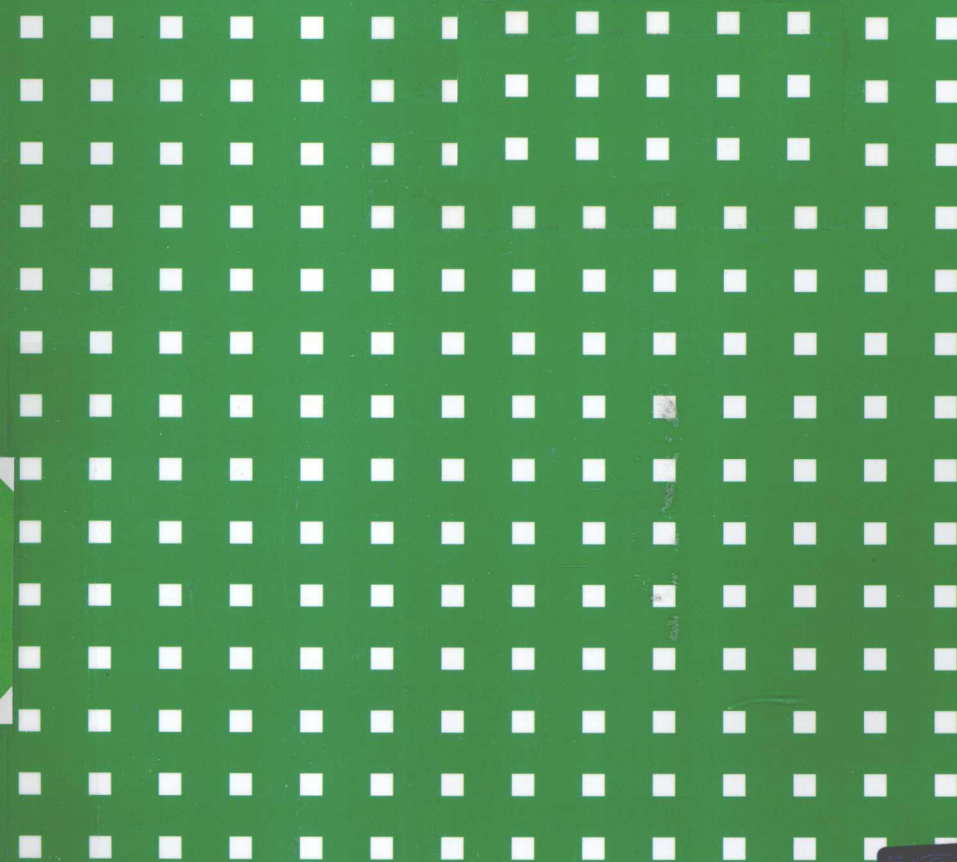




普通高等教育“十一五”国家级规划教材

C程序设计实训 (第2版)

林小茶 编著



清华大学出版社



内 容 简 介

本书是普通高等教育“十一五”国家级规划教材《C 程序设计教程》(第 2 版)的配套实训教材,也可以单独作为学习 C 语言的学习辅导书。

本书的内容安排与主教材完全一致。每一章都包括了 3 部分内容,第 1 部分对主教材每一章的内容进行了概括和总结,列举知识重点和编程要点,帮助学习者更好地掌握学习内容;第 2 部分针对主教材内容而精心设计实训内容,通过对实训内容和步骤的详细描述,使读者能够由浅入深地掌握调试程序的方法和技巧;第 3 部分则给出了主教材对应章节全部习题的参考答案,由于本书是为高校学生量身定做的,因此对大部分习题不仅仅给出一个答案,还尽可能地做一些说明,对于编程习题,在解答程序中尽量给出注释,这些习题是作者多年以来在教学中积累、收集并经过验证的习题,全部上机调试(Visual C++ 6.0 环境)通过。

本书适合作为高等院校计算机相关专业程序设计课程的实训教材,也可作为初学者的参考资料和各类计算机培训教材。

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 程序设计实训/林小茶编著. —2 版. —北京:清华大学出版社,2010.4

(高等学校计算机专业教材精选·算法与程序设计)

ISBN 978-7-302-21794-7

I. ①C… II. ①林… 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(2010)第 002324 号

责任编辑:张 民 李玮琪

责任校对:梁 毅

责任印制:杨 艳

出版发行:清华大学出版社

地 址:北京清华大学学研大厦 A 座

<http://www.tup.com.cn>

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

印 装 者:北京国马印刷厂

经 销:全国新华书店

开 本:185×260 印 张:13.25

字 数:325 千字

版 次:2010 年 4 月第 2 版

印 次:2010 年 4 月第 1 次印刷

印 数:1~3000

定 价:21.00 元

本书如存在文字不清、漏印、缺页、倒页、脱页等印装质量问题,请与清华大学出版社出版部联系调换。
联系电话:010-62770177 转 3103 产品编号:035889-01

出版说明

我国高等学校计算机教育近年来迅猛发展,应用所学计算机知识解决实际问题,已经成为当代大学生的必备能力。

时代的进步与社会的发展对高等学校计算机教育的质量提出了更高、更新的要求。现在,很多高等学校都在积极探索符合自身特点的教学模式,涌现出一大批非常优秀的精品课程。

为了适应社会的需求,满足计算机教育的发展需要,清华大学出版社在进行了大量调查研究的基础上,组织编写了《高等学校计算机专业教材精选》。本套教材从全国各高校的优秀计算机教材中精挑细选了一批很有代表性且特色鲜明的计算机精品教材,把作者们对各自所授计算机课程的独特理解和先进经验推荐给全国师生。

本系列教材特点如下。

(1) 编写目的明确。本套教材主要面向广大高校的计算机专业学生,使学生通过本套教材,学习计算机科学与技术方面的基本理论和基本知识,接受应用计算机解决实际问题的基本训练。

(2) 注重编写理念。本套教材作者群为各校相应课程的主讲,有一定经验积累,且编写思路清晰,有独特的教学思路和指导思想,其教学经验具有推广价值。本套教材中不乏各类精品课配套教材,并力图努力把不同学校的教学特点反映到每本教材中。

(3) 理论知识与实践相结合。本套教材贯彻从实践中来到实践中去的原则,书中的许多必须掌握的理论都将结合实例来讲,同时注重培养学生分析、解决问题的能力,满足社会用人要求。

(4) 易教易用,合理适当。本套教材编写时注意结合教学实际的课时数,把握教材的篇幅。同时,对一些知识点按教育部教学指导委员会的最新精神进行合理取舍与难易控制。

(5) 注重教材的立体化配套。大多数教材都将配套教师用课件、习题及其解答,学生上机实验指导、教学网站等辅助教学资源,方便教学。

随着本套教材陆续出版,相信能够得到广大读者的认可和支持,为我国计算机教材建设及计算机教学水平的提高,为计算机教育事业的发展做出应有的贡献。

清华大学出版社

第 2 版前言

本书是与清华大学出版社出版的普通高等教育“十一五”国家级规划教材《C 程序设计教程(第 2 版)》(林小茶、陈昕编著)配套的实训教材。

由于《C 程序设计教程(第 2 版)》采用了 Visual C++ 6.0 作为调试环境,因此,本书在第 1 版的基础上做了一些修订。

考虑到初学者对调试工具的掌握有一定的难度,因此将实训过程写得非常详细,对调试程序的步骤描述由难而易、由浅入深,并尽量使用一些有意义的截图来说明。如果学生能够耐心地按照书中描述的步骤去做,一定可以逐步掌握调试程序的方法,给指导上机的老师带来极大的方便。

书中所有的习题答案在 Visual C++ 6.0 调试环境下进行了重新调试。

第 1 版习题中的程序运行提示使用的都是英语提示,第 2 版中将大部分的提示改成了中文提示,避免学生们在阅读程序时由于英语的问题而影响学习兴趣(尽管笔者认为程序设计中适当地使用英语提示对学习是有好处的)。

最后,借此次本书再版的机会,向使用本书作为教材和学习参考书的教师和读者表示衷心的感谢,并殷切希望您对本书的内容和编写方法提出宝贵的意见和建议。

2010 年 3 月

目 录

第 1 章 C 语言概述	1
1.1 知识重点和编程要点	1
1.1.1 知识重点	1
1.1.2 编程要点	1
1.2 C 调试环境初识实训	1
1.2.1 实训目的和要求	1
1.2.2 实训内容和具体步骤	1
1.3 习题答案和分析	5
第 2 章 C 语言基础知识	8
2.1 知识重点和编程要点	8
2.1.1 知识重点	8
2.1.2 编程要点	8
2.2 C 语言基础知识实训	9
2.2.1 实训目的和要求	9
2.2.2 实训内容和具体步骤	9
2.3 习题答案和分析	13
第 3 章 结构化程序设计	26
3.1 知识重点和编程要点	26
3.1.1 知识重点	26
3.1.2 编程要点	26
3.2 结构化程序设计实训	27
3.2.1 实训目的和要求	27
3.2.2 实训内容和具体步骤	27
3.3 习题答案和分析	36
第 4 章 模块化程序设计	70
4.1 知识重点和编程要点	70
4.1.1 知识重点	70
4.1.2 编程要点	70
4.2 模块化程序设计实训	71
4.2.1 实训目的和要求	71
4.2.2 实训内容和具体步骤	71

4.3	习题答案和分析	78
第5章	数组和指针	100
5.1	知识重点和编程要点	100
5.1.1	知识重点	100
5.1.2	编程要点	101
5.2	数组和指针实训	101
5.2.1	实训目的和要求	101
5.2.2	实训内容和具体步骤	102
5.3	习题答案和分析	108
第6章	结构体等构造数据类型	145
6.1	知识重点和编程要点	145
6.1.1	知识重点	145
6.1.2	编程要点	145
6.2	结构体等构造数据类型实训	146
6.2.1	实训目的和要求	146
6.2.2	实训内容和具体步骤	146
6.3	习题答案和分析	151
第7章	文件	176
7.1	知识重点和编程要点	176
7.1.1	知识重点	176
7.1.2	编程要点	176
7.2	文件实训	176
7.2.1	实训目的和要求	176
7.2.2	实训内容和具体步骤	177
7.3	习题答案和分析	182

第 1 章 C 语言概述

1.1 知识重点和编程要点

1.1.1 知识重点

从编程的角度看,第 1 章需要掌握的知识重点包括:

(1) C 程序的构成。

① 程序的顶部一般要使用 `#include "stdio.h"` 语句。

② C 程序由函数构成,至少包含一个 `main` 函数,函数用一对大括号 `{}` 括起来。

③ 定义变量。变量在使用之前一定要先定义,定义变量的语句要尽量放在程序开始的位置。

④ 通过调用 `printf` 函数显示信息给用户。

(2) 要完成一个 C 程序的调试,必须经过编辑源程序、编译源程序、连接目标程序和运行可执行程序 4 个步骤。

1.1.2 编程要点

(1) C 语言严格区分大小写,因此关键字和系统函数名一定要用小写字母。

(2) 编译程序出现多个错误提示时,优先处理前面的错误。因为后面的错误很可能是由于前面的错误引起的,修改前面的一个错误经常可以消除后面的大量错误。

(3) 有时候,编译程序检查下一行时才能发现前面的程序出现了错误,因此,错误信息给出的提示信息中的行号不一定正确,需要程序员检查前面一行甚至几行程序。

1.2 C 调试环境初识实训

1.2.1 实训目的和要求

(1) 了解 Visual C++ 6.0 的调试环境,掌握进入 Visual C++ 6.0 调试环境的方法。

(2) 掌握调试 C 程序的 4 个步骤(编辑、编译、连接、运行)及在 Visual C++ 6.0 环境下的实现方法。

(3) 掌握在 C 程序中定义主函数的方法。

(4) 掌握在 C 程序中书写注释的方法。

(5) 了解函数 `printf` 的基本功能。

1.2.2 实训内容和具体步骤

1. 进入 Visual C++ 6.0 调试环境

如果用户使用的是 Windows 操作系统,进入 Visual C++ 6.0 调试环境是非常简单的,

选择“开始”→“程序”→Microsoft Visual Studio 6.0→Microsoft Visual C++ 6.0 命令,就可以进入 Visual C++ 6.0 调试环境了,如图 1-1 所示。

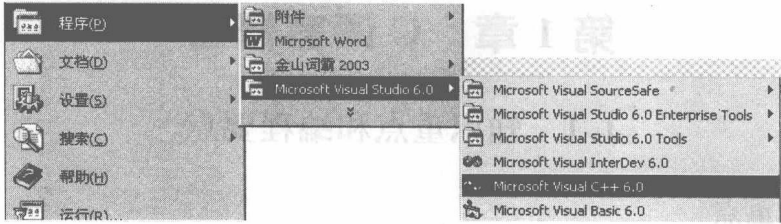


图 1-1 在 Windows 环境下进入 Visual C++ 6.0 调试环境

2. 调试第一个程序

请参照主教材的第 1 章中讲解的在 Visual C++ 6.0 环境下调试程序的过程,调试第一个程序。

- (1) 在 Visual C++ 6.0 环境下建立一个新的工作空间 MyWorkSpace。
- (2) 在 MyWorkSpace 工作空间中建立一个新的项目 MyProject。
- (3) 在 MyProject 项目中新建一个 Source Files 类型的文件 myprogram.cpp。
- (4) 在编辑窗口中编辑 myprogram.cpp 文件,内容如下:

```
#include "stdio.h"
void main()
{
    printf("欢迎你\n");
}
```

- (5) 按快捷键 Ctrl+F7 或选择 Build→Compile 命令,对程序进行编译。
- (6) 按快捷键 F7 或选择 Build→Build 命令,对程序进行编译和连接。
- (7) 按快捷键 Ctrl+F5 或直接单击运行按钮“!”执行程序。

3. 调试具有编译错误的程序

请按如下步骤操作。

- (1) 在编辑窗口中继续编辑 myprogram.cpp 文件,去掉“printf(“欢迎你\n”);”语句后面的分号,使程序变为:

```
#include "stdio.h"
void main()
{
    printf("欢迎你\n")
}
```

- (2) 按快捷键 Ctrl+F7 或选择 Build 菜单的 Compile 命令,对程序进行编译,此时,编译器提示的主要编译错误信息是:

```
Compiling...
myprogram.cpp
...
myprogram.cpp (6)
```

```
error C2143:syntax error:missing';'before'}}'  
Error executing cl.exe.  
myprogram.exe-1 error(s),0 warning(s)
```

表示正在对程序 myprogram. cpp 做编译,在生成目标程序 myprogram. obj 时编译程序检查出了问题,总共发现了 1 个错误。

错误信息还明确告诉程序员编译错误出现在第 6 行,语法错误是“printf 标识符之前缺少分号”,也就是说第 5 行后面缺少分号。

C 语言规定,每条语句都必须以分号结束,如果不写分号,编译器将会提示错误,这种错误就是所谓的编译错误,也可以说是语法错误。

因为有编译程序的帮助,从理论上说,语法错误是最容易纠正的错误。但由于 C 语言自身非常灵活,对于初学者来说,寻找到编译错误也不是一件容易的事情,需要在实践中学习提高。

(3) 将分号加回去,程序就可以正确编译、连接和运行了。

4. 调试具有连接错误的程序

按如下步骤操作。

(1) 在编辑窗口中继续编辑 myprogram. cpp 文件。

程序代码为:

```
#include "stdio.h"  
void print();  
void main()  
{  
    printf("欢迎你\n");  
    print();  
}
```

(2) 按快捷键 Ctrl+F7 或选择 Build→Compile 命令,对程序进行编译,编译成功。

(3) 按快捷键 F7 或选择 Build→Build 命令,对程序进行连接,连接失败。系统会提示下面的错误信息:

```
Linking...  
myprogram.obj:error LNK2001:unresolved external symbol"void __cdecl print(void)"(?  
print@@YAXXZ)  
Debug/MyProject.exe:fatal error LNK1120:1 unresolved externals  
Error executing link.exe.  
MyProject.exe-2 error(s),0 warning(s)
```

提示信息的第 1、2 行是“Linking...first. obj”,表示正在对“first. obj”做连接,由于 print 函数没有定义,对该函数的调用将使连接程序找不到这个函数的原型,因此报错。这种错误称之为连接错误。

连接错误比较容易纠正,需要掌握有关函数定义和说明的概念,并要保证相关系统文件在磁盘中存在。例如,stdio. h 文件应该存在,“#include "stdio. h"”语句才能将该文件包含到当前程序中。

(4) 在编辑窗口修改程序如下:

```
#include"stdio.h"
void print()
{ printf("-----\n");
}
void main()
{
    printf("Welcome\n");
    print();
    printf("Your Name:\n");
}
```

(5) 再次对程序进行编译和连接均成功,运行程序得到正确结果。

5. 调试具有运行错误的程序

按如下步骤操作。

(1) 在编辑窗口中重新编辑 myprogram. cpp 文件,使程序变为:

```
#include"stdio.h"
void main()
{   int a,b;
    printf("请输入两个整数:");
    scanf("%d%d",&a,&b);
    printf("%d/%d=%f\n",a,b,a * 1.0/b);
}
```

(2) 按快捷键 F7 或选择 Build→Build 命令,对程序进行编译和连接。

(3) 按快捷键 Ctrl+F5 或直接单击运行按钮“!”执行程序。输入两个整数:

```
6 2
6/2=3.000000
```

这是一个正确的结果,是计算 6 除以 2 的结果。

(4) 再次运行程序,输入 6 和 0 两个整数;系统会弹出如图 1-2 所示的对话框,提示程序在运行过程中发生了错误。

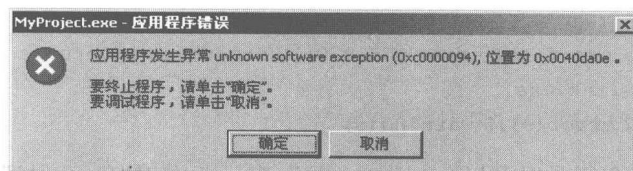


图 1-2 运行错误提示对话框

这是由于除数为 0 产生的运行错误。

由于系统对于运行错误有明确的提示,所以找到错误的原因也不难。

6. 调试具有逻辑错误的程序

按如下步骤操作。

• 4 •

(1) 在编辑窗口中重新编辑 myprogram.cpp 文件,使程序变为:

```
#include "stdio.h"
void main()
{
    printf("%d/%d=%f\n", 6, 4, 6/4);
}
```

(2) 按快捷键 F7 或选择 Build→Build 命令,对程序进行编译和连接,成功。

(3) 按快捷键 Ctrl+F5 或直接单击运行按钮“!”执行程序。程序运行结果是:

```
6/4=0.000000
```

显然,这不是正确的结果,其实正确的结果是 1.5。用 %f 显示浮点数。但程序的执行结果却是 0。

在程序经过正确的编译和连接以后,执行也有结果,但是却不是正确的结果,这就是所谓的“逻辑错误”。逻辑错误是最难查找的错误。这类错误往往来源于编程者对问题的错误理解和对程序设计工具的不了解。

本例程序中,表达式 6/4 中的运算符/是整除符号,其结果是整数 1,整数 1 却用 %f 控制显示,当然不能得到正确的结果。

正确的显示方法是:

```
printf("%d/%d=%f\n", 6, 4, 6.0/4);
```

显示结果是:

```
6/4=1.500000
```

当然,如果要求整除的结果,则可以使用语句:

```
printf("%d/%d=%d\n", 6, 4, 6/4);
```

显示结果是:

```
6/4=1
```

关于逻辑错误的查找方法将在后面的练习中讲解。

7. 对习题中所有要求编程的题目进行程序设计和程序调试

请读者自己编写程序,答案作为参考。

调试步骤略。

1.3 习题答案和分析

1. 回答下列问题

(1) 什么是计算机程序?

解: 计算机程序是由计算机指令构成的序列。

(2) 什么是程序设计语言?

解: 程序设计语言是一组用来定义计算机程序的语法规则。

(3) 程序设计语言如何分类?

解: 计算机程序设计语言分为两类, 一类是低级语言, 另一类是高级语言。

(4) 低级语言和高级语言的特点是什么?

解: 低级语言的主要特点是直接依赖计算机的硬件系统, 一种机型能识别的低级语言, 另一种机型可能完全不能识别; 而高级语言与低级语言不同, 它不再依赖计算机的硬件系统, 用高级语言编写的程序几乎可以不加修改地运行在不同机型的计算机上。

(5) 调试 C 程序需要经过几个步骤?

解: 要完成一个 C 程序的调试, 必须经过编辑源程序、编译源程序、连接目标程序和运行可执行程序 4 个步骤。

2. 单项选择题

(1) 对 printf 函数正确调用的语句是_____。

- A. printf("%d %d %d\n", x, y, z); B. printf("%d %d %d\n", x, y);
C. printf("%d %d \n", x, y, z); D. printf("%d %d %d\n");

解: 正确答案是 A。

B 选项中有 3 个 %d, 但是要显示的数据只有两个: x 和 y, C 语言规定, 转换说明符的个数应与数据参数的个数相等。

C 选项中有 2 个 %d, 但是要显示的数据却有三个: x、y 和 z, 也不符合 C 语言的规定。

D 选项中有 3 个 %d, 但是却没有需要显示的数据, 仍然不符合 C 语言的规定。

(2) 经过连接以后生成的可执行文件的扩展名是_____。

- A. com B. c
C. obj D. exe

解: 正确答案是 D。

C 的源程序是符合 C 语言语法的程序文本文件, 扩展名为 c; 对编辑好的文本文件进行成功编译后将生成目标程序, 目标程序文的扩展名是 obj。将目标程序和库函数连接为一个整体, 生成的可执行文件的扩展名是 exe。因此, 正确答案是 D。

3. 按照下列要求编写程序

(1) 下面的程序解决的问题是: 假设每个月有线电视的费用是 18 元, 请将 1 年需要交纳的费用显示在屏幕上。

```
/* ----- 求有线电视一年的费用 ----- */
# include "stdio.h"
void main()
{   int a,b,total;           /* 变量定义 */
    a=18;                   /* 每个月的费用 */
    b=12;                   /* 一年 12 个月 */
    total=a*b;              /* 计算 */
    printf("一年的费用是%d元.\n",total); /* 输出 */
}
```

如果不定义变量 a 和 b, 要求程序执行的结果一样, 如何修改程序? 请读者思考。

(2) 编写程序输出下列图案:

```
*****
*****
***
*
```

解:

```
#include "stdio.h"
void main()
{
    printf("\n");           /* 换行 */
    printf(" *****\n");
    printf("  ***** \n");
    printf("    ***  \n");
    printf("      *   \n");
}
```

(3) 编写程序输出下列字符:

```
-----
1. Open
-----
2. Close
-----
3. Exit
-----
```

解:

```
#include "stdio.h"
void main()
{
    printf("\n");           /* 换行 */
    printf("-----\n");
    printf("1. Open \n");
    printf("-----\n");
    printf("2. Close \n");
    printf("-----\n");
    printf("3. Exit \n");
    printf("-----\n");
}
```

第2章 C语言基础知识

2.1 知识重点和编程要点

2.1.1 知识重点

(1) C语言的变量和常量。

变量是用来存储一个可以变化的数值,在程序执行过程中值是可变的。常量的数值在程序执行过程中不会发生改变。变量需要定义,常量不需要定义。常量可以通过宏定义做替换。

(2) C语言基本数据类型。

C语言基本数据类型有整型、浮点型、字符型。整型数据又分普通整型、短整型和长整型;关键字分别使用 int、short 和 long;浮点型又分为单精度和双精度,关键字是 float 和 double;字符型的关键字是 char。

(3) C语言的运算符以及运算符的优先级和结合性。

C语言运算符分为以下几类:算术运算符、逻辑运算符、关系运算符、位逻辑运算符、移位运算符、赋值运算符、增/减1运算符、条件运算符以及一些特殊的运算符。书写复杂表达式时最好使用圆括号来明确地指定运算的先后顺序。逻辑运算符、关系运算符的练习放在第3章。

(4) C语言常用的输入输出函数。

printf 函数和 scanf 函数可以接收和显示各种数据类型的数据, getchar 函数和 putchar 函数分别负责接收和显示一个字符。

(5) 数据类型的转换。

不同类型的数据在一起运算时,需要转换为相同的数据类型。转换的方式有两种:自动转换和强制转换。

当赋值运算符左边的变量与赋值运算符右边的表达式的数据类型相同时,不需要进行数据类型的转换。当赋值运算符左边的变量与赋值运算符右边的表达式的数据类型不相同,系统负责将右边的数据类型转换成左边的数据类型。

2.1.2 编程要点

(1) 递增/减是有副作用的运算符,不要在一个表达式中多次使用一个带有递增/减运算符的变量。例如,表达式“j=i++++i;”就不是一个好的表达式。

(2) 注意不要在一个表达式中多次使用赋值和复合赋值运算符。

(3) scanf 函数中第二个参数是一个地址。

如果已经有定义“int i;”,那么“scanf("%d",i);”是错误的。但是,如果有定义“int i,*p=&i;”,则“scanf("%d",p);”是正确的。

(4) 用 scanf 函数和 printf 函数控制输入输出时,控制格式的使用应该与变量数据类型的一致。如果不一致,可能出现意想不到的后果。

(5) 程序在编译以后,系统在 Message 窗口列出编译错误信息和警告信息。尽管大多数警告信息都不影响程序的执行,但是,警告信息可以帮助程序员改正一些逻辑错误。

(6) 不要错误理解和使用指针运算符 * 和 &。

2.2 C 语言基础知识实训

2.2.1 实训目的和要求

- (1) 掌握 C 语言的变量的定义方法,这些变量的数据类型是基本数据类型。
- (2) 掌握整型数据、浮点数据和字符型数据的格式输入输出方法。
- (3) 掌握 putchar 函数和 getchar 函数的调用方法。
- (4) 掌握算术运算符、位逻辑运算符、增 1/减 1 运算符和 sizeof 运算符的用法。
- (5) 掌握调试程序的简单技巧。

2.2.2 实训内容和具体步骤

1. 练习用各种基本数据类型定义变量和输出方式

按如下步骤操作。

(1) 打开工作空间 MyWorkSpace 的 MyProject 项目。

(2) 打开 Windows 的记事本,将 myprogram.cpp 文件中的整个程序复制到记事本的编辑窗口中,另存为一个文件。这样,就可以将第一个练习中做的源程序保存起来。

(3) 将新的程序输入 MyProject 项目文件 myprogram.cpp 中(覆盖原来的内容),并对其进行调试。本书中的实验指导将主函数所在的源程序都存入文件 myprogram.cpp 中。

当然也可以建立一个新的工作空间、新的项目以及文件 myprogram.cpp。

(4) 在编辑窗口中编辑 myprogram.cpp 文件,代码如下:

```
#include"stdio.h"
void main()
{   int a=1,b=2,c=3;
    printf("%d+%d+%d=%d\n",a,b,c,a+b+c);
}
```

(5) 按快捷键 Ctrl+F7 或选择 Build→Build 命令,对程序进行编译和连接。

(6) 按快捷键 Ctrl+F5 或直接单击运行按钮“!”执行程序,得到正确的运行结果。

(7) 在编辑窗口中编辑 myprogram.cpp 文件,代码如下:

```
#include"stdio.h"
void main()
{   double a=1.0,b=2.0,c=3.0;
    printf("%d+%d+%d=%d\n",a,b,c,a+b+c);
}
```


(8) 重新编译、连接并执行程序,得到的运行结果是:

```
0+1072693238+0=1073741824
```

这是一个错误的结果。程序不能完成预定的功能,原因是显示格式的错误,初学者通常不能注意到这一点,数据类型变化以后,输出方式也必须随之变化,否则就会遇到像本例这样的问题。

(9) 将语句“`printf("%d+%d+%d=%d\n",a,b,c,a+b+c);`”改为“`printf("%lf+ %lf+ %lf= %lf\n",a,b,c,a+b+c);`”。

(10) 编译、连接并运行程序,得到正确结果。

2. 练习使用 `scanf` 函数接收数据

按如下步骤操作。

(1) 在编辑窗口中重新编辑 `myprogram.cpp` 文件,代码如下:

```
#include"stdio.h"
void main()
{   int a,b,c;
    printf("请输入 3 个整数:");
    scanf("%d%d%d",&a,&b,&c);
    printf("%d+%d+%d=%d\n",a,b,c,a+b+c);
}
```

(2) 编译、连接并执行程序。

程序运行时,在提示语句“请输入 3 个整数:”出现后,输入 1 2 3 ✓,得到正确结果。

(3) 再次执行程序。

程序运行时,在提示语句“请输入 3 个整数:”出现后,输入 123 ✓,但是系统却在等待操作者输入下一个数。因为 123 三个数中间没有用空白符分隔,系统将其认为是一个数。必须再输入两个数,34 5 ✓,程序才能正确接收数据,但这时系统计算的是 123+34+5 的值。

空白符是指空格、回车和 Tab 键,因此,要想让程序正确接收数据,正确的输入格式不止一个,还可以有许多组合。

例如:输入可以为:

```
1 ✓
2 ✓
3 ✓
```

(4) 再次执行程序,选择不同的输入格式进行测试。

3. 练习使用 `scanf` 函数接收带有特殊格式的数据

按如下步骤操作。

(1) 在编辑窗口中重新编辑 `myprogram.cpp` 文件,代码如下:

```
#include"stdio.h"
void main()
{   int a,b,c;
    printf("请输入时间(格式为 00:00:00):");
```