



教育部高职高专计算机教指委规划教材

任务引领型

C语言程序设计 项目教程

C YUYAN CHENGXU SHEJI
XIANGMU JIAOCHENG

主 编/吕新平

 中国人民大学出版社

教育部高职高专计算机教指委规划教材

C 语言程序设计项目教程

主 编 吕新平
副主编 孟祥瑞 邱发林 池 云
参 编 李超燕 沈宇平 邱 斌

中国人民大学出版社

· 北京 ·

图书在版编目 (CIP) 数据

C 语言程序设计项目教程/吕新平主编. —北京: 中国人民大学出版社, 2011
(教育部高职高专计算机教指委规划教材)
ISBN 978-7-300-12889-4

I. ①C… II. ①吕… III. ①C 语言-程序设计-高等学校: 技术学校-教材 IV. ①TP312

中国版本图书馆 CIP 数据核字 (2010) 第 205099 号

教育部高职高专计算机教指委规划教材

C 语言程序设计项目教程

主 编 吕新平

副主编 孟祥瑞 邱发林 池 云

参 编 李超燕 沈宇平 邱 斌

出版发行	中国人民大学出版社	
社 址	北京中关村大街 31 号	邮政编码 100080
电 话	010-62511242 (总编室)	010-62511398 (质管部)
	010-82501766 (邮购部)	010-62514148 (门市部)
	010-62515195 (发行公司)	010-62515275 (盗版举报)
网 址	http://www.crup.com.cn	
	http://www.ttrnet.com (人大教研网)	
经 销	新华书店	
印 刷	北京市鑫霸印务有限公司	
规 格	185 mm×260 mm 16 开本	版 次 2011 年 3 月第 1 版
印 张	18.75	印 次 2011 年 3 月第 1 次印刷
印 数	443 000	定 价 32.00 元

版权所有 侵权必究 印装差错 负责调换

中国人民大学出版社
教育部高职高专计算机教指委规划教材
编委会委员名单

主任委员 温 涛

委 员 (按姓名拼音排序)

崔连和	褚建立	高爱国	姜广坤	靳广斌
李 红	李明革	李万龙	李永平	连为民
刘昌明	邵东华	申亚宁	时武略	宋维堂
王 伟	王跃胜	吴建宁	向 隅	肖 玉
谢伟红	许文宪	严仲兴	杨淑清	易著梁
尹敬齐	张 辉	章忠宪	赵晓玲	甄立常
周 奇				

秘 书 董本清 孙 琳



总 序

近年来，我国高等教育取得了跨越式发展，毛入学率由1998年的8%迅速增长到2010年的25%，已经进入到大众的发展阶段，这其中，高等职业教育对实现“形成全民学习、终身学习的学习型社会”、“构建终身教育体系”的宏伟目标，发挥着其他教育形式不可替代的作用。

质量是职业教育的生命，社会需求是职业教育发展的终极动力。新颁布的《国家中长期教育改革和发展规划纲要》特别强调通过推进教育教学改革来提高质量。《纲要》要求通过课程、教材、教学模式和评价方式的创新，推进就业创业教育，实现人才培养方式转变，着力提高学生的职业道德、职业技能和就业创业能力。

实际上，为了适应我国高等职业教育的发展，全面提高教育教学质量，教育部主管部门先后启动了“国家精品课程建设”和“国家示范性高等职业院校建设计划”，经过四年的建设，无论是办学条件、人才培养模式，还是学生的就业质量都取得了显著进步；同时，也涌现出了一批高水平的优秀课程和优秀教材，为传播优秀教学理念、教学方法和教学内容起到了重要作用，为提高教学质量奠定了坚实基础。

为进一步深化教育教学改革和精品课程建设，进一步挖掘优秀的课程和教材，推广优秀的教育成果，扩大精品课程的受益面，在教育部高等学校高职高专计算机类专业教学指导委员会的指导下，中国人民大学出版社组织召开了计算机类专业的教材研讨会，并成立了教材编审委员会，计划在未来两三年内陆续推出百种高职高专计算机系列精品教材。

此套教材的作者大都是有着丰富的职业教育教学经验和较高专业学术水平的专家和教授。教材内容的选择克服了追求理论“大而全”的不足，做到了少而精，有针对性，突出了能力的训练和培养；教材体例的安排突出了学习使用的弹性和灵活性，形成文字教材和多媒体教程相结合的立体化教材，加强了教师对学生学习过程的指导和帮助，形象生动、灵活方便，更能适应学员在职、业余自学，或配合教师讲授时使用，相信会起到很好的教学效果。为满足教师在实际教学中的需求，本套教材在编写体例形式上不拘一格，具备“任务引领型”、“案例型”、“项目实训型”等写作特点，其目的是让学生在学中练、练中学，在实际动手练习中掌握理论知识的专业技能。

我们期待，这套高职高专计算机精品教材能够为促进我国高校 IT 职业教育的教学质量做出积极的贡献；我们也相信，这套教材必将在实践中日臻完善、追求卓越！

教育部高等学校高职高专计算机类专业教学指导委员会 主任委员

大连东软信息学院院长 温涛教授

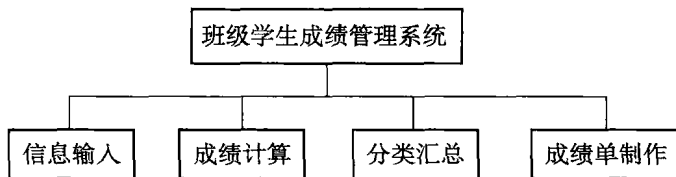
二〇一〇年六月

前言

C语言是一种被广泛学习、普遍使用的计算机程序设计语言。它的高级语言形式、低级语言功能具有特殊的魅力。由于C语言具有完整的编程语言特点，因而被作为典型的教学语言。在计算机等级考试、全国计算机应用证书考试等多种计算机知识考试中都有C语言。另一方面，C语言又作为一门实用的、功能强大的程序设计语言，被程序设计人员广泛使用。因此，C语言是一门十分重要的程序设计语言。

在多年的教学实践中我们体会到，要真正掌握C语言，学习者的难度较大。其原因一方面是C语言自身的难度，另一方面是现有的教材以学习C语言要掌握的知识为重点，不能很好地将教学过程中出现的知识、技能与实际软件开发所需要的知识、技能结合起来，学习者的积极性和主动性不能得到充分的发挥。我们在开展该课程的教学活动中，以“职业活动导向、任务驱动、项目载体”，结合C语言的特点，通过“班级学生成绩管理系统”项目的开发，使学习者掌握C语言程序设计的知识：数据类型、分支控制、循环控制、函数的定义及调用、结构体及数组、指针、文件操作、编译预处理等，使学习者学会用C语言程序解决实际问题的能力。

“班级学生成绩管理系统”共设计了四个功能，如下图所示。





“班级学生成绩管理系统”包含如下项目：

- 项目 1 项目菜单设计
- 项目 2 学生成绩的输入与计算
- 项目 3 项目菜单的选择执行
- 项目 4 项目的整体框架设计
- 项目 5 项目中数组的应用
- 项目 6 项目中指针的应用
- 项目 7 项目中自定义数据类型
- 项目 8 项目中学生数据的存储与重用

我们将这 8 个项目作为本书的 8 个章标题，“班级学生成绩管理系统”采用“循序渐进”的原则，分为 21 个任务实施。各项目以 1~4 个任务为驱动，围绕完成任务设计必备的知识与理论进行讲解，使学习与应用融为一体。通过学习的深入逐步完善程序的功能，最后形成一个较为完整的程序。随着学习的逐步深入，学习者也可以自己增加新的模块，使程序逐步完善、实用。项目开发中的任务如下：

- 任务 1：用输入/输出函数初步设计项目菜单
- 任务 2：学生成绩的输入/输出
- 任务 3：总分与平均分的计算
- 任务 4：用 if 语句实现菜单的选择执行
- 任务 5：用 switch 语句实现菜单的选择执行
- 任务 6：用循环语句实现菜单的选择执行
- 任务 7：整体项目菜单函数
- 任务 8：子项目菜单函数
- 任务 9：系统实现的主函数
- 任务 10：使用数组查找学生最高、最低成绩
- 任务 11：使用数组查找成绩不合格的学生
- 任务 12：使用数组对学生的成绩进行排序
- 任务 13：使用指针查找学生最高、最低成绩
- 任务 14：使用指针查找成绩不合格的学生
- 任务 15：使用指针对学生的成绩进行排序
- 任务 16：学生记录的增加
- 任务 17：学生记录的删除
- 任务 18：学生记录的修改
- 任务 19：学生记录的显示

- 任务 20：学生信息的保存
- 任务 21：学生信息文件的打开

程序设计课程的教学是引导学习者利用计算机进行解题能力的培养过程。本书的项目和例题选择了比较典型的问题，强调对问题的分析过程，其目的在于通过对典型问题的分析，使学习者能举一反三，不断积累解决复杂问题的能力。

本书的实例、项目任务、习题都用 VC++ 6.0 调试通过。

除通过本书的项目开发可以培养学习者项目开发的技能外，本书各章后包含的大量习题，可使学习者很好掌握所学知识 with 技能。习题的形式采用与全国计算机等级考试类似的题型，因此该书也可作为全国计算机等级考试（二级 C 语言）的参考书目。

本书由吕新平老师主编，副主编孟祥瑞、邱发林、池云等老师对本书提出了许多建设性的建议，并设计了本书的项目，李超燕、沈宇平、邱斌等老师设计并调试了本书的项目，并编写了本书的课后习题。由于作者的水平和能力有限，书中难免有不当之处，恳请读者批评指正。

作者

2010 年 9 月

目 录

项目 1 项目菜单设计	1	2.3.9 变量的初始化	27
1.1 任务 1: 用输入/输出函数初步 设计项目菜单	1	2.3.10 数据类型的自动转换	27
1.2 必备知识与理论	2	2.3.11 强制类型转换	28
1.2.1 计算机程序设计语言	2	2.4 扩展知识与理论	29
1.2.2 高级程序设计语言的开发 过程	3	2.4.1 printf()函数	29
1.2.3 C 语言标准	4	2.4.2 scanf()函数	33
1.2.4 C 语言的特点	4	2.4.3 字符输入/输出函数 getchar() 与 putchar()	37
1.2.5 使用 Microsoft Visual C++ 6.0 开发 C 语言程序	4	习题 2	38
1.3 扩展知识与理论	5	项目 3 项目菜单的选择执行	40
1.3.1 函数	6	3.1 任务 4: 用 if 语句实现菜单的 选择执行	41
1.3.2 函数的组成部分	10	3.2 必备知识与理论	41
1.3.3 语句	11	3.2.1 算法的概念	41
1.3.4 名字与声明	11	3.2.2 算法的表示方法	43
1.3.5 变量及其赋值	13	3.2.3 结构化程序设计	47
1.3.6 算术运算	14	3.2.4 命题与 C 语言中的逻辑值	47
习题 1	16	3.2.5 关系运算符与关系表达式	48
项目 2 学生成绩的输入与计算	18	3.2.6 逻辑运算符与逻辑表达式	49
2.1 任务 2: 学生成绩的输入/输出	18	3.3 扩展知识与理论	49
2.2 任务 3: 总分与平均分的计算	19	3.3.1 if(表达式)语句	49
2.3 必备知识与理论	19	3.3.2 if...else 结构	51
2.3.1 定点数与浮点数的表示	20	3.3.3 if...else if 结构	53
2.3.2 整数的有符号类型与无符号 类型	20	3.4 任务 5: 用 switch 语句实现菜单的 选择执行	54
2.3.3 类型宽度与取值范围	21	3.5 必备知识与理论	55
2.3.4 整型常量	22	3.6 扩展知识与理论	59
2.3.5 实型常量	23	3.7 任务 6: 用循环语句实现菜单的 选择执行	60
2.3.6 字符类型及其常量	24	3.8 必备知识与理论	62
2.3.7 符号常量	26	3.8.1 while 语句	62
2.3.8 变量的定义	26	3.8.2 do...while 结构	65



3.8.3 for 结构	67	5.4.6 字符串的输入与输出	133
3.9 扩展知识与理论	71	5.4.7 字符串处理函数	135
3.9.1 break 语句	71	5.5 扩展知识与理论	138
3.9.2 continue 语句	72	5.5.1 二维数组的定义与引用	138
习题 3	73	5.5.2 二维数组的初始化	139
项目 4 项目的整体框架设计	85	5.5.3 向函数传递二维数组	141
4.1 任务 7: 整体项目菜单函数	85	习题 5	144
4.2 任务 8: 子项目菜单函数	86	项目 6 项目中指针的应用	154
4.3 任务 9: 系统实现的主函数	89	6.1 任务 13: 使用指针查找学生 最高、最低成绩	154
4.4 必备知识与理论	90	6.2 任务 14: 使用指针查找成绩 不合格的学生	155
4.4.1 设计 C 语言程序就是设计 函数	90	6.3 任务 15: 使用指针对学生的 成绩进行排序	155
4.4.2 函数结构	91	6.4 必备知识与理论	156
4.4.3 函数定义与函数声明	92	6.4.1 地址与指针	156
4.4.4 函数调用	95	6.4.2 指针变量的定义、初始化与 赋值	158
4.4.5 函数的递归调用	96	6.4.3 指针变量的引用	158
4.5 扩展知识与理论	99	6.4.4 指针的运算	161
4.5.1 变量的作用域和生存期	99	6.4.5 指向指针变量的指针与多级 指针	163
4.5.2 C 语言中变量的存储 类型	102	6.4.6 指向 void 类型的指针	163
4.5.3 “文件包含”处理	104	6.4.7 数组元素的指针引用	164
4.5.4 宏定义	106	6.4.8 多字符串的存储与处理	167
习题 4	109	6.4.9 内存的动态分配与动态数组的 建立	172
项目 5 项目中数组的应用	119	6.5 扩展知识与理论	174
5.1 任务 10: 使用数组查找学生 最高、最低成绩	119	6.5.1 指针参数与函数的地址传送 调用	174
5.2 任务 11: 使用数组查找成绩 不合格的学生	120	6.5.2 带参数的主函数	181
5.3 任务 12: 使用数组对学生的 成绩进行排序	120	6.5.3 返回指针值的函数	183
5.4 必备知识与理论	121	6.5.4 指向函数的指针	185
5.4.1 一维数组定义及数组元素 引用	121	习题 6	189
5.4.2 一维数组的初始化	123	项目 7 项目中自定义数据类型	199
5.4.3 一维数组元素的查找与 排序	124	7.1 任务 16: 学生记录的增加	199
5.4.4 数组与函数	128	7.2 任务 17: 学生记录的删除	201
5.4.5 字符数组与字符串	132	7.3 任务 18: 学生记录的修改	202

7.4 任务 19: 学生记录的显示	204
7.5 必备知识与理论	205
7.5.1 结构体类型及其定义	205
7.5.2 定义结构体类型变量及对 变量的初始化	207
7.5.3 结构体变量的操作	209
7.5.4 嵌套结构体类型	210
7.5.5 位段	211
7.5.6 结构体数组的定义与 初始化	213
7.5.7 对结构体数组元素的 操作	214
7.5.8 指向结构体变量的指针	215
7.5.9 指向结构体数组的指针	217
7.6 扩展知识与理论	218
7.6.1 链表的概念	218
7.6.2 链表结点的定义与链接 关系的建立	219
7.6.3 动态链表的建立	220
7.6.4 结构体变量作为函数 参数	226
7.6.5 用指向结构体变量的指针 作函数参数	226
7.6.6 返回结构体类型值的 函数	226
7.6.7 共用体的特点	227
7.6.8 共用体变量的应用	229
7.6.9 枚举类型数据	230
7.6.10 用 Typedef 定义类型	232
习题 7	234

项目 8 项目中学生数据的存储 与重用

8.1 任务 20: 学生信息的保存	240
--------------------------	-----

8.2 任务 21: 学生信息文件的 打开	241
8.3 必备知识与理论	242
8.3.1 文件及其分类	242
8.3.2 文件名	243
8.3.3 文件的位置指针与读写 方式	243
8.3.4 FILE 类型指针	243
8.3.5 流	244
8.3.6 文件的打开与关闭——流的 创建与撤销	245
8.3.7 文件的字符读/写	246
8.3.8 文件的字符串读/写	249
8.3.9 文件的格式化读/写	251
8.3.10 文件的记录读/写	252
8.4 扩展知识与理论	254
8.4.1 文件位置指针的定位 函数	254
8.4.2 文件随机读写程序的应用	255
8.4.3 ferror 函数	256
8.4.4 fcerrorr 函数	256
习题 8	256

“班级学生成绩管理系统”项目

总结	263
----------	-----

附录 1 常用字符与 ASCII 码对照表

(附表 1)	278
--------------	-----

附录 2 C 语言运算符的优先级与结合性

(附表 2)	279
--------------	-----

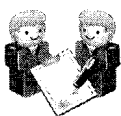
附录 3 C 语言中的关键字

.....	280
-------	-----

参考文献

.....	282
-------	-----

项目 1 项目菜单设计



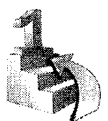
技能目标

- 能进行数据的运算并能编写输出数据的程序；
- 能使用 Microsoft Visual C++ 6.0 进行 C 程序的开发。



知识目标

- 了解 C 语言的特点和 C 语言的开发步骤；
- 初步了解 C 程序的组成结构和主函数的作用；
- 掌握五种基本算术运算符的使用：+、-、*、/、%，了解优先级与结合性；
- 理解 C 语言中各种运算的运算规则，由运算符和相关数据组成表达式的方法；
- 能够使用 printf() 函数进行信息的输出；
- 能编写简单程序。



项目任务与解析

建立班级学生成绩管理系统的菜单。

本项目包含 1 个任务：

- 用输入/输出函数初步设计项目菜单。

1.1 任务 1：用输入/输出函数初步设计项目菜单

1. 问题描述

现在要开发一套用于班级学生成绩管理的程序，该系统的功能包括信息输入、成绩计算、分类汇总以及成绩单制作。

根据 C 语言提供的输出函数，编写出一个显示功能菜单的主函数。

2. 具体实现

```
#include<stdio.h>
```

```
int main(void)
{
    /* 显示菜单 */
    printf("==== 班级学生成绩管理系统 ==== \n");
    printf("----- \n");
    printf("    1. 信息输入                \n");
    printf("    2. 成绩计算                \n");
    printf("    3. 分类汇总                \n");
    printf("    4. 成绩单制作            \n");
    printf("----- \n");
    return 0;
}
```

3. 知识分析

选择 C 语言作为程序的开发语言，那么首先就要了解这种语言的特点、特征以及运行环境等。

1.2 必备知识与理论

1.2.1 计算机程序设计语言

现代计算机可以自动完成计算过程，但必须要编制相应的程序。程序实际上是一个非常普通的概念，就是按照一定顺序安排的工作步骤。可以说，做任何事情都有相应的程序。做的事情不同，要求的效果不同，程序也就不同。例如，用同样的原料，采用不同的程序，会做出不同的菜肴来。

计算机程序需要用某种形式(语言)来描述。例如，用算盘进行计算，程序语言是用口诀来描述的。现代计算机的程序则是用计算机程序设计语言来描述的。程序(program)是由一系列指令组成的，是为解决某一具体问题而设计的一系列排列有序的指令的集合。设计及书写程序的过程称为程序设计。从计算机诞生到现在，程序设计语言也在伴随着计算机技术的进步不断升级换代。

使用计算机解决问题就需要编写程序，编写计算机程序就必须掌握计算机的程序设计语言。程序设计语言分为三类：机器语言、汇编语言和高级语言。

1. 机器语言

一台计算机中所有指令的集合称为该计算机的指令系统，这些指令就是机器语言，它是一种二进制语言。

由于计算机的机器指令和计算机的硬件密切相关，所以用机器语言编写的程序具有充分发挥硬件功能的特点，程序简洁、运行速度快。其缺点是编写的程序不直观、难懂、难记、难写，并且难以修改和维护。另外，机器语言是每一种计算机所固有的，不同类型计算机的指令系统和指令格式是不同的，因此机器语言没有通用性，是“面向机器”的语言。

2. 汇编语言

鉴于机器语言的缺点，人们用符号(称为助记符)来代替机器语言中的二进制代码，设计了“汇编语言”。汇编语言与机器语言基本上是一一对应的，但采用助记符来代替操作码，用符号来表示操作数地址(地址码)。这些助记符通常使用命令功能英文单词的缩写，便于记

忆，如用 ADD 表示加法、MOVE 表示传送等。

用汇编语言编写的程序具有质量高、执行速度快、占用内存少的特点，因此常用来编写系统软件、实时控制程序等。汇编语言同样是“面向机器”的语言，机器语言所具有的缺点，汇编语言也都有，只不过程度上不同而已。

3. 高级语言

高级语言与汇编语言相比，具有下面的优点：接近于自然语言，一般用英语单词表达语句，便于理解、记忆和掌握；语句与机器指令不存在一一对应的关系，一条语句通常对应多个机器指令；通用性强，基本上与具体的计算机无关，编程者无须了解具体的机器指令，可以把精力集中于解题思路和方法上去。

由于高级语言程序主要是描述计算机的解题过程，即描述复杂的加工处理过程，所以也称这种高级语言为“面向过程”的语言。

用高级语言编写的程序称为“源程序”。计算机不能直接运行源程序，通常有解释方式和编译方式两种方法在计算机上执行源程序。

解释方式，即让计算机运行解释程序，解释程序逐句取出源程序中的语句，对它作解释执行，输入数据，产生结果。解释方式的主要优点是计算机与人的交互性好，调试程序时，能一边执行一边直接改错，能较快得到一个正确的程序。其缺点是逐句解释执行，运行速度慢。

编译方式，即先运行编译程序，从源程序一次翻译产生计算机可直接执行的二进制程序（称为目标程序）；然后让计算机执行目标程序，输入数据，产生结果。编译方式的主要优点是计算机运行目标程序快，缺点是修改源程序后必须重新编译以产生新的目标程序。现在学习的 C 语言就采用编译方式运行。

也有将上述两种方式结合起来的，即先编译源程序，产生计算机不能直接执行的中间代码，然后让解释程序解释执行中间代码，如 Java 程序。这样做的好处首先是比直接解释执行快；更大的好处是中间代码独立于计算机，只要有相应的解释程序，就可在任何计算机上运行。

1.2.2 高级程序设计语言的开发过程

1. 分析问题、建立模型

一个具体的问题要涉及许多方面，这是问题的复杂性所在。为了便于求解，往往要忽略一些次要方面。这种通过忽略次要方面而找出解题规律的方法，就称为建立模型。

2. 表现模型

表现模型就是用一种符号语言系统来描述模型。模型的表现会随着对问题抽象程度的加深和细化，不断由领域特色向计算机可解释、可执行靠近，中间也可能采用一些其他的符号系统，如流程图等，直到最后用一种计算机程序设计语言描述出来。

3. 源程序的编写

源程序的编写就是在集成开发环境下，用具体的程序设计语言书写并修改程序的过程。为此就要掌握一种计算机程序设计语言。

4. 程序的编译与连接

写出一个高级语言程序后，并不是就可以立即执行。要让机器执行，还要将它翻译成由机器可以识别的机器语言程序。为区别它们，把用高级语言编写的程序（文件）称为源程序（文件），把机器可以直接辨认并执行的程序（文件）称为可执行程序（文件）。这一过程一般分

为以下两步：

步骤 1：在程序编辑过程中输入到源文件中的是一些字符码，但是机器可以直接处理的是 0、1 信息。为此，首先要将源程序文件翻译成用 0、1 码表示的信息，并用相应的文件保存。这种保存 0、1 码信息的文件称为目标程序文件。由源文件翻译成目标文件的过程称为编译。在编译过程中，还要对源程序中的语法和逻辑结构进行检查。编译任务由编译器 (compiler) 完成。编译后的目标程序文件还不能被执行，它们只是一些目标程序模块。

步骤 2：将目标程序模块以及程序所需的系统中固有的目标程序模块 (如执行输入/输出操作的模块) 连接成一个完整的程序。经正确连接所生成的文件才是可执行文件。完成连接过程的软件称为连接器 (linker)。

程序在编译、连接过程中，也可能发现错误。这时要重新对源程序进行编辑。

5. 程序的测试与调试

经编译、连接的程序文件，生成可执行文件，就可以让计算机执行了。但并不是一定会得到预期的结果，因为程序仍然可能存在某些错误。因此在程序交付用户使用前，需要测试一下。

测试是找出程序中可能存在的错误并加以改正。因此，应该测试程序在不同情况下运行的结果。输入不同的数据可以检测出程序在不同情况下运行的结果。测试的数据应是以“程序是会有错误的”为前提精心设计出来的，而不是随心所欲地乱凑而成的。它不仅应包含被测程序的输入数据，而且还应包括程序执行它们后的预期结果。每次测试都要把实际结果与预期结果相比较，以检验程序是否出错。

1.2.3 C 语言标准

C 语言在 1978 年由美国电话电报公司 (AT&T) 贝尔实验室正式发表。同时由 B. W. Kernighan 和 D. M. Ritchie 合著的《The C Programming Language》一书对 C 语言作了详细的描述。在此之后，由美国国家标准学会 (ANSI) 制定了一个 C 语言标准，于 1989 年发表，通常称之为 ANSI C (简称“C89”)。国际化标准组织在 1990 年制定的 C 标准，通常称为“C90”。从 1995 年开始，国际化标准组织着手对 C 标准作全面的修订，并于 1999 年形成正式的 C 语言标准，简称“C99”。

本书主要介绍 ANSI C，同时也会介绍 C99 中新增加的一些功能。C++、Visual C++、Java、C# 这些程序设计语言都是在 C 语言的基础上产生的。

1.2.4 C 语言的特点

C 语言的优点主要可以概括为以下几个方面：

- (1) 既有高级语言的程序思想与设计方法，又有低级语言的操作能力，所以它也被称为“中级语言”。
- (2) 结构化的体系。层次清晰，便于按模块化方式组织程序，易于调试和维护。
- (3) 在处理能力上具有丰富的运算符和数据类型，便于实现各类复杂的数据结构。可以直接访问内存的物理地址。
- (4) 在可移植性上可以方便地移植到不同的软、硬件环境中。
- (5) 代码效率高。

1.2.5 使用 Microsoft Visual C++ 6.0 开发 C 语言程序

目前可用于 C 语言的集成开发环境有很多，如 Turbo C、Borland C++、Microsoft Visual C++ 等。Microsoft Visual C++ 6.0 是一个功能强大的可视化软件开发工具，在全国

计算机等级考试二级的 C 语言考试中就采用这种开发环境，因此本书采用 Microsoft Visual C++ 6.0 作为 C 语言的集成开发环境。下面说明在 Microsoft Visual C++ 6.0 环境中开发 C 语言程序的过程。

(1) 启动 Microsoft Visual C++ 6.0。选择“开始 | 程序 | Microsoft Visual C++ 6.0 | Microsoft Visual C++ 6.0”菜单命令，启动 Microsoft Visual C++ 6.0。

(2) 新建项目。在 Microsoft Visual C++ 6.0 IDE 环境中，选择“File | New”菜单命令，打开“New”对话框。

在“New”对话框中，选择“Project”选项卡(如图 1—1 所示)，在其中的列表框中选择“Win32 Console Application”选项；在“Location”文本框中，输入或选择项目要保存的路径；在“Project name”文本框中输入项目的名称；单击“OK”按钮。

在出现的向导对话框中，单击“Finish”按钮完成新建项目。

(3) 新建文件。在“New”对话框中，选择“File”选项卡(如图 1—2 所示)，在其中的列表框中选择“C++ Source File”选项；在“Location”文本框中，输入或选择文件要保存的路径(一般与项目文件保存在相同的路径中)；在“File”文本框中输入文件的名称；单击“OK”按钮。

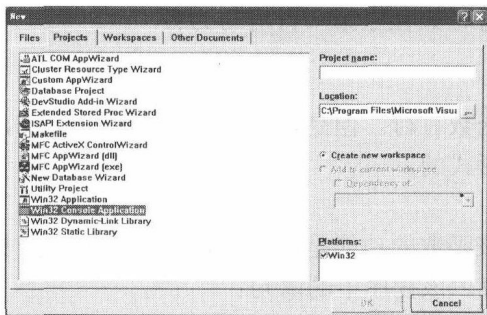


图 1—1 “New”对话框的“Project”选项卡

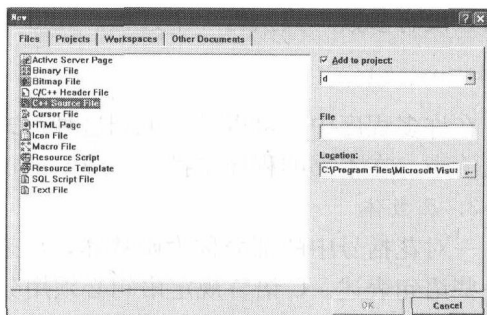



图 1—2 “New”对话框的“File”选项卡

(4) 在出现的文本窗口中输入例 1—1 中的程序。

【例 1—1】 输出字符串的简单程序。

```
#include<stdio.h>
int main(void)                /* 主程序名 */
{                               /* 表示函数的开头 */
    printf("Hello, World!\n"); /* 函数体 */
    return 0;
}                               /* 表示函数的结束 */
```

(5) 编译并运行程序。单击工具栏上的“运行程序”按钮, 若程序没有错误，就会在弹出的 MS-DOS 窗口中显示出“Hello, World!”文字；若运行程序有错误，返回到步骤(3)对程序进行修改，直至得到正确的结果。

1.3 扩展知识与理论

这里介绍一些 C 源程序的结构特征。