

C语言教学做一体化教程

主 编 王贵玲

 武汉理工大学出版社
WUTP Wuhan University of Technology Press

C 语言教学做一体化教程

主 编 王贵玲

副主编 叶惠卿 刘永明 张 毅



武汉理工大学出版社

· 武 汉 ·

图书在版编目(CIP)数据

C 语言教学做一体化教程/王贵玲主编. —武汉:武汉理工大学出版社,2018.8
ISBN 978-7-5629-5843-7

I. ① C… II. ① 王… III. ① C 语言-程序设计-教材 IV. ① TP312.8

中国版本图书馆 CIP 数据核字(2018)第 170280 号

项目负责人:彭佳佳

责任编辑:彭佳佳

责任校对:李正五

封面设计:付群

出版发行:武汉理工大学出版社

社址:武汉市洪山区珞狮路 122 号

邮编:430070

网址:<http://www.wutp.com.cn>

经销:各地新华书店

印刷:武汉兴和彩色印务有限公司

开本:787×1092 1/16

印张:10.25

字数:250 千字

版次:2018 年 8 月第 1 版

印次:2018 年 8 月第 1 次印刷

印数:1000 册

定价:38.00 元

凡购本书,如有缺页、倒页、脱页等印装质量问题,请向出版社发行部调换。

本社购书热线电话:027-87515778 87515848 87384729 87165708(传真)

· 版权所有 盗版必究 ·

前 言

C语言是目前使用最广泛的程序设计语言之一,也是计算机课程体系中的第一门重要的基础课程。它具有简洁紧凑、使用灵活方便、执行效率高等特点。本书“淡化语法、强调应用”,在书中注入了新的教学思想和方法,力争改变过去定义和规则讲授过多的弊端,从现实的具体问题入手,尽量把枯燥无味和抽象的编程语言讲得生动;让学生明白如何分析问题、解决问题,逐渐培养学生程序设计的逻辑思维能力;注重“通俗性、可接受性”的原则,把教学重点放在程序设计方法上,由例子引出语法,通过一些具体问题的程序来分析算法,介绍程序设计的基本方法和技巧,注重易读性和启发性;从最简单的问题入手,通过编写、运行程序,调试程序来掌握C语言的语法和程序设计方法,然后再分析最易出错的语法。

本书有丰富的项目实例,并与在线评测系统进行配套,为“教、学、做”一体化教学提供了保障。通过选用大量贴近生活实际的问题进行项目教学设计和在线评测系统,激发学生的学习兴趣,使学生带着真实的任务在探索中学习,增强学生主动学习的积极性。

本书由王贵玲主编、统稿并编写模块1、模块3、模块6和模块7,叶惠卿编写模块2和模块8,张毅编写模块4,刘永明编写模块5,企业一线工程师何剑云为本书的实例和练习题及知识点都给了很好的建议。感谢大连源代码软件有限公司提供的在线评测系统。在此对所有给予本书支持和帮助的同仁们致以最真挚的感谢!

本书提供微课视频、课程标准、授课计划、单元设计、教学课件PPT、思维导图、案例源代码、习题答案等丰富的数字化资源。如读者需要数字化资源请发电子邮件至287297783@qq.com(王贵玲)。

要编写一本令人满意的教材并不是一件容易的事,尽管我们非常认真和严谨,但书中难免存在疏漏之处,敬请读者指正。

编 者

2018年5月

目 录

模块 1 C 语言概述	1
1.1 C 语言概况	1
1.1.1 C 语言的发展	1
1.1.2 C 语言的特点	2
1.1.3 C 语言的未来	3
1.2 C 语言编程环境	3
1.3 C 语言的构成	6
1.3.1 函数和语句	7
1.3.2 printf()函数	8
1.3.3 scanf()函数	10
1.4 综合应用	13
模块 2 C 语言的数据类型及运算符	15
2.1 关键字与标识符	15
2.1.1 关键字	15
2.1.2 标识符	16
2.2 C 语言的基本数据类型	16
2.2.1 常量与变量	17
2.2.2 数据类型转换	23
2.3 运算符与表达式	25
2.3.1 算术运算符与算术表达式	25
2.3.2 赋值运算符与赋值表达式	27
2.3.3 关系运算符与关系表达式	29
2.3.4 逻辑运算符与逻辑表达式	30
2.3.5 条件运算符与条件表达式	33
2.3.6 逗号运算符与逗号表达式	34
2.3.7 自增自减运算符	35
2.3.8 位运算符	36
模块 3 顺序和分支	40
3.1 顺序程序结构	40

3.2 选择结构语句	41
3.2.1 if 条件语句	41
3.2.2 if 语句的嵌套	47
3.2.3 switch 条件语句	50
3.2.4 if 语句与 switch 条件语句的异同	54
3.3 综合应用	54
模块 4 循环结构	57
4.1 用 while 语句实现循环	57
4.2 用 do-while 语句实现循环	60
4.3 用 for 语句实现循环	63
4.3.1 for 语句	63
4.3.2 三种循环结构的比较	66
4.4 循环嵌套	67
4.5 break 和 continue 语句	71
4.5.1 break 语句	71
4.5.2 continue 语句	73
4.6 循环程序应用举例	75
模块 5 数组	81
5.1 数组概述	81
5.2 一维数组	82
5.2.1 一维数组的定义	82
5.2.2 一维数组的初始化	84
5.2.3 一维数组的引用	84
5.2.4 一维数组的排序	86
5.3 二维数组	90
5.3.1 二维数组的定义	90
5.3.2 二维数组的初始化	91
5.3.3 二维数组的引用	92
5.3.4 多维数组	94
5.4 字符数组与字符串	95
5.4.1 字符数组的定义与初始化	95
5.4.2 字符串	97
5.4.3 字符串的处理函数	98
5.5 数组应用实例	103

模块 6 函数	108
6.1 初识函数	108
6.1.1 函数的概念	110
6.1.2 函数定义	110
6.1.3 函数的返回值	112
6.2 函数的调用	113
6.2.1 函数的调用方法	113
6.2.2 函数调用时的数据传递	116
6.2.3 嵌套调用	117
6.2.4 递归调用	118
6.3 局部变量与全局变量	123
6.3.1 局部变量	123
6.3.2 全局变量	123
6.3.3 变量的作用域	123
模块 7 指针	124
7.1 指针与指针变量	124
7.1.1 指针的概念	125
7.1.2 指针变量的定义	125
7.1.3 指针变量的引用	126
7.1.4 指针的运算	127
7.2 指针与数组	128
7.3 指针与函数	131
7.4 指针与字符串	133
7.4.1 字符串的表示形式	133
7.4.2 字符串的指针变量	134
7.4.3 字符串指针变量做函数参数	135
模块 8 结构体和共用体	138
8.1 结构体类型和结构体变量	138
8.1.1 结构体类型的定义	138
8.1.2 结构体变量的定义	139
8.1.3 结构体变量的初始化	140
8.1.4 结构体变量的引用	141
8.2 结构体数组	143
8.2.1 结构体数组的定义	143

8.2.2 结构体数组的初始化	143
8.2.3 结构体数组的引用	144
8.3 结构体指针变量	144
8.3.1 结构体指针变量	144
8.3.2 结构体数组指针	145
8.4 共用体	146
8.4.1 共用体数据类型的定义	146
8.4.2 共用体变量的定义	147
8.4.3 共用体变量的初始化和引用	148
8.5 typedef——给数据类型取别名	149
附录 A 程序调试常见错误信息	151
参考文献	155

模块 1 C 语言概述

【模块介绍】

本章主要介绍 C 语言的发展和特点;C 语言的基本结构;C 语言编程环境及上机执行的过程;C 语言的代码风格。

【知识目标】

1. 了解 C 语言的发展历程、特点;
2. 掌握 C 语言的编程环境和执行过程;
3. 掌握 C 语言的代码风格。

【技能目标】

1. 通过学习 C 语言概况,学生能够对 C 语言的历史、现状和未来有所了解;
2. 通过学习 C 语言编程环境和执行过程,学生能够熟悉 C 语言编程环境及 C 语言源代码的执行过程,提高程序调试能力和排错能力。

【素质目标】

1. 培养学生认真负责的工作态度和严谨细致的工作作风;
2. 培养学生自主学习探索新知识的意识;
3. 培养学生的团队协作精神;
4. 培养学生的诚实守信意识和职业道德。

1.1 C 语言概况

1.1.1 C 语言的发展

C 语言是一种强大的专业化编程语言,深受业余和专业编程人员的欢迎。在学习之前应先了解和认识它。C 语言的原型是 A 语言(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 Richards 对 CPL 语言进行了简化,于是产生了 BCPL 语言。

1969 年,美国贝尔实验室的 Ken Thompson 将 BCPL 语言进行了修改,提炼出它的精华,并为它起了一个有趣的名字“B 语言”。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,他取了 BCPL 的第二个字母作为这种语言的名字,这就是 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》,即著名的 ANSI C。

1978 年由 AT&T(美国电话电报公司)贝尔实验室正式发表了 C 语言。同时 Brian W. Kernighan 和 Dennis M. Ritchie 出版了名著 *The C Programming Language* 一书。通常简称为 K&R,也有人称之为 K&R 标准。但是,在 K&R 中并没有定义一个完整的标准 C 语言。后来由美国国家标准协会(American National Standards Institute, ANSI)在此基础上制定了一个 C 语言标准,并于 1983 年发表,通常称之为 ANSI C,从而使 C 语言成为目前世界上应用最广泛的高级程序设计语言之一。

由于 C 语言的不断发展,1987 年,美国国家标准协会在综合各种 C 语言版本的基础上,又颁布了新标准,为了与标准 ANSI C 区别,称为 87 ANSI C。1990 年,国际标准化组织 ISO 接受了 87 ANSI C 作为 ISO 标准。这是目前功能最完善、性能最优良的 C 语言版本。目前流行的 C 语言编译系统都是以此为基础的。本书讲述的内容是以 ANSI C 为基础的。

1.1.2 C 语言的特点

在过去的几十年里,C 语言已经成为世界上最流行、最重要的一种编程语言。为什么它为计算机界人士所广泛接受呢?因为 C 语言是一种融合了控制特性的现代语言,而人们已发现在计算机科学的理论和实践中,控制特性是很重要的。其设计使得用户可以自然地采用自顶向下的规划,结构化的编程以及模块化的设计。这种做法使得编写出的程序更可靠、更易懂。它的主要特点如下。

1. 高效性

在设计上它充分利用了当前计算机在性能上的优点。C 程序往往很紧凑且运行速度快。事实上,C 语言可以表现出通常只有汇编语言才具有的精细控制能力(汇编语言是特定的 CPU 设计所采用的一组内部指令的助记符。不同的 CPU 类型使用不同的汇编语言)。如果愿意,编程人员可以细调程序以获得最大速度或最大内存使用率。

2. 可移植

C 语言是一种可移植语言。这意味着,在一个系统上编写的 C 程序经过很少改动或不经修改就可以在其他系统上运行。如果修改是必要的,则通常只改变伴随主程序的一个头文件中的几项内容即可。多数语言原本都想具有可移植性,但任何曾将 IBM PC BASIC 程序转换为 Apple BASIC 程序(它们还是近亲)的人,或者试图在 UNIX 系统上运行一个 IBM 大型机 FORTRAN 程序的人都知道,移植至少是在制造麻烦。C 语言在可移植性方面处于领先地位。C 编译器(将 C 代码转换为计算机内部使用的指令的程序)在 40 多种系统上可用,包括从使用 8 位微处理器的计算机到 Cray 超级计算机。不过要

知道,程序中为访问特定硬件设备(例如显示器)或操作系统(如 Windows XP 或 OS X)的特殊功能而专门编写的部分,通常是不能移植的。由于 C 语言与 UNIX 的紧密联系,UNIX 系统通常都带有一个 C 编译器作为程序包的一部分。Linux 中同样也包括一个 C 编译器。个人计算机,包括运行不同版本的 Windows 和 Macintosh 的 PC,可使用若干种 C 编译器。所以不论使用的是家用计算机、专业工作站还是大型机,都很容易得到针对特定系统的 C 编译器。

3. 强大的功能和灵活性

C 语言强大而又灵活(计算机世界中经常使用的两个词)。例如,强大而灵活的 UNIX 操作系统的大部分便是用 C 语言编写的。其他语言(如 FORTRAN、Perl、Python、Pascal、LISP、Logo 和 BASIC)的许多编译器和解释器也都是用 C 语言编写的。结果是,当在一台 UNIX 机器上使用 FORTRAN 时,最终是由一个 C 程序负责生成最后的可执行程序。C 程序已经用于解决物理学和工程学问题,甚至用来为《角斗士》这样的电影制造特殊效果。

4. 面向编程人员

C 语言面向编程人员的需要。它允许编程人员访问硬件,并可以操纵内存中的特定位置。它具有丰富的运算符供选择,可让编程人员简洁地表达自己的意图。在限制编程人员所能做的事情方面,C 语言不如 Pascal 这样的语言严格。这种灵活性是优点,同时也是一种危险。优点在于:许多任务(如转换数据形式)在 C 语言中都简单得多。危险在于:使用 C 语言时,编程人员可能会犯在使用其他一些语言时不可能犯的错误。C 语言给予编程人员更多的自由,但同时也让编程人员承担更大的风险。多数 C 语言实现都有一个大型的库,其中包含有用的 C 函数。这些函数能够处理编程人员通常会面对的许多需求。

1.1.3 C 语言的未来

越来越多的计算机用户已使用 C 语言以便利用其优点。

不管 C++ 和 Java 还是其他较新的语言如何流行,C 语言在软件产业中仍然是一种重要的工具,在最想获得的技能中,它一般都列在前 10 名。C 语言也一直位列主流编程语言的三甲!特别是在嵌入式系统的编程中,C 语言一直占据主导地位。

最后,由于它是一种适合用来开发操作系统的语言,C 语言在 Windows 以及 Linux 的开发中也扮演着重要的角色。因此,在未来几十年当中,C 语言还将继续强势!

1.2 C 语言编程环境

编写出的 C 语言源代码如何在计算机上进行调试运行直到得到正确的运行结果呢?

本书中 C 程序的编辑、调试及运行采用 Microsoft Visual C++ 6.0(以下简称为 VC++ 6.0)开发平台,读者也可以采用其他的开发平台,如 dev CPP。

要运行一个 C 程序一般都要经过编辑、编译、链接、运行四个步骤,如图 1-1 所示。



图 1-1 C 程序执行过程

将 C 语言代码经过编译、组建(也称链接)、运行才可以看到输出结果。编译是将 C 语言源代码“翻译”成机器码(0、1 代码),生成 .cpp 文件,然后再将工程所需要的所有资源集合在一起,生成 .exe 文件,最后运行了 .exe 文件就可以看到结果。详细执行步骤如下:

1. 进入 VC

开始→程序→Microsoft Visual C++ 6.0→Microsoft Visual C++ 6.0,如图 1-2 所示。

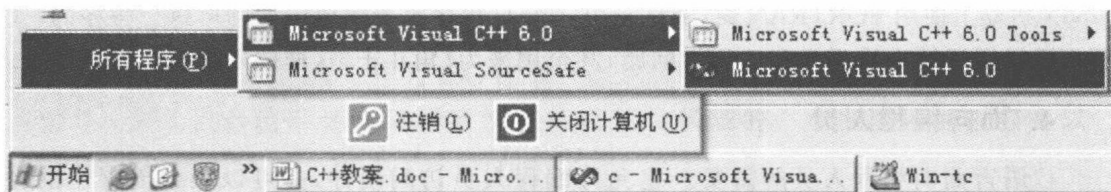


图 1-2 打开 VC++6.0 方法示意图

2. 新建工程

文件→新建→WIN32 Console Application→输入工程名(如 Project1)→确定,如图 1-3 所示。

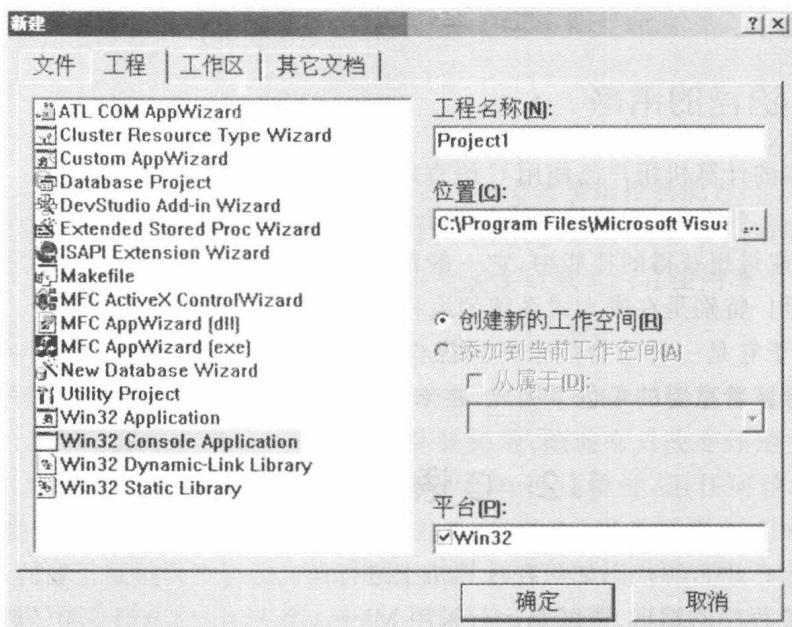


图 1-3 新建工程

3. 新建文件

文件→新建→C++ Source File→输入文件名(如 F1.c,默认扩展名 *.CPP)→确定,如图 1-4 所示。

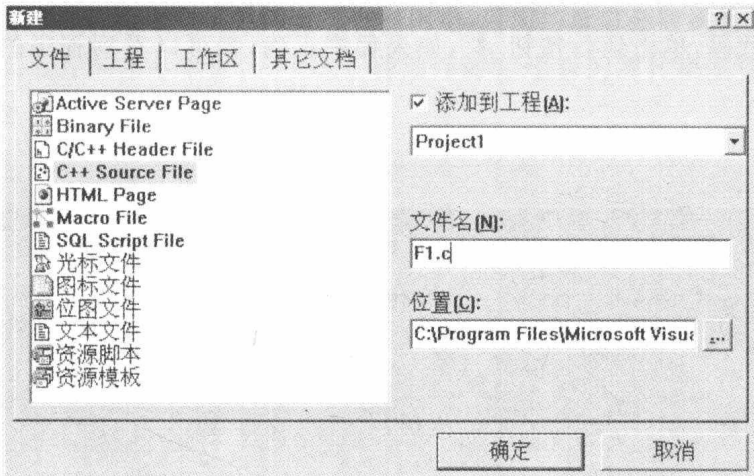


图 1-4 新建 C 语言源文件

4. 在工作空间中可以看到刚才创建的工程和源文件(图 1-5)

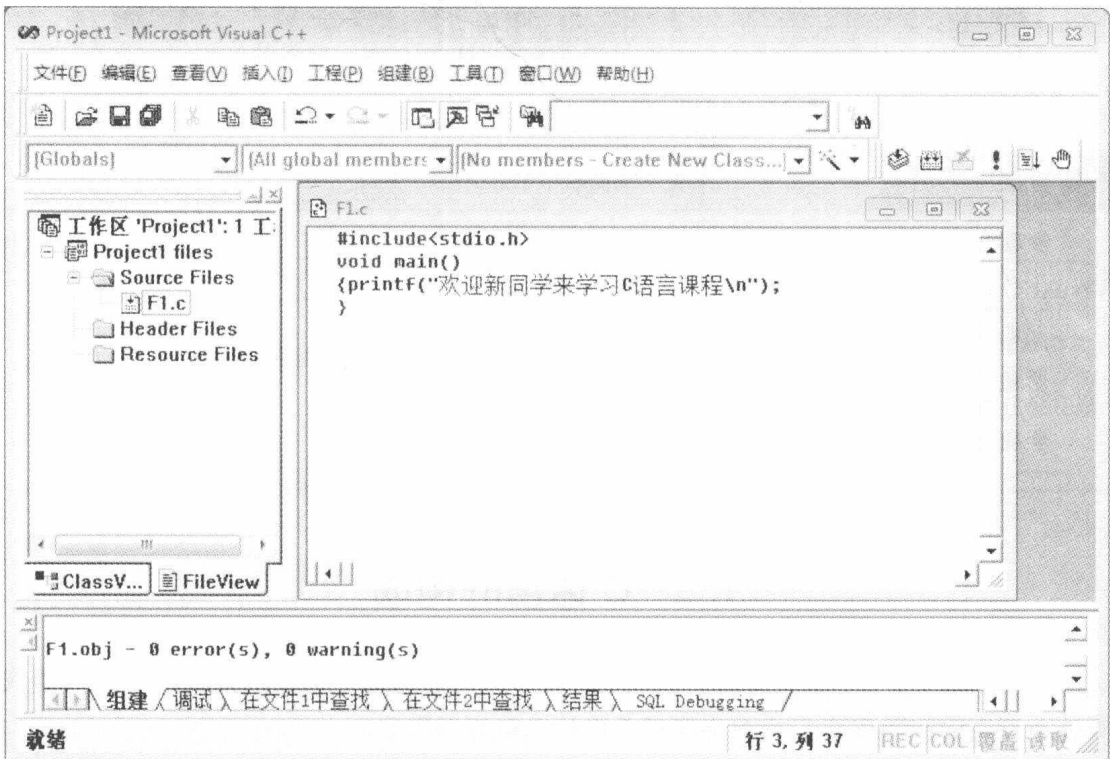




图 1-5 C 语言源程序编辑界面

5. 编译、运行代码

对于初学者,最常用的是按运行按钮 。编写完代码,一次单击就可以看到输出结果。也可以使用快捷键:Ctrl+F7(编译)、Ctrl+F5(运行)或 F7(组建)。

编写完源代码,单击运行按钮 ,如果程序正确,就可以看到程序的运行结果,如图 1-6 所示。

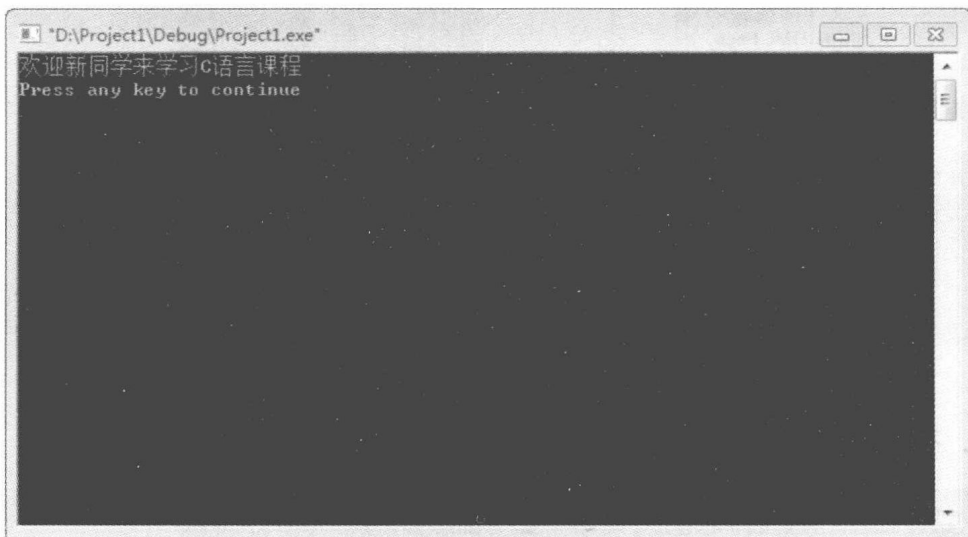



图 1-6 运行结果

体验编程环境

步骤 1:启动 VC++6.0,开始→程序→Microsoft Visual C++ 6.0→Microsoft Visual C++ 6.0 命令,进入编程环境。

步骤 2:新建 F1.c。

步骤 3:在编辑窗口录入源程序,并保存。

步骤 4:单击运行按钮 ,输出结果如图 1-6 所示。

1.3 C 语言的构成

在前面讲了一个 F1.c 完成了一个简单的输出。那么这个简单的小程序的代码都代表什么意思呢?简单的 C 语言程序都是由什么组成的呢?

1.3.1 函数和语句

1. 什么是函数

对初学者来说现在理解函数的含义有点困难,后面的章节会对函数进行详细讲解。

函数是 C 程序的基本单位,一个能正确编译的 C 语言的源代码文件,它里面至少包含了一个函数。

函数名所在的一行为函数头,{}括起来的内容称为函数体。函数的大致格式如下:

```
函数类型  函数名(...)  
{  
...  
}
```

例如:

```
void main()  
{  
...  
}
```

前面编写的 F1.c 程序里面有一个 main()函数,如图 1-7 所示。

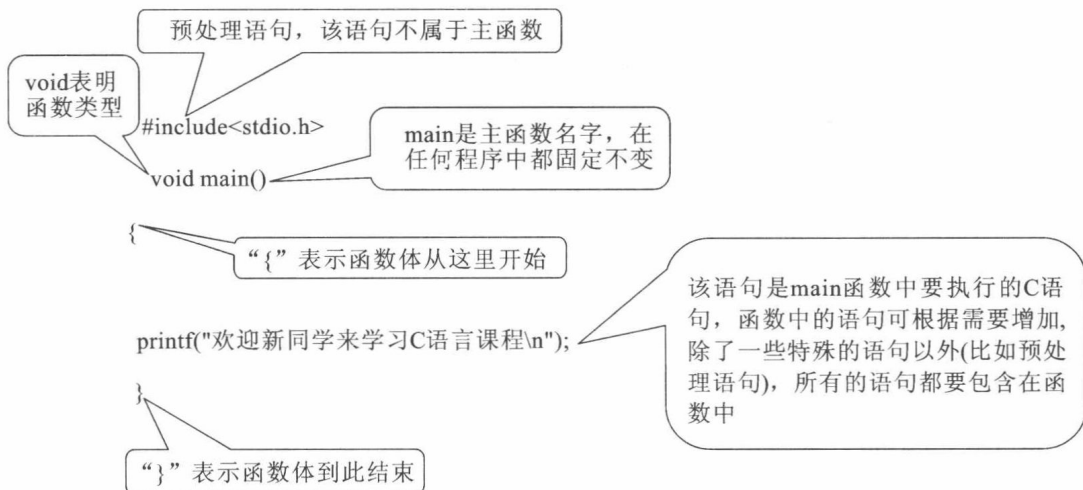


图 1-7 F1.c 结构分析图

一个 C 程序必须有且只能有一个主函数,即 main()函数。不管一个程序包含多少个函数,也不管 main()函数躲在哪个角落,在运行该程序时,系统都会千方百计地找到 main()函数,从 main()函数开始运行。所以我们把 main()函数称作程序的“入口函数”。

main()函数可以调用其他函数,其他函数不能调用 main()函数。

2. C 语言书写格式

C 程序是由函数组成的,而函数是由各种语句构成的。在书写 C 语言的语句时,要遵从以下规定:

- (1) 一条完整的语句必须以英文分号结束。
- (2) 一行可以写多条语句。
- (3) 一条语句也可以占多行,但在何处断行很重要。

比如:下面的两个 printf 语句同样是占据两行。哪个运行是正确的呢?

```
printf
("大家好\n"); ✓
printf("
大家好\n"); ✗
```

3. C 语言注释语句

在 C 语言源代码中,如果某些内容以 /* 开始并以 */ 结束,则编译系统会把这之间的内容作注释处理,对它不进行语法检查——这些语句称为注释语句。注释语句不执行,只是便于用户对程序的阅读理解,对编译和运行不起作用,注释语句可以出现在程序中的任意位置。具体如下:

(1) 单行注释

单行注释通常用于对程序中某一行代码进行解释,用“//”符号表示。

```
printf("Hello, world\n"); // 输出“Hello, world”
```

(2) 多行注释

多行注释就是在注释中的内容可以为多行,它以符号“/*”开头,并以“*/”结束。

```
/*      printf("Hello, world\n");
      return 0;      */
```

1.3.2 printf()函数

printf()函数是 C 语言系统提供的一个基本输出函数,它的函数原型包含在 stdio.h 头文件里面。它的功能是输出一些指定的内容,在前面的程序中通过此函数 printf(“欢迎新同学学习 C 语言课程”);构成了输出语句,因此输出了“欢迎新同学学习 C 语言课程”这条语句。

1. printf()函数的一般格式

```
printf(格式输出控制,输出列表);
```

- (1) 格式输出控制:需用一对双引号括起来,双引号内包含“格式说明”和“普通字符”

两部分。

(2) 输出列表:需要输出的数据,多个数据之间用逗号隔开。

例如:

```
printf("%d,%o\n",m,n);  
printf("m=%d,n=%o\n",m,n);
```

2. 格式说明

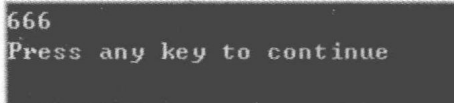
(1)“格式说明”由“%”和格式字符组成,如%d,%c,等。它的作用是把输出数据转换为指定格式输出。

注:在 printf()函数中,若格式说明的个数少于输出项的个数,多余的输出项则忽略不输出。

【例 1-1】 以下程序:

```
#include<stdio.h>  
void main( )  
{  
    int a=666,b=888;  
    printf("%d\n",a,b);  
}
```

运行结果(图 1-8):



```
666  
Press any key to continue
```

图 1-8 例 1-1 运行结果

(2) 可以在%和格式字符之间加入一个整数来控制输出数据所占的宽度,如果整数指定的宽度大于实际输出数据的宽度,数据的输出采用右对齐的方式,左边自动补空格;反之,则以输出数据的实际宽度输出。

【例 1-2】

输出语句	输出结果
printf("%d",258);	258
printf("%2d",258);	258
printf("%4d",258);	□ 258
printf("%f",1.25);	1.250000
printf("%10f",1.25);	□□1.250000

(3) 当在%和格式字符f之间加入“整数 1. 整数 2”来控制输出数据的格式时,“整数 1”用于指定输出数据占的总宽度,“整数 2”用于指定输出实数的小数部分的个数。