

高等院校信息技术课程精选规划教材

C语言程序设计实验指导

李莉 主编



南京大学出版社

高等院校信息技术课程精选规划教材

C语言程序设计实验指导

主 编 李 莉
副主编 冯小虎 李 慧
主 审 成 静 刘防修
 韩 飞



【微信扫码】
本书导学，领你入门



南京大学出版社

图书在版编目(CIP)数据

C语言程序设计实验指导 / 李莉主编. — 南京: 南京大学出版社, 2017.8

高等院校信息技术课程精选规划教材

ISBN 978-7-305-18995-1

I. ①C… II. ①李… III. ①C语言—程序设计—高等学校—教学参考资料 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 166122 号

出版发行 南京大学出版社

社 址 南京市汉口路 22 号 邮 编 210093

出 版 人 金鑫荣

丛 书 名 高等院校信息技术课程精选规划教材

书 名 C语言程序设计实验指导

主 编 李 莉

责任编辑 钟亭亭 苗庆松 编辑热线 025-83597482

照 排 南京南琳图文制作有限公司

印 刷 南京人文印务有限公司

开 本 787×1092 1/16 印张 14.25 字数 347 千

版 次 2017 年 8 月第 1 版 2017 年 8 月第 1 次印刷

ISBN 978-7-305-18995-1

定 价 32.80 元

网址: <http://www.njupco.com>

官方微博: <http://weibo.com/njupco>

官方微信号: njupress

销售咨询热线: (025) 83594756

* 版权所有,侵权必究

* 凡购买南大版图书,如有印装质量问题,请与所购
图书销售部门联系调换

前 言

C语言是目前为止应用广泛的高级程序语言之一,也是各类高等学校开设最为普遍的大学计算机程序设计课程。让学生具备运用计算机技术解决复杂问题的能力,是学生适应社会发展和进行科学研究的需要,也是工程教育专业认证的基本目标之一,而提高学生编程能力则是解决这些问题的重要措施。

本实验指导书是面向C程序设计使用的上机实践教材,旨在提高学生的编程能力,主要包括上机实验和模拟题两部分。

第一部分是基于教材内容的上机实验,共有13个实验。每个实验都包括实验要求、实验指导、实验内容(根据难度又分为夯实基础和应用提高两部分)以及思考与练习。该部分精选了与实验内容有关的代表性强、重点突出的各类上机题目,将对学生编程实践有很大的帮助。

第二部分给出了两套全国计算机等级考试二级C语言操作题。

本实验指导共包含15个实验,其中,实验1由韩飞编写,实验2、3、4由李莉编写,实验5、12、13由刘防修编写,实验6、7由成静编写,实验8、9由李慧编写,实验10、11由冯小虎编写,实验14、15由李莉、韩飞整理编写,全书由李莉统稿。

由于篇幅和课时的限制,以及作者水平有限,书中难免有不妥之处,敬请读者批评指正。

编 者

2017年3月

目 录



【微信扫码】
本书习题答案 & 资源

实验 1 VC 环境下 C 语言程序的调试 1

- 1.1 实验要求 1
- 1.2 实验指导 1
- 1.3 实验内容 7
- 1.4 总结 8

实验 2 顺序结构程序设计 9

- 2.1 实验要求 9
- 2.2 实验指导 9
- 2.3 实验内容 15
- 2.4 思考与练习 16

实验 3 选择结构程序设计 19

- 3.1 实验要求 19
- 3.2 实验指导 19
- 3.3 实验内容 25
- 3.4 思考与练习 28

实验 4 循环结构程序设计 31

- 4.1 实验要求 31
- 4.2 实验指导 31
- 4.3 实验内容 38
- 4.4 思考与练习 40

实验 5 程序设计常用算法 43

- 5.1 实验要求 43
- 5.2 实验指导 43

5.3 实验内容 50

5.4 思考与练习 52

实验 6 一维数组 54

6.1 实验要求 54

6.2 实验指导 54

6.3 实验内容 63

6.4 思考与练习 65

实验 7 二维数组 69

7.1 实验要求 69

7.2 实验指导 69

7.3 实验内容 78

7.4 思考与练习 83

实验 8 字符数组 86

8.1 实验要求 86

8.2 实验指导 86

8.3 实验内容 95

8.4 思考与练习 98

实验 9 函数 100

9.1 实验要求 100

9.2 实验指导 100

9.3 实验内容 112

9.4 思考与练习 114

实验 10 指针与数组 120

10.1 实验要求 120

10.2 实验指导 120

10.3 实验内容 131

10.4 思考与练习 133

实验 11 指针与字符串 138

11.1 实验要求 138

11.2 实验指导 138

11.3 实验内容 146

11.4 思考与练习 149

实验 12 结构体 154

12.1 实验要求 154

12.2 实验指导 154

12.3 实验内容 169

12.4 思考与练习 175

实验 13 文件 181

13.1 实验要求 181

13.2 实验指导 181

13.3 实验内容 190

13.4 思考与练习 191

实验 14 综合程序设计一 197

14.1 实验要求 197

14.2 实验指导 197

14.3 实验内容 199

实验 15 综合程序设计二 208

15.1 实验要求 208

15.2 实验内容 208

实验 1 VC 环境下 C 语言程序的调试

1.1 实验要求

1. 熟悉掌握集成开发环境 Visual C++。
2. 掌握 C 语言程序调试的基本步骤(即编辑、编译、连接和运行)。
3. 掌握 C 语言程序设计的基本框架,能够编写简单的 C 语言程序。
4. 掌握 C 源程序的编辑、保存和调用。
5. 编写程序的文件名均采用 ex1_题号.c 的形式命名,如【1.1】程序文件名为 ex1_1.c。

1.2 实验指导

集成开发环境(Integrated Development Environment, IDE)就是将编程过程中需要使用的所有支持软件都集成在一起,统一进行管理和使用。具体地说集成开发环境包括源程序的录入、编辑、编译、连接、程序运行时的调试和跟踪等,以及为程序开发提供的各种工具,并具有窗口管理和联机帮助等功能。

常用的 C 语言集成开发环境有 Visual C++ 6.0、DEV-CPP、Turbo C2.0/3.0 等。

Visual C++ 系列产品是微软公司推出的一款优秀的 C++ 集成开发环境,本实验指导是以 Visual C++ 为编程软件进行阐述。

1. 启动 Visual C++ 6.0 集成开发环境

单击“开始”→“程序”→“Microsoft Visual studio 6.0”→“Microsoft Visual C++ 6.0”命令,启动 Visual C++,启动后主窗口界面如图 1.1 所示,窗口界面各区域功能如图 1.2 所示。

2. 新建一个文件

单击菜单栏的【File】,选择【New...】,如图 1.3 所示,弹出“新建”对话框,如图 1.4 所示。

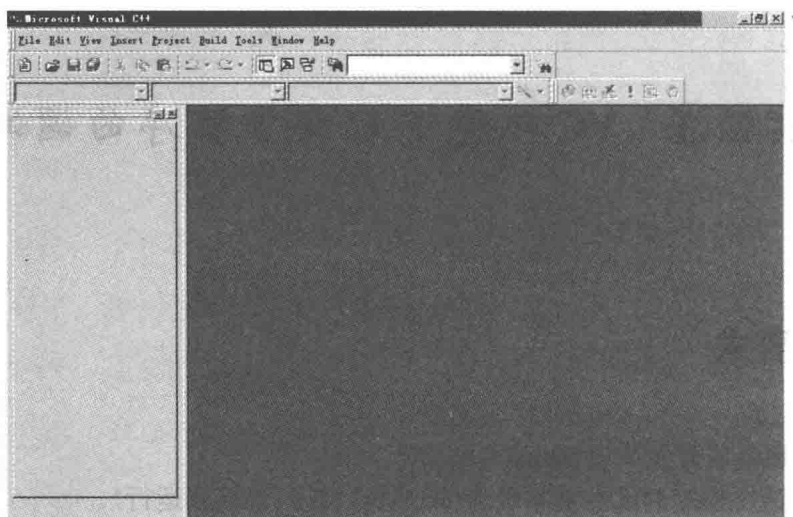


图 1.1 Visual C++ 6.0 集成开发环境



图 1.2 窗口界面区域功能

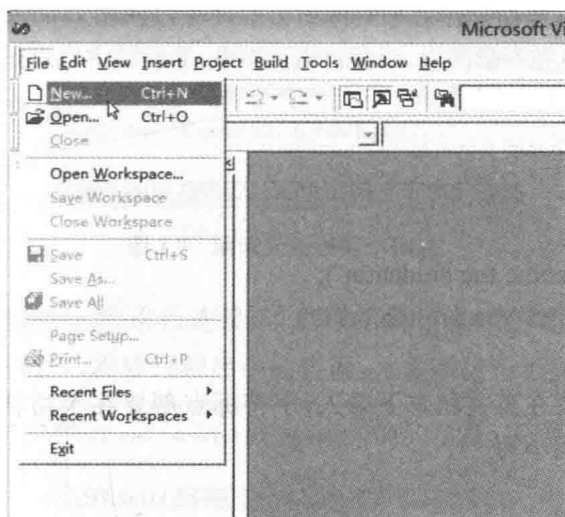


图 1.3 【File】菜单

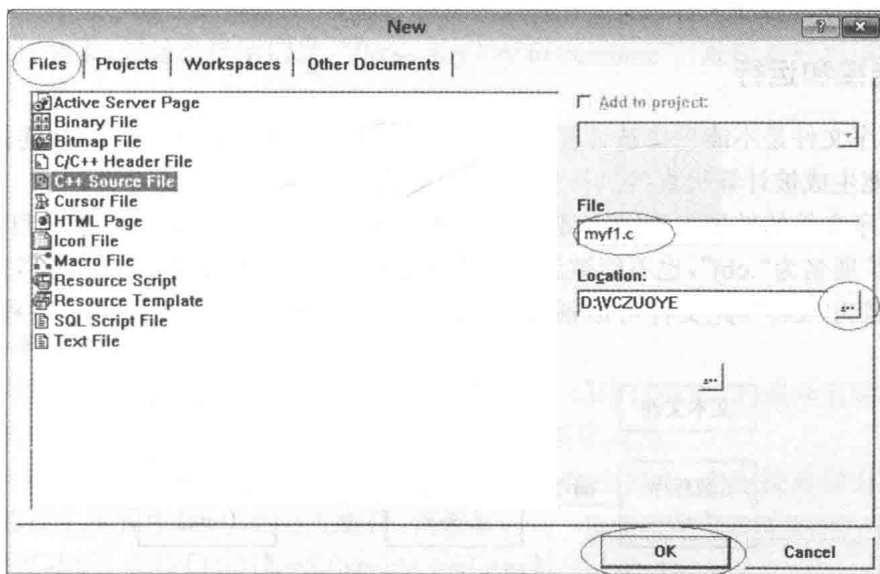


图 1.4 新建文件向导

按下列操作步骤逐一操作：

- (1) 步骤 1: 选择对话框【Files】页面；
- (2) 步骤 2: 选择【C++ Source File】选项；
- (3) 步骤 3: 单击对话框右边【Location:】的  按钮，打开选择位置对话框，选择 C 源程序要保存的位置；
- (4) 步骤 4: 在对话框右边【File】下方输入源程序的文件名。如输入 myf1.c，也可省略扩展名，在 VC 环境下若省略了文件的扩展名，则以“.cpp”为默认扩展名，源程序保存的文件名为 myf1.cpp；
- (5) 步骤 5: 单击【OK】按钮。

3. 编辑源程序

在程序编辑区录入下列程序。

```
#include <stdio.h>

void main()
{
    printf("Hello, Welcome the student\n");
    printf("Study hard!\n You are the best!");
}
```

注意:源程序一定要在英文状态下录入,字符标点都要在半角状态下录入,同时注意英文字母大小写,一般采用小写。

4. 保存源程序

程序录入完成后,单击工具栏的保存按钮,或选择【File】菜单中的【Save】,或按快捷键 Ctrl+S,就可保存源程序文件。

5. 编译、连接和运行

C 源程序文件是不能直接被计算机执行的,它需要通过编译(Compile)和连接(Link)两个步骤,才能生成被计算机直接执行的“可执行文件”。

C 源程序文件的扩展名为“.c”,不能直接被计算机执行;经过编译后,生成目标文件(二进制文件)扩展名为“.obj”,也不能被计算机直接执行;再经过连接,生成可执行文件(二进制文件)扩展名为“.exe”,此文件可以被计算机直接执行。C 源程序文件编译过程,如图 1.5 所示。

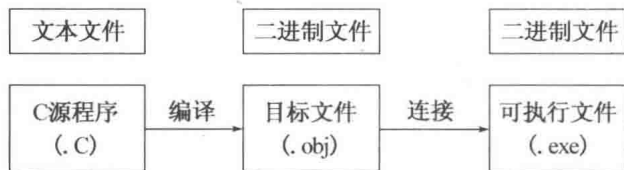


图 1.5 C 源程序的编译、连接过程

在 VC 环境下可通过菜单栏或快捷工具栏两种操作方法,实现对 C 源程序进行调试。

(1) 【快捷工具栏】操作步骤。

采用快捷工具栏上的按钮,如图 1.6 所示,可快速实现编译、连接和运行。



图 1.6 编译、连接和运行按钮工具栏

先选择工具栏中的编译按钮进行编译,若编译结果信息区显示编译无错误,即“error(s)”与“warning(s)”前的数是 0 时,则说明编译成功。其中,“error(s)”代表语法错误;“warning(s)”代表警告

错误,如图 1.7 所示。若“error(s)”前的数字不是 0,说明程序有语法错误,此时编译失败,则无法进入下一步的连接,需要修改程序中的错误,并重新进行编译,直到编译成功。

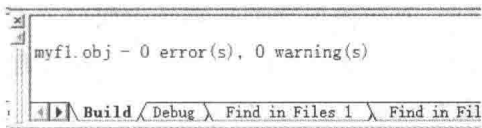


图 1.7 编译无错误提示信息

选择工具栏中的连接按钮运行连接,若编译结果信息区连接无错误,如图 1.8 所示,说明连接成功;若连接有错误,需修改程序中的错误,并重新进行编译连接,直到连接成功。

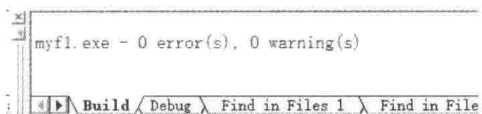


图 1.8 连接无错误提示信息

选择工具栏中的运行按钮。当编译和连接都无误时,点击运行按钮,则在另一窗口显示程序运行结果,并显示提示信息:“Press any key to continue”。此时按任意键返回 VC 主窗口,如图 1.9 所示。

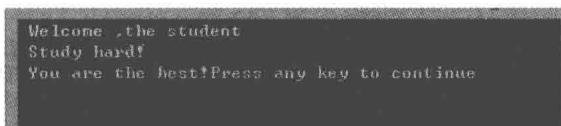


图 1.9 程序运行结果窗口

(2) 【菜单栏】操作步骤。

选择【菜单栏】→【Build】菜单,先选择【Compile myfl.c】进行编译,若编译有错误,修改源程序,再次选择【Build】→【Compile myf】重新编译,直到编译成功。

当编译成功后,选择【Build】→【Build myf.exe】,进行连接操作,当连接有误时,需重新修改源程序,修改完源程序后,仍然是先编译,再连接。

当连接成功后,选择【Build】→【Execute myf.exe】,在另一打开的窗口中查看程序运行结果。

6. 关闭工作空间

在完成一个程序的编写、编译、连接和运行后,如果要编写另一个 C 程序,则必须关闭当前程序的编辑区及其工作空间。

关闭工作空间可采用两种操作方法:

① 选择菜单栏【File】菜单中的【Close Workspace】,如图 1.10 所示。

② 最简单有效的方法就是直接关闭整个 VC。

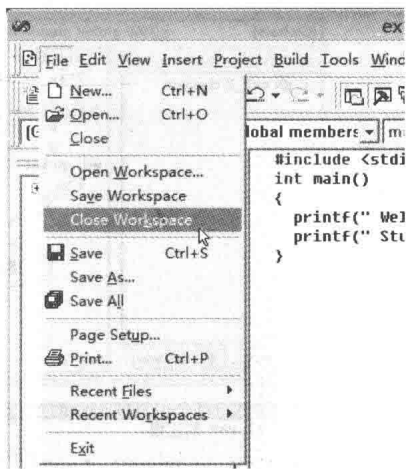


图 1.10 【File】菜单的子菜单
【Close Workspace】

编写一个新的 C 源程序前,必须确保当前程序的编辑区及其工作空间都是关闭的。如图 1.1 所示的 Microsoft Visual C++ 启动后主窗口界面。若不能确定,可以直接关闭整个 VC,然后重新启动 Microsoft Visual C++。

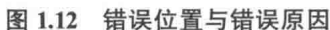
```
# include <stdio.h>
```

```
{
    float x, y;
    x=25;
    y=x*x
    printf("06.2f square: 06.2f\n", x,y);
}
```

8. 如何查看错误行

```
myf2.obj - 1 error(s), 0 warning(s)
```

若要想找到哪条语句出错,以及错误的原因,可拖动信息框右边的垂直滚动条,将鼠标移至错误信息提示行双击,则在程序编辑区便出现一个蓝色箭头,此位置便是程序错误的所在行,根据错误提示的信息,修改程序。如图 1.12 所示。



错误提示信息为:

syntax error: "missing ';' before identifier 'printf'"

其含义是在标识符 printf 之前丢了分号‘;’。修改错误继续调试(编译、连接)程序。

1.3 实验内容

【1.1】 按实验指导步骤录入并运行下列程序,观察其运行结果。

```
#include <stdio.h>

void main()
{
    printf("This is my first program!");
    printf("\n");
    printf("Study hard!");
}
```

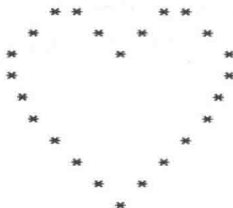
【1.2】 按实验指导步骤录入并运行下列程序,观察其运行结果。

```
#define PI 3.14159
#include <stdio.h>
void main()
{
    double area,r;
    r=4.0;
    area=2*PI*r*r;
    printf("\n the area of yuan is %lf\n",area);
}
```

【1.3】 按实验指导步骤录入并运行下列程序,观察其运行结果。

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

【1.4】 编写 C 语言程序,使其程序运行后输出下列图案。



1.4 总结

编程时可能会出现各种错误,程序设计中出现的错误大致可分成四类:

(1) 编译错误。源程序编译时主要是对源程序进行词法分析和语法分析,通常称为检查语法错误。例如,不符合规定的语句格式、数据类型说明与使用不一致、不正确的分隔符或符号以及不完整语句结构等。

(2) 连接错误。连接是把要执行的程序与库文件或其他已经翻译好的子程序连接在一起,形成机器能执行的程序。例如,函数名书写错误、缺少包含文件或包含文件的路径错误等。

(3) 运行错误。所谓运行期,即程序在编译连接后产生可执行文件后,执行该文件。所以,运行期错误是指可执行程序执行过程中发现的错误。如在计算过程中遇到除数为 0 的错误、求一个负数的平方根等等。编译系统发现这类错误后,如无特殊指示,通常告知一些适当信息,然后会立即停止程序的执行。为阻止这类错误的发生,程序设计者可在程序中编入一些由设计者自己来检查这类错误的程序段,这可能更适合于设计者自己的处理要求。

(4) 逻辑错误。程序运行后,其结果与问题的结论有偏差,即没有得到预期的结果,如表达式出错、运算符出错或算法出错等,编译器在编译时,这类错误无法检测,也不会对这类错误提示错误信息,因此逻辑错误比较难排查。

(5) 警告错误。警告错误就是在经过编译器检查后没有出现语法错误,但程序中存在些代码编写不是非常恰当的问题,这些代码不会影响程序编译,而在少数情况下是会影响程序运行。如源程序中发现一个已定义但从未使用过的变量、函数参数的数据类型说明不一致。这类错误从语法上讲是正确的,因此系统一般不会停止编译。在大多数情况下不会阻止目标程序与可执行程序的生成、连接和运行,但是对这类错误不应掉以轻心,应仔细检查程序,这往往存在着某种潜在的运行错误。

C 语言程序设计的学习是循序渐进的过程,熟悉和看懂程序设计中错误提示信息是非常必要的。因此在学习程序设计时,要不断地掌握错误提示信息的含义。

实验 2 顺序结构程序设计

2.1 实验要求

1. 掌握程序设计的基本思想。
2. 熟悉、掌握 C 语言基本数据类型(int, long, float, double, char)的说明。
3. 掌握基本运算符及表达式,尤其是算术表达式在实际问题中应用。
4. 掌握符号常量的正确使用。
5. 掌握格式输入函数 scanf 并能灵活应用。
6. 掌握格式输出函数 printf 并能灵活应用。
7. 了解常用的数学函数以及数学函数的头文件。
8. 必须掌握的算法:① 两数的交换;② 拆数法;③ 大小写字母的转换。
9. 编写程序的文件名均采用 ex2_题号.c 的形式命名,如【2.1】程序文件名为 ex2_1.c。

2.2 实验指导

1. 程序设计的基本思想

C 语言采用结构化程序设计的思想。结构化程序设计包括三大结构:顺序结构、选择结构和循环结构。

程序设计的基本思想是:

- (1) 分析问题:理解题意,提出问题。
- (2) 建立模型:寻找解决问题的有效方法或直接公式。
- (3) 算法设计:算法的好坏直接影响着程序的可读性、通用性、有效性。
- (4) 编写程序:用 C 语言代码编写程序。
- (5) 调试程序:若结果正确,结束;否则检查程序,修改程序,重新调试程序。依次类推,直到调试结果正确为止。

程序设计的核心是算法,学习程序设计的过程也是算法积累的过程。

算法是在有限步骤内求解某一问题所使用的基本运算,及规定的运算顺序所构成的完整解题步骤,又称为计算机解题的过程。

2. C 语言程序的结构

C 语言程序是由函数组成,函数是 C 语言程序设计的基础。C 语言程序是由一个且仅有一个主(main)函数和若干个子函数组成,子函数可有可无。

从一个简单的程序设计开始学习,学习主函数(main)的程序设计。

主函数(main)的程序设计的基本框架:

```
main()
{
    所需数据的数据类型定义说明;
    数据输入;
    算法或公式;
    数据输出;
}
```

数据类型描述了某种数据的特性,其表现形式是占据存储空间的多少以及构造特点等,不同的数据类型具有不同的取值范围和存储格式。

在程序设计中,先要对所需数据进行数据类型的定义说明,并提供算法或公式中部分变量的初始值,程序设计的最终目的是输出结果。

3. 数据的输入 / 输出

C 语言没有输入/输出语句,而是通过调用标准库函数提供的格式输入函数 scanf 和格式输出函数 printf 来实现数据的输入和输出。它们的函数原型包含在头文件 stdio.h 中。因此在源程序中使用这两个函数时通常在程序的头部加一条命令行,# include “stdio.h”或 #include <stdio.h> 。

格式输入函数 scanf 的一般格式:

```
scanf("格式控制",地址表列)
```

格式输出函数 printf 的一般格式:

```
printf("格式控制",输出表列)
```

格式输入函数 scanf 和格式输出函数 printf 的第一个参数必须用双引号括起来,包括格式控制字符和普通字符,格式字符都是以“%”开头。无论是输入还是输出,不同的数据类型对应不同的格式说明字符,使用时必须统一,不能混用,也就是说,整型数据类型不能用实型格式字符,实型数据类型不能用整型格式字符等。

格式输出函数 printf 常用的格式字符说明见表 2-1 所示。

表 2-1 printf 常用的输出格式字符及用法

格式符	含 义	用 法
d	以十进制形式输出带符号整数(正数不输出符号)	%d: 以整型数据的实际长度输出 %md: m 为输出数据指定所占有的宽度,若输出数据的实际长度小于 m,则在左端补空格,否则按数据的实际长度输出