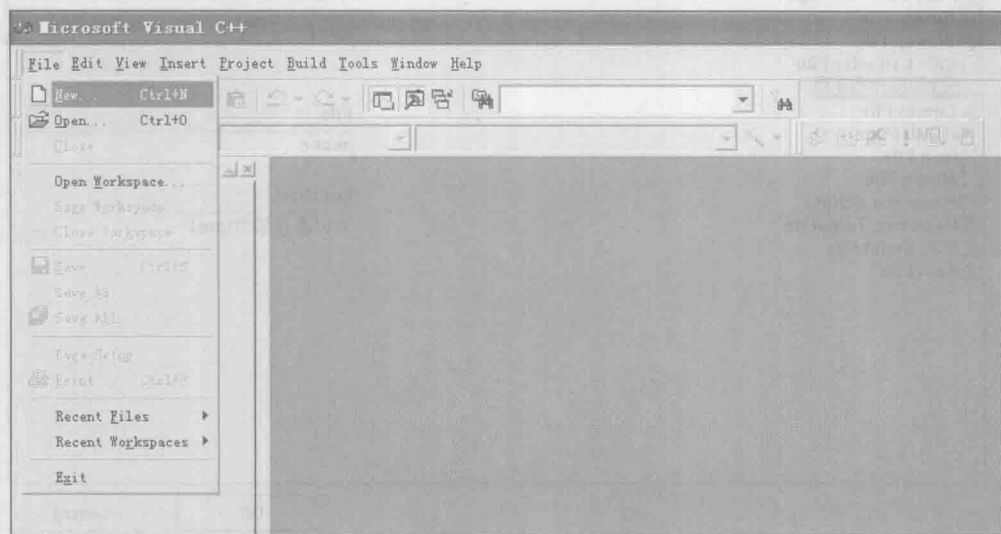


图 1.4 选择新建项菜单

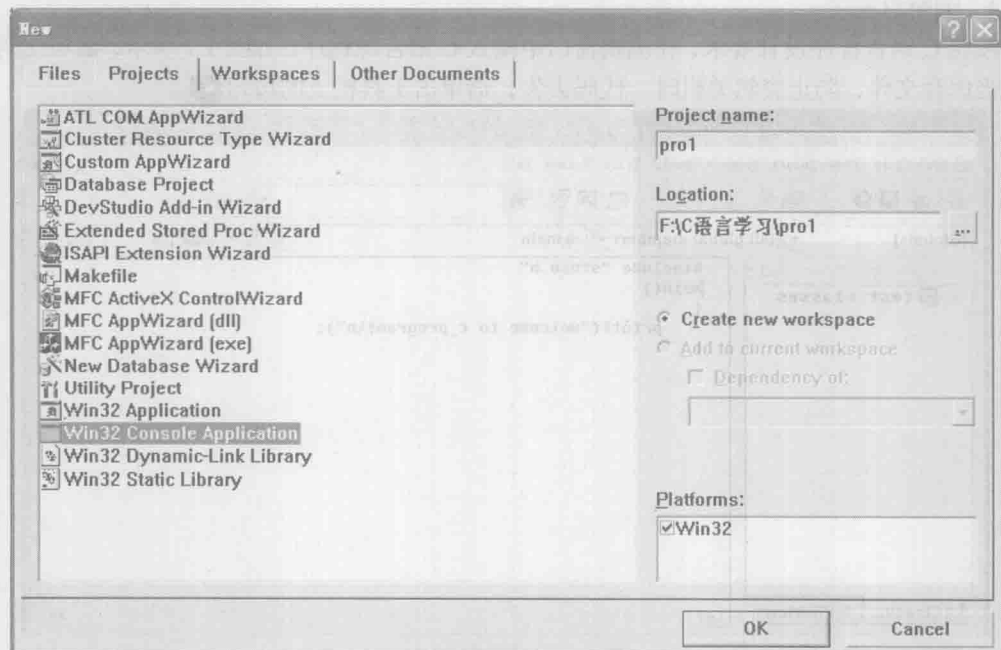


图 1.5 新建项目窗口

## 3. 新建文件

选择“File”菜单中的“New”项，如图 1.4 所示，在出现的窗口中选择“Files”选项卡中的 C++ Source File，如图 1.6 所示，在“File”下面的文本框中输入文件名 test.c。注意扩展名必

须是“.c”，否则是 C++程序。单击“OK”按钮，弹出如图 1.7 所示的窗口。

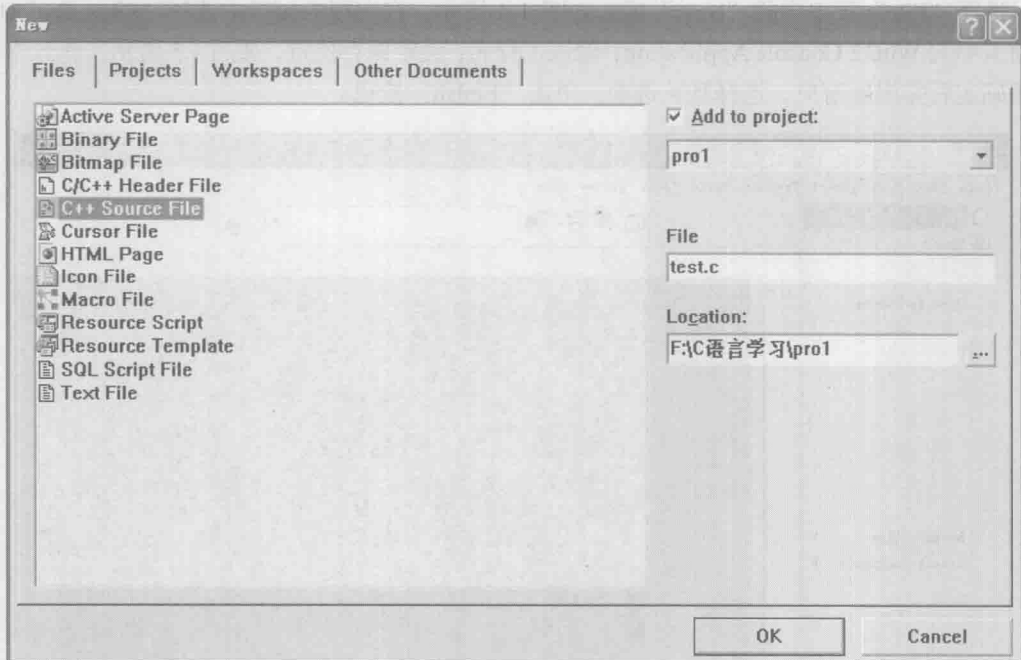



图 1.6 新建文件窗口

#### 4. 编辑源程序

按照 C 语言程序设计要求，在编辑窗口中输入 C 语言源程序，如图 1.7 所示，编写代码时，要适当保存文件，防止突然关机时，代码丢失，请单击工具栏上的按钮。

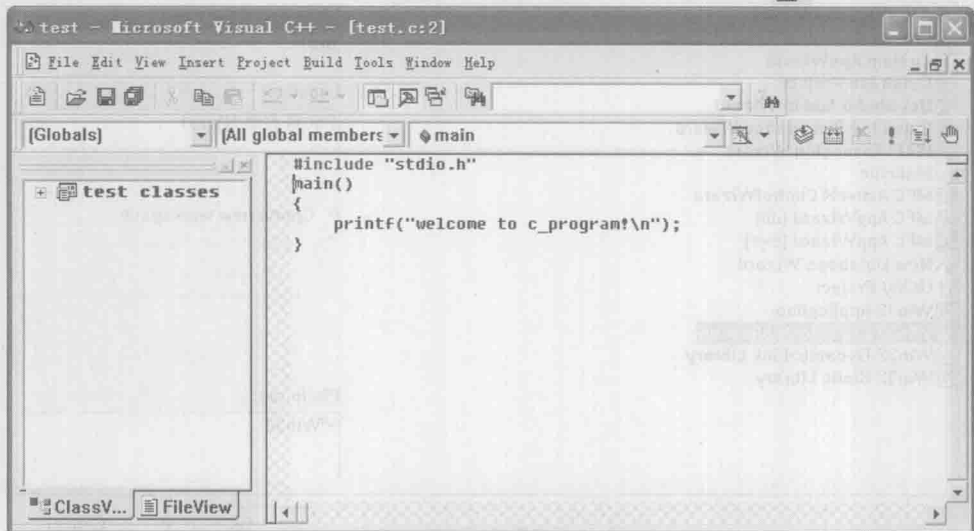



图 1.7 编辑源程序窗口

#### 5. 编译源程序

选择“Build”菜单中的“Compile test.c”项或单击工具栏上的按钮进行编译。如果程序未存盘，系统在编译前自动打开保存对话框，提示用户保存程序。在编译过程中如果出现错误，将在下方窗口中列出所有错误和警告。双击显示错误或警告的当前行，则光标定位在有错误的



代码行，修改错误后重新编译，反复修改至无错误为止。没有任何错误时，显示错误和警告数都为 0，如图 1.8 所示。

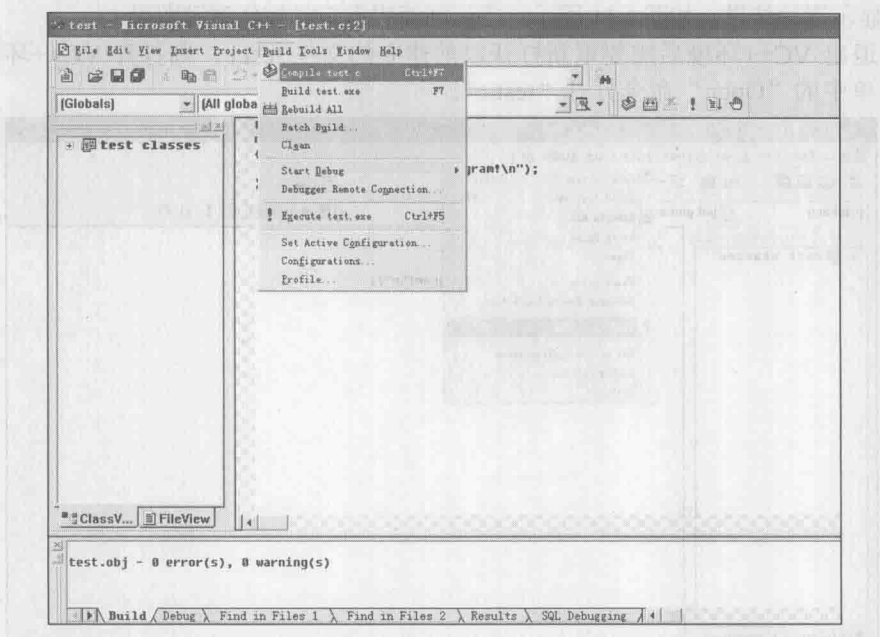



图 1.8 编译窗口

### 6. 连接目标程序

编译没有错误之后需要构建.exe 文件，选择“Build”菜单中的“Build test.exe”项或单击连接按钮，与编译时一样，如果系统在连接过程中发现错误，将在如图 1.9 所示的窗口中列出所有错误与警告。修改错误重新编译和连接，直到编译和连接都没有错误为止。

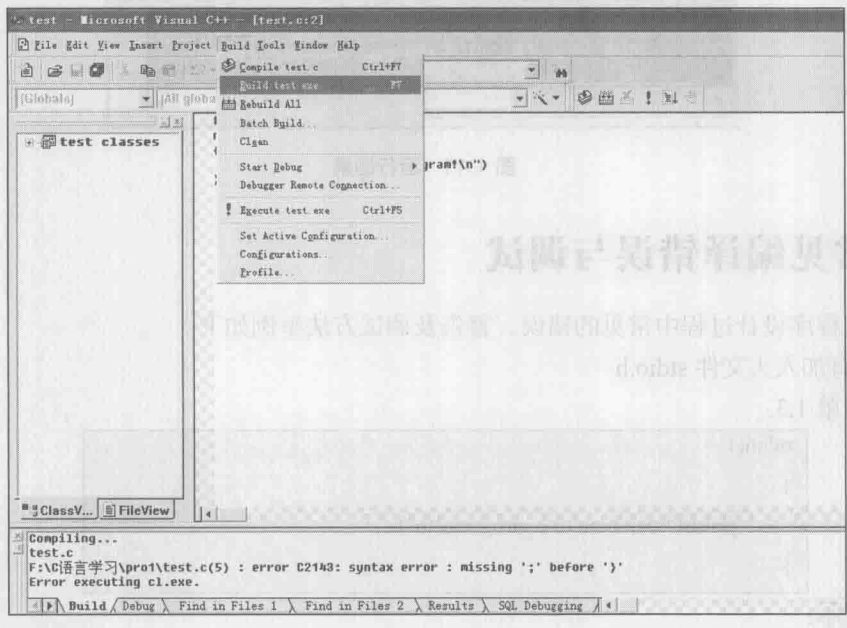


图 1.9 连接窗口