

高职高专计算机基础教育精品教材



普通高等教育“十一五”国家级规划教材

前一版 荣获“2011年北京高等教育精品教材”称号
荣获“2009年高职高专计算机类专业优秀教材”称号

C语言程序设计 (第2版)

崔武子 主 编 李红豫 副主编

清华大学出版社



普通高等教育“十一五”国家级规划教材
高职高专计算机基础教育精品教材

C 语言程序设计(第2版)

崔武子 主 编
李红豫 副主编

清华大学出版社
北 京

内 容 简 介

本书第1版于2008年出版。该书是普通高等教育“十一五”国家级规划教材,荣获“2011年北京高等教育精品教材”和“2009年高职高专计算机类专业优秀教材”称号。本书力求突破高职高专旧的教学限制,用全新的方法组织编写。本书始终以应用为目的,从应用入手,采用了任务驱动方式。书中内容讲授精练,循序渐进,重点突出,易于理解。本书将公交一卡通管理程序作为贯穿全书知识点的实例在各章中分别介绍,使学生尽早体会较大程序的编写过程。全书共8章,分别是C语言基础与顺序结构、分支结构、循环结构、数组、指针、函数、结构体和文件。

本书还提供免费的授课素材,包括丰富的电子教案(含动画)和全书的源代码。全书程序的运行环境是 Visual C++ 6.0。

本书主要针对C语言零起点的高职高专学生和自学者,也适合C语言的初学者学习。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C语言程序设计/崔武子主编.--2版.--北京:清华大学出版社,2013

高职高专计算机基础教育精品教材

ISBN 978-7-302-32060-9

I. ①C… II. ①崔… ②李… III. ①C语言—程序设计—高等职业教育—教材 IV. ①TP312

中国版本图书馆CIP数据核字(2013)第078953号

责任编辑:张龙卿

封面设计:徐日强

责任校对:刘 静

责任印制:沈 露

出版发行:清华大学出版社

网 址: <http://www.tup.com.cn>, <http://www.wqbook.com>

地 址:北京清华大学学研大厦A座 邮 编:100084

社总机:010-62770175 邮 购:010-62786544

投稿与读者服务:010-62776969, c-service@tup.tsinghua.edu.cn

质量反馈:010-62772015, zhiliang@tup.tsinghua.edu.cn

课 件 下 载: <http://www.tup.com.cn>, 010-62795764

印 装 者:北京嘉实印刷有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:15.75 字 数:357千字

版 次:2008年3月第1版 2013年8月第2版 印 次:2013年8月第1次印刷

印 数:1~3000

定 价:33.00元

前 言

作者在围绕“教师方便教,学生容易学”为主题开展的一系列探索与实践活动后,以C语言程序设计零起点学习者作为主要对象编写了本教材。本书于2008年出版了第1版,该书是普通高等教育“十一五”国家级规划教材,前一版荣获“2011年北京高等教育精品教材”和“2009年高职高专计算机类专业优秀教材”称号。

随着我国现代化建设的需要和社会的不断发展,高职高专教育显得越来越重要,对相应教材的要求也越来越高。为了破除旧的限制,写出独具一格的、体现高职高专教育新理念和教学特点的教材,我们本着体系得当、循序渐进、台阶要小、分解难点以及通俗易懂、例题丰富、易于理解的原则,正确选择典型任务,选好切入点,在编写此书的过程中力求做到从实际到理论、从具体到抽象、从个别到一般、从零散到系统,注意培养学生的学习能力、工作能力和创造能力。本书具有如下特点。

1. 精简教学内容

考虑到各学校的课时限制和高职高专学生的实际情况,内容上不贪多求全,合理舍去不常使用的内容,但对于C语言的基本内容予以细致的介绍,做到重点突出,易于理解。

2. 采用任务驱动方式

以“应用为目的,从应用入手”的原则,将各章所讲授内容都按任务驱动的方式组织,并在实现其任务的过程中,逐步引进新的知识点。对于每个任务通过“问题提出→编程思路→程序代码→运行结果→归纳分析”这几个步骤分步完成。

3. 教学难点适中,增加趣味性

采用简单易懂的实例降低教学难度,强调实用性和趣味性,激发学生的学习积极性,并使学生在解决问题的过程中获得成就感,对学习有信心。力求把复杂的问题简单化,采用生动活泼的风格和语言讲授所有内容,因此本书也适合初学者自学。

4. 强调学习方法

只看懂别人的程序或只运行现有的程序是永远学不会编程的,为了使通过有限的实例学会解决新问题的方法,本书对于每一个任务或实例首先给出编程思路,再给出解决的方案,并进行了归纳分析,从而起到举一反三的作用。

5. 举用实例贯穿整个知识点

将公交一卡通管理程序分七部分(分散在各章)介绍,各部分程序随着讲授内容的增多逐步完善,用此方法可使学生明确学习目标,尽早体会较大程序的编写过程。

6. 加强互动教学

书中提供的讨论题和思考题以及丰富的电子教案(含动画)有利于培养学生的实践能力和创新精神,同时还可加强课内互动教学。本书提供的免费素材包括电子教案和全书的源代码,源代码均在 Visual C++ 6.0 环境下运行通过。

全书由崔武子主编并统稿,李红豫任副主编。

写一本书不容易,写一本好书更不容易,虽然我们尽力写好高职高专特色的优秀教材,但限于作者水平有限,书中难免有不足之处,恳请读者批评和指正。

作 者

2013 年 3 月于北京

目 录

第 1 章	C 语言基础与顺序结构	1
1.1	认识 C 语言程序	1
1.1.1	了解 C 语言程序的构成	1
1.1.2	熟悉主函数框架	3
1.2	合理选用数据类型	7
1.2.1	合理选用整型数据	7
1.2.2	合理选用实型数据	8
1.2.3	合理选用字符型数据	10
1.3	学会使用常用运算符	11
1.3.1	学会使用算术运算符	11
1.3.2	学会使用赋值运算符	13
1.4	学会控制输入输出数据	14
1.4.1	使用格式输入输出函数出加法题	14
1.4.2	单个字符输入输出函数的使用	15
1.5	画顺序结构的流程图	16
1.6	贯穿教学全过程的实例——公交一卡通管理程序 (1)	17
1.7	本章总结	19
	思考题	21
	上机练习	21
第 2 章	分支结构	22
2.1	if 语句	22
2.1.1	学习使用关系运算符和 if 语句	22
2.1.2	认识省略 else 的 if 语句	24
2.1.3	学会使用逻辑运算符	27
2.1.4	学会使用嵌套的 if 语句	29
2.1.5	学会使用 if-else if 形式的嵌套 if 语句	31
2.2	switch 语句	32
2.2.1	认识 switch 语句	32
2.2.2	多个 case 语句相同情况的处理	35
2.2.3	用 switch 语句的技巧计算分段函数的值	36

2.3	用条件运算符转换大小写字母	38
2.4	程序举例	39
2.4.1	掌握三个数中求最大数的方法	39
2.4.2	掌握三个数排序的方法	39
2.4.3	熟悉菜单设计操作	40
2.5	贯穿教学全过程的实例——公交一卡通管理程序 (2)	41
2.6	本章总结	43
	思考题	44
	上机练习	45
第 3 章	循环结构	46
3.1	使用 for 语句实现循环控制	46
3.1.1	使用 for 语句重复显示信息	46
3.1.2	使用 for 语句重复出加法题	48
3.2	使用 while 语句实现循环控制	50
3.2.1	使用 while 语句为学生分班	51
3.2.2	使用 while 语句计算近似值	53
3.3	使用 do-while 语句实现循环控制	54
3.3.1	使用 do-while 语句计算加法题正确率	54
3.3.2	使用 do-while 语句编写打字练习程序	57
3.4	使用 break 语句强行退出循环	58
3.5	学会使用循环设计典型算法	60
3.5.1	学会找出 Fibonacci 数列的各项,掌握递推算法	60
3.5.2	用迭代算法求解某数的平方根	62
3.5.3	用枚举算法求解百元百鸡问题	64
3.5.4	学会判断质数的算法	68
3.6	贯穿教学全过程的实例——公交一卡通管理程序 (3)	71
3.7	本章总结	73
	思考题	75
	上机练习	76
第 4 章	数组	77
4.1	认识数组	77
4.2	使用一维数组	77
4.2.1	定义与引用一维数组	77
4.2.2	在字符串中找出数字字符构造新数组	79
4.2.3	判断密码是否正确	80
4.2.4	求一批数据中的最大值	81
4.2.5	在有序数据中插入一个数	83

4.2.6	排序数据	85
4.3	使用二维数组	88
4.3.1	求两个矩阵的和	88
4.3.2	求方阵对角线上元素之和	90
4.3.3	显示算术题和学生答题信息	91
4.3.4	编写不同级别的打字练习程序	93
4.3.5	统计一个学习小组的成绩	94
4.4	贯穿教学全过程的实例——公交一卡通管理程序 (4)	96
4.5	本章总结	103
	思考题	104
	上机练习	105
第 5 章	指针	106
5.1	认识变量的地址和指针变量	106
5.2	通过指针访问普通变量	106
5.3	通过指针访问数组	108
5.3.1	通过指针计算总分	108
5.3.2	通过指针将数据逆置	110
5.3.3	通过指针找出最大值	112
5.3.4	通过指针排序数据	114
5.3.5	通过指针计算两个矩阵的和	114
5.4	通过指针访问字符串	116
5.4.1	通过指针判断回文	116
5.4.2	在三个字符串中找出最大的字符串	119
5.4.3	将三个字符串从大到小进行排序	120
5.4.4	连接两个字符串	121
5.5	本章总结	123
	思考题	124
	上机练习	124
第 6 章	函数	127
6.1	了解 C 语言程序的执行过程	127
6.2	掌握自定义函数的编写与调用方法	130
6.2.1	调用自定义函数计算 1~n 的和	130
6.2.2	调用自定义函数进行四则运算	134
6.2.3	在被调函数中交换数据	137
6.2.4	用嵌套调用的方法进行计算	139
6.3	调用自定义函数处理数组	140
6.3.1	调用自定义函数输入输出一维数组	140

6.3.2	调用自定义函数将数据逆置	142
6.3.3	调用自定义函数求最大值	143
6.3.4	调用自定义函数判断回文	145
6.3.5	调用自定义函数计算两个矩阵的和	147
6.4	变量的存储类别	148
6.4.1	内部变量和外部变量	148
6.4.2	动态存储变量和静态存储变量	150
6.5	贯穿教学全过程的实例——公交一卡通管理程序 (5)	152
6.6	本章总结	160
	思考题	162
	上机练习	163
第 7 章	结构体	165
7.1	了解结构体类型数据的使用场合	165
7.2	掌握结构体变量的使用方法	165
7.2.1	使用结构体变量处理实际问题	166
7.2.2	将结构体变量作为实参处理实际问题	170
7.3	掌握结构体数组的使用方法	171
7.3.1	使用结构体数组处理实际问题	171
7.3.2	将结构体数组名作为实参处理实际问题	173
7.4	贯穿教学全过程的实例——公交一卡通管理程序 (6)	181
7.5	本章总结	189
	思考题	191
	上机练习	193
第 8 章	文件	195
8.1	了解文件的处理过程	195
8.2	掌握文件的基本操作本领	197
8.2.1	创建文本文件	197
8.2.2	读取文本文件中的数据	199
8.2.3	创建二进制文件	200
8.2.4	读取二进制文件中的数据	201
8.3	文件的应用举例	203
8.3.1	编写算术考试程序	203
8.3.2	编写阅卷程序	204
8.3.3	复制文件	205
8.3.4	调用函数修改文件中的内容	207
8.4	贯穿教学全过程的实例——公交一卡通管理程序 (7)	209
8.5	本章总结	231

思考题	232
上机练习	232
附录	
附录 A C 语言关键字	234
附录 B 常用字符与 ASCII 代码对照表	234
附录 C 运算符的优先级和结合方向	235
附录 D 常用 C 库函数	235
参考文献	239

第1章 C语言基础与顺序结构

学习目标

1. 合理选用数据类型
2. 使用算术运算符和赋值运算符
3. 控制数据的输入、输出
4. 使用顺序结构以及顺序结构的流程图
5. 使用 Visual C++ 6.0 集成环境
6. 转换大小写字母的方法
7. 产生随机整数的方法

计算机由硬件系统和软件系统组成,其中硬件是物质基础,而软件是计算机的灵魂。没有软件的计算机是什么也干不了的“裸机”。所有软件要用计算机语言编写。

计算机语言是人和计算机交换信息的工具。随着计算机技术的发展,计算机语言逐步得到完善。最初使用的计算机语言是用一串串 0 和 1 表达的语言——机器语言,后来使用的计算机语言是用简洁的英文字母或符号串表达的语言——汇编语言。机器语言和汇编语言都是低级语言,用这种语言编写的程序执行效率高,但程序代码很长,又都依赖于具体的计算机,因此编码、调试、阅读程序都很困难,通用性也差。

目前使用最广泛的计算机语言是用接近于人们自然语言表达的语言——高级语言。用高级语言编写的程序完全不依赖于计算机硬件,编码相对短,可读性强。C 语言属于高级语言。用高级语言编写的程序叫做源程序。

1.1 认识 C 语言程序

1.1.1 了解 C 语言程序的构成

任何一个 C 语言程序都是由若干个函数构成的,所以编写一个程序的过程,就是根据功能要求并按照 C 语言语法规则逐个编写各函数的过程。下面给出一个较完整的程序,以便读者尽早了解 C 语言程序。程序中的各行含义将在后续章节中详细介绍,在实例 6.1 中给出其执行过程。

【实例 1.1】 观察下面的程序,认识一个完整的 C 语言程序,了解 C 语言程序的结构。

```
// 下面 3 行是预处理命令部分
#include <stdio.h>
#include <math.h>
```

```
#define PI 3.14159
```

```
// 下面 2 行是函数的原型说明部分
```

```
double sup_area(double r);
```

```
double volume(double r);
```

```
// 下面是主函数部分
```

```
main()
```

```
{ double a=-5,b,c,d;
```

```
    b=fabs(a);
```

```
    c=sup_area(b);
```

```
    d=volume(b);
```

```
    printf("c=%lf,d=%lf\n",c,d);
```

```
}
```

```
// 下面是 sup_area 函数的定义部分,函数功能是计算球的表面积
```

```
double sup_area(double r)
```

```
{ double s;
```

```
    s=4*PI*r*r;
```

```
    return s;
```

```
}
```

```
// 下面是 volume 函数的定义部分,函数功能是计算球的体积
```

```
double volume (double r)
```

```
{ double v;
```

```
    v=4.0/3.0*PI*r*r*r;
```

```
    return v;
```

```
}
```

1. 运行结果

```
c=314.159000,d=523.598333
```

2. 归纳分析

(1) 任何一个 C 语言程序都由若干个函数构成,而且必须有且仅有一个主函数(函数名必须是 main),其他函数的多少由实际情况而定,处理简单问题时也可以没有其他函数。本程序包括主函数、sup_area 函数和 volume 函数。

(2) 程序中用“//”引导的部分叫做注释部分。注释部分对程序的运行不起作用。

在程序中加入注释是为了便于阅读。

(3) C 语言提供丰富的标准库函数以便直接使用,但要求在程序的开头加上包含该函数信息的命令行。本程序中使用了库函数 `printf` (功能是输出数据,是输出函数)和 `fabs` (功能是求绝对值,是数学函数)。C 语言系统将所有输入、输出函数的信息存放在“`stdio.h`”文件中,而将所有数学函数的信息存放在“`math.h`”文件中,所以程序的开头加了两个命令行“`#include <stdio.h>`”和“`#include <math.h>`”。

(4) 程序的开头除了上述两个命令行外,又有了命令行“`#define PI 3.14159`”。有此命令行后,程序中所用到的所有 `PI` 均用 `3.14159` 代替。

(5) 程序中编写了 3 个函数,除了主函数外,在使用其他自定义的函数前,应对这些函数逐一进行函数原型说明。因此程序中加了“`double sup_area(double r);`”和“`double volume(double r);`”。

1.1.2 熟悉主函数框架

在 C 语言中可以编写程序进行算术运算,就像日常生活中人们使用计算器计算一样。

【实例 1.2】 编写程序,计算两个数的和与差,要求从键盘输入两个数。

1. 编程思路

在 C 语言中,数据的输入操作使用标准库函数 `scanf` 实现,而通过标准库函数 `printf` 实现输出功能。加法操作和减法操作分别使用算术运算符“+”和“-”。

2. 程序代码

```
#include <stdio.h>
main()
{   int x,y,a;                        // 定义三个变量

    printf("Input x and y:");        // 显示提示信息
    scanf("%d%d",&x,&y);             // 要求从键盘输入两个整数
    a=x+y;                           // 计算两个数的和
    printf("The sum of the two numbers:%d\n",a); // 输出两个数的和

    a=x-y;                           // 计算两个数的差
    printf("The difference:%d\n",a);   // 输出两个数的差
}
```

3. 运行结果

```
Input x and y:1200 180
The sum of the two numbers:1380
The difference:1020
```

4. 归纳分析

(1) 本程序只包含主函数,主函数的一般框架是:

```
main()
```

```

{  定义变量部分
    功能语句部分
}

```

定义变量和功能语句部分均可以是多条,而且每条都以“;”结束。

(2) 在屏幕上显示内容要使用 printf 函数。此函数有两种形式。参见实例 1.8。

语句“printf(“Input x and y:”);”的功能是屏幕上显示字符串“Input x and y:”,而语句“printf(“The sum of the two numbers:%d\n”,a);”的功能是在显示双引号中字符的同时,“%d”的位置上用 a 的值替换。“%d”是格式说明符,在输出整数时使用,“\n”是换行符。

(3) 从键盘输入数据要使用 scanf 函数。参见实例 1.8。

语句“scanf(“%d%d",&x,&y);”的功能是要求用户从键盘输入两个整数,并把它们分别存放在变量 x 和 y 中。在输入整数时也要使用格式说明符“%d”。

(4) 编写程序后应上机验证。学习者可以按照如下顺序尝试上机。更多的上机操作方法将在后续章节中陆续介绍。

5. 操作步骤

第 1 步: 安装 Visual C++ 6.0。如果已经安装,则跳过此步。

第 2 步: 启动 Visual C++ 6.0。为了编写 C 语言程序,首先应启动 Visual C++ 6.0 集成环境,其方法是:在 Windows 的“开始”菜单中,依次选择“程序”|“Microsoft Visual Studio 6.0”|“Microsoft Visual C++6.0”,此时弹出 Visual C++ 6.0 的主窗口。

第 3 步: 新建 C 源程序。在 Visual C++ 6.0 的主窗口选择“文件”|“新建”命令,出现“新建”对话框,在此选择“文件”选项卡下的 C++ Source File,在右侧“位置”栏中输入“D:\C 语言”或通过单击按钮并选择 D 盘上的“C 语言”文件夹,再在“文件名”栏中输入“实例 1_2.c”(如图 1.1 所示),然后单击“确定”按钮,便建立了新的 C 源程序,如图 1.2 所示。该界面的右侧为编辑窗口,这时编辑窗口是空的。

第 4 步: 编辑源程序。在编辑窗口中输入实例 1.2 的代码。可以不输入注释部分。

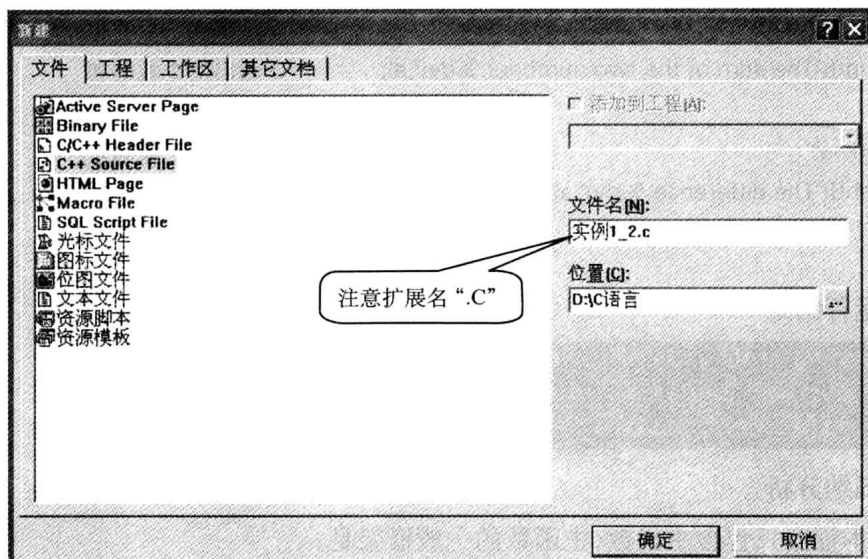


图 1.1 “新建”对话框

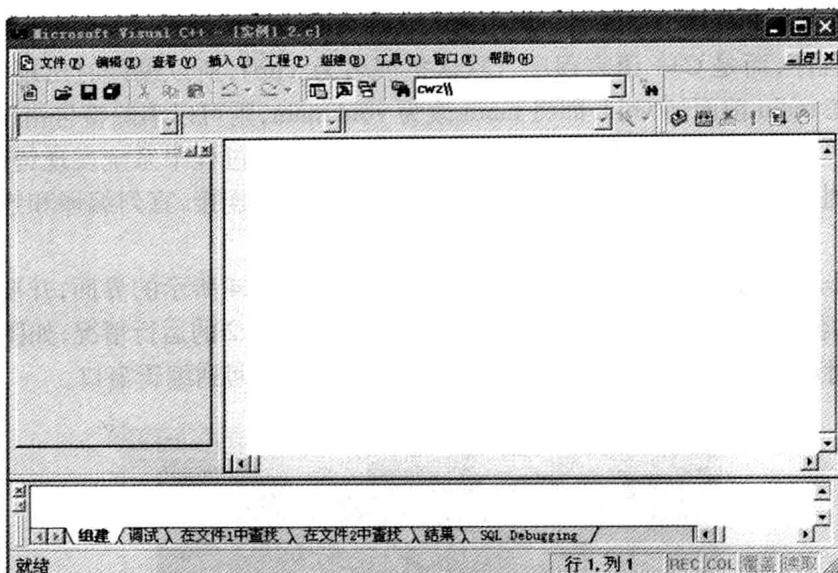


图 1.2 新建 C 源程序后的界面

第 5 步：保存文件。因为上机时经常会发生预料不到的事情，一定要养成随时存盘的好习惯。单击工具栏中的“保存”按钮，按原名将文件存盘。

第 6 步：编译和连接程序。C 源程序通过编译和连接之后才能运行。单击“编译”按钮，进行程序的编译（在出现的提示信息框中单击“是”按钮）。系统在编译前会自动将程序保存，然后进行编译。如果在编译的过程中发现错误，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第 1 行，则光标自动跳到代码的错误行。修改该错误后，对程序重新进行编译，若程序还有错误或警告，可继续修改和编译，直到没有错误为止。编译本实例时没有出现任何错误和警告，所以错误和警告数都为 0，如图 1.3 所示。

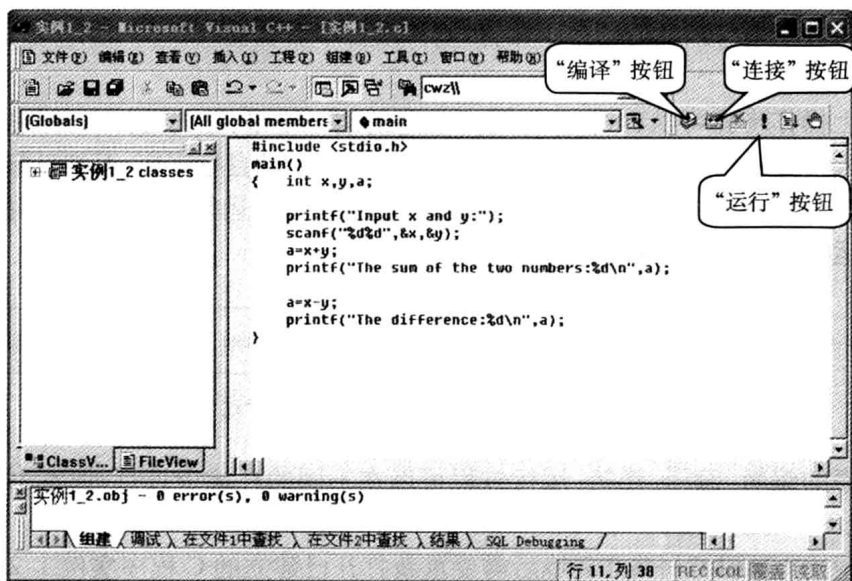


图 1.3 编译后的情况

如果在图 1.1 右侧“文件名”栏中输入文件名时省略扩展名,则系统建立的文件不是 C 语言程序,而是 C++ 语言程序(默认扩展名为 .cpp),所以编译时出现警告错误。这时如果在 main 前面加 void,即将 main 改为 void main,则可以消除警告错误。

单击“连接”按钮后,与编译时一样,如果在连接的过程中系统发现错误,将在下面窗口中列出所有错误和警告。修改错误后重新编译和连接,直到编译和连接都没有错误为止。

第 7 步:运行程序。单击“运行”按钮,出现如图 1.4 所示的界面,并要求用户输入 x 和 y 的值。输入 1200 和 180 后按 Enter 键,看到实例 1.2 的运行情况,如图 1.5 所示。运行程序时可以按 Shift 键切换中英文输入法。按任意键回到编辑窗口。

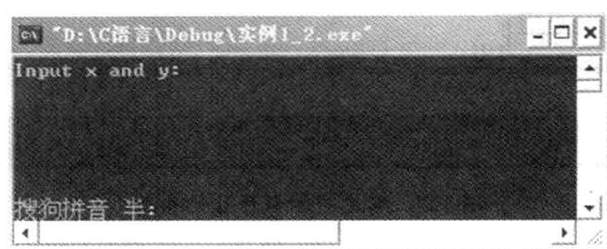


图 1.4 实例 1.2 的运行界面

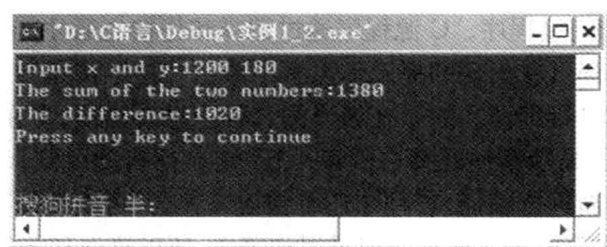


图 1.5 实例 1.2 的运行情况

如果运行结果与预期的结果不相符,则修改程序后重复第 6 步和第 7 步的操作,直到结果正确为止。

从以上操作步骤可以看到,要得到 C 程序的运行结果,首先选择一种集成环境,将源程序输入计算机内,然后把源程序翻译成机器能识别的目标程序,再把目标程序和系统提供的库函数等连接起来产生可执行文件,这时才可以运行程序。C 程序的编辑、编译、连接、运行过程如图 1.6 所示。



图 1.6 C 程序的编辑、编译、连接、运行过程

C 程序的编辑、编译、连接、运行过程可以在不同的环境中进行,而本书选用的是 Visual C++ 6.0 集成环境。

如果退出 Visual C++ 6.0 环境后需要重新打开已建立的 C 程序实例 1_2.c,则在资源管理器中双击“实例 1_2.c”,或先启动 Visual C++ 6.0 环境后通过“文件”|“打开”

菜单项打开文件。

1.2 合理选用数据类型

在编写 C 语言程序时,都需要处理大量的数据,其中最常用的数据类型为整型、实型、字符型,而常用的整型又有基本整型和长整型,常用的实型有单精度实型和双精度实型。不同类型的数据有不同的特性和处理方法,因此编写程序时合理选用相应的数据类型是至关重要的。

1.2.1 合理选用整型数据

在日常生活中我们经常需要处理整数,即不带小数点的数。例如,计算某人的年龄、统计学生人数等。在 C 语言中,常用基本整型 (int) 或长整型 (long) 变量存放整型数据。由于在 Visual C++ 6.0 环境中,基本整型和长整型所占字节数相同 (4 字节),因此本书只介绍基本整型。定义变量时必须根据需要给出其类型。

【实例 1.3】 编写程序,计算正方形铁板的面积,铁板边长为 150,如图 1.7 所示。



图 1.7 正方形铁板

1. 编程思路

利用公式“面积 = 边长 × 边长”计算正方形的面积,而且定义变量 `area` 存放正方形的面积。由于需要处理的数据是整数,所以选用 `int` 型。

2. 程序代码

```
#include <stdio.h>

main()
{   int area;                // 定义基本整型变量 area

    area=150;                // 把 150 赋给变量 area
    area=area*area;          // 计算正方形的面积
    printf("area=%d\n",area); // 输出正方形的面积
}
```

3. 运行结果

`area=22500`

4. 归纳分析

(1) 程序中使用了变量 `area`,所谓变量就是在程序执行过程中其值可以变化的量。C 语言将数据分为常量和变量,常量是在程序执行过程中其值永远不变的量。变量的实质是内存中的一个存储单元,因此在使用变量前应向系统申请存储单元,这一过程就是定义变量的过程。本程序用“`int area;`”定义了变量 `area`。