



普通高等教育“十二五”规划教材

(C语言)

CHENGXU SHEJI JICHU
SHILIHUA FANZHUAN JIAOCHENG

程序设计基础

实例化翻转教程

主 编 刘华富 魏 歌



北京邮电大学出版社
www.buptpress.com



普通高等教育“十二五”规划教材

程序设计基础 实例化翻转教程 (C 语言)

主 编 刘华富 魏 歌

北京邮电大学出版社
• 北京 •

内 容 简 介

本书按照由“范例导学”(自学)到“知识详解”(解惑)再到“应用实践”(应用)的结构,较系统地介绍了 C 语言的语法、语义与语用及结构化程序设计的概念、方法与基本技术。本书的主要内容有 C 语言概述、数据类型、运算符、表达式、输入和输出、程序的控制结构、数组、指针、结构体、联合体、枚举、链表、函数、文件等内容。本书适合作为本科高等学校计算机专业 C 语言程序设计课程教学教材,或职业技术学院计算机专业教材,也可作为自学 C 语言或培训程序设计的教学机构使用,还可作为工程设计人员的参考用书。

图书在版编目(CIP)数据

程序设计基础实例化翻转教程 / 刘华富, 魏歌主编. — 北京: 北京邮电大学出版社, 2016. 8

ISBN 978-7-5635-4591-9

I. ①程… II. ①刘… ②魏… III. ①程序设计—教材 IV. ①TP311.1

中国版本图书馆 CIP 数据核字(2015)第 301169 号

书 名	程序设计基础实例化翻转教程
主 编	刘华富 魏 歌
责任编辑	向 蕾
出版发行	北京邮电大学出版社
社 址	北京市海淀区西土城路 10 号(100876)
电话传真	010-82333010 62282185(发行部) 010-82333009 62283578(传真)
网 址	www.buptpress3.com
电子信箱	ctrd@buptpress.com
经 销	各地新华书店
印 刷	北京九州迅驰传媒文化有限公司
开 本	787 mm×1 092 mm 1/16
印 张	18.5
字 数	460 千字
版 次	2016 年 8 月第 1 版 2016 年 8 月第 1 次印刷

ISBN 978-7-5635-4591-9

定价: 39.00 元

如有质量问题请与发行部联系

版权所有 侵权必究

前 言

在培养应用型人才教学改革和教学实践中,如何调动学生自主学习的积极性、自觉参与到教学过程中来,是我们始终关注的焦点之一。

C语言是大部分本科高等学校计算机专业教学使用最广泛的程序设计语言,也是大部分学生从中学到大学后学的第一门计算机专业课程;因此,如何解决学生对于学习计算机专业课程的畏难情绪,使学生能积极参与到教学过程中来,并能较早地开始动手获得学习过程中的满足感,是我们编写本书的出发点。

翻转教学是以学生自学为主,由自学、质疑到解惑的教学过程。本书基于对这一思想的理解,尝试性地将简单程序纳入到学生自学范围,从而形成由“范例导学”(自学)到“知识详解”(解惑)再到“应用实践”(应用)的格局。希望以这种形式,部分解决学生自学参与度不足的问题。

本书参考了国内外著名的C语言教材,根据我们在教学改革和教学实践中的体会,按照应用型人才培养目标,力求体现如下特色:

1. 难度上递进。第一部分最为容易,适用于自学;第二部分详细讲述,适用于教学;第三部分只作介绍,适用于课外兴趣。

2. 以“读”促练。通过大量的基础程序阅读,提高学生的程序思维能力。每个程序都有其运行结果,方便学生在没有机器的条件下自学。

3. 一点一例。每个知识点都有一个简单范例展示其功能,让学生能直观地了解知识点的应用。

本书是在我校与安博教育集团合作编写的教材《C语言项目化实践教程》和《C语言程序设计教程》的基础上重新构思编写的,由刘华富和魏歌共同构思策划,由魏歌执笔,由卓琳、孟志刚等多位教学经验丰富并且在教学一线任教的教师和有多年实际项目开发经验的大型软件企业工程师共同参与讨论,真正做到了教学实践和工程实践的深切融合,适合作为高等学校C语言程序设计课程的教材,也可以作为工程设计人员的参考用书。

长沙学院ACM(美国计算机协会)程序设计竞赛队员全贺菁、石文、欧建桓、李亦轩、余星、陈琳、王重和、丁金如等同学参与了范例代码的编写、录入和全书校对等工作,在此对这些同学表示由衷感谢。

最后向所有关心、支持本书出版的领导和同事表示由衷的感谢,向使用我们教材的读者表示诚挚的谢意,教材中可能存在的缺点和错误,也请各位读者不吝指教。

长沙学院 刘华富

2016.8

目 录

第 1 章 程序入门与开发环境说明	1
第 1 节 程序与语言	1
第 2 节 准备开发环境	2
第 3 节 程序的开发过程	4
第 4 节 程序设计风格	6
第 5 节 应用实践	6
第 2 章 数据类型与基本运算	8
第 1 节 范例导学	8
第 2 节 知识详解	23
2.1 基本数据类型	23
2.1.1 整型	24
2.1.2 浮点型	25
2.1.3 字符型	26
2.2 C 语言的运算符	28
2.3 C 语言的表达式	29
2.4 C 语言各类运算符和表达式	29
2.4.1 算术运算符和算术表达式	29
2.4.2 关系运算符和关系表达式	30
2.4.3 逻辑运算符和逻辑表达式	31
2.4.4 位运算符和位运算表达式	31
2.4.5 赋值运算符和赋值表达式	33
2.4.6 条件运算符和条件表达式	33
2.4.7 逗号运算符和逗号表达式	33
2.4.8 指针运算符	34
2.4.9 求字节数运算符	34
2.4.10 强制类型转换运算符	35
2.4.11 分量运算符	37
2.4.12 下标运算符	37



第3节 应用实践	37
第3章 输入与输出	38
第1节 范例导学	38
第2节 知识详解	53
3.1 字符输入、输出函数	54
3.1.1 字符输入函数 getchar	54
3.1.2 字符输出函数 putchar	54
3.2 格式化输入、输出函数	55
3.2.1 格式化输出函数 printf	55
3.2.2 格式化输入函数 scanf	59
第3节 应用实践	62
第4章 循环与分支	64
第1节 范例导学	64
第2节 知识详解	77
4.1 if 语句的用法	80
4.2 if...else 的用法	80
4.3 if...else 的嵌套	82
4.4 else if 的用法	82
4.5 悬垂 else	83
4.6 switch 语句	84
4.7 for 循环	87
4.7.1 循环次数	88
4.7.2 取值范围	88
4.7.3 条件省略	89
4.8 while 循环	89
4.9 do...while 循环	90
4.10 break 语句	92
4.11 continue 语句	93
第3节 应用实践	93
第5章 数组	97
第1节 范例导学	97
第2节 知识详解	107
5.1 数组的基本概念	107
5.2 一维数组	108



5.2.1 一维数组的声明及使用	108
5.2.2 一维数组元素的初始化	109
5.2.3 一维数组元素的访问	109
5.2.4 一维数组应用实例	110
5.3 多维数组	112
5.3.1 二维数组的声明与使用	112
5.3.2 多维数组应用举例	113
5.4 字符串与字符数组	116
5.4.1 一维字符数组	116
5.4.2 二维字符数组	117
5.4.3 字符串的输入和输出	117
5.4.4 常用的字符串处理函数	119
5.4.5 字符串应用举例	120
第3节 应用实践	122
第6章 指针	125
第1节 范例导学	125
第2节 知识详解	138
6.1 指针的定义与运算	138
6.1.1 指针变量的定义	139
6.1.2 指针的运算	141
6.2 const 限定指针	147
6.3 动态内存分配	147
6.3.1 什么是动态内存分配	147
6.3.2 使用函数动态分配内存	148
6.3.3 使用 C++ 运算符动态分配内存	149
6.4 指针与数组	150
6.4.1 指针与一维数组	150
6.4.2 指针与二维数组	152
6.4.3 多级指针	154
6.4.4 指针数组	156
6.5 字符指针	157
6.6 指针与函数	159
6.6.1 指针作函数参数	159
6.6.2 指针函数	160
6.6.3 函数指针	162
6.6.4 命令行参数	165



6.7 指针与结构体	166
第3节 应用实践	170
第7章 结构体、联合体与枚举	171
第1节 范例导学	171
第2节 知识详解	180
7.1 结构体	180
7.1.1 结构体类型的定义	180
7.1.2 结构体成员的访问	183
7.1.3 结构体变量的初始化	184
7.2 结构体与函数	184
7.3 结构体与数组	185
7.3.1 结构体数组的定义	185
7.3.2 结构体数组的初始化	186
7.3.3 结构体数组的引用	187
7.3.4 结构体数组应用举例	187
7.4 结构体与指针	188
7.4.1 结构体指针变量的定义	188
7.4.2 结构体指针变量的引用	189
7.4.3 指向结构体数组的指针	190
7.4.4 结构体指针变量作函数参数	191
7.5 联合体	193
7.5.1 联合体类型的定义	193
7.5.2 联合体变量的定义	193
7.5.3 联合体类型成员的访问	194
7.5.4 联合体应用举例	194
7.6 枚举	196
7.6.1 枚举类型的定义	196
7.6.2 枚举类型变量的定义	197
7.6.3 枚举类型变量的赋值和使用	197
第3节 应用实践	198
第8章 链表	199
8.1 链表结构	199
8.2 单向链表	200
8.3 双向链表	209



第 9 章 函数	214
第 1 节 范例导学	214
第 2 节 知识详解	223
9.1 自定义函数	223
9.2 C 语言的标准库函数	225
9.3 函数的参数传递	226
9.3.1 传值调用	226
9.3.2 数组作函数参数	228
9.3.3 指针作函数参数	229
9.4 函数递归	231
9.5 变量的作用域	233
9.5.1 局部变量	233
9.5.2 全局变量	234
9.6 变量的存储类别	235
9.6.1 自动变量	235
9.6.2 全局变量	235
9.6.3 外部变量	236
9.6.4 寄存器变量	237
9.7 编译预处理命令	237
9.7.1 文件包含指令	237
9.7.2 宏定义指令	239
9.7.3 条件编译预处理指令	241
第 3 节 应用实践	242
第 10 章 文件	245
附录 I 结构化程序设计与面向对象程序设计简介	255
一、结构化程序设计	255
二、面向对象程序设计	256
三、结构化程序设计与面向对象程序设计的关联	257
附录 II 程序设计基础实验指导	270
实训 1 程序设计入门	270
实训 2 数据类型与输入、输出	271
实训 3 运算符与表达式	272
实训 4 选择结构	273



实训 5 循环结构(1)	274
实训 6 循环结构(2)	274
实训 7 函数	276
实训 8 数组	277
实训 9 数组、函数综合训练	278
实训 10 结构体	279
实训 11 指针	280
实训 12 文件	281
附录Ⅲ ASCII 码表	282
参考文献	286

第 1 章 程序入门与开发环境说明

第 1 节 程序与语言

语言是交流工具,是人类描述思维的基础,是一套符号和语法规则的集合。程序设计语言(programming language)作为一种语法规则相对简单的人工语言,它提供了人类与计算机交流的途径。使用程序设计语言的人被称为程序员(programmer),他们通过程序设计语言向计算机传达人类思想和指令;而程序(program)则是这些指令的载体,它指导计算机完成特定的工作。

本书将带领读者走进程序的世界,教读者一种将思想传达给计算机的方法,让计算机帮助人们完成复杂的工作。

程序设计语言的发展有一个从低级到高级的过程。从低级的机器语言(machine language)和汇编语言(assembly language)发展到各种形式的高级语言(high-level language)历经数十年。早期的语言如机器语言和汇编语言更接近机器的逻辑,执行效率高但却不易被人们掌握和使用。高级语言如 BASIC 语言、PASCAL 语言、Java 语言和 C 语言(C language)相对而言易学、易用,能更好地描述复杂的逻辑。其中的 C 语言更是一种使用非常广泛的计算机高级程序设计语言。

1972 年,美国贝尔实验室的丹尼斯·里奇(Dennis Ritchie)和肯·汤普森(Ken Thompson)设计并实现了 C 语言。1989 年,美国国家标准局(American National Standard Institution)为 C 语言制定了一套完整的国际标准语法,称为 ANSI C,是现行的 C 语言标准。1983 年,贝尔实验室的本贾尼·斯特劳斯特卢普(Bjarne Stroustrup)博士发明并实现了 C++ 语言(C plus plus language)。C++ 语言在 C 语言的基础上引入了面向对象程序设计的思想,是 C 语言的超集。

虽然 C++ 语言有 ANSI 标准和 ISO(国际标准化组织)标准,但各软件公司开发的 C++ 语言版本并不都严格遵守它们,而是与标准保持兼容且稍有修改和扩充。GNU GCC(GNU 编译器套件)是目前最符合和接近 C++ 标准的编译器之一,也是国际大学生程序设计竞赛(ACM-ICPC)上使用的标准编译器。本书采用 bloodshed Dev-C++ 作为标准开发环境。Dev-C++ 是一个开源(open source)的开发环境,拥有适合初学者使用的各种语言版本,当然也包括中文版。



第 2 节 准备开发环境

Dev-C++ 的安装比较简单, 下载安装包后直接双击, 基本上一路单击“next”按钮(“下一步”按钮)即可完成。如需使用中文版, 只需在 Dev-C++ first time configuration (Dev-C++ 第一次配置) 中, 将语言设置为 Chinese(中文)即可。安装过程如图 1-1~图 1-8 所示。



图 1-1 选择安装语言(英语)

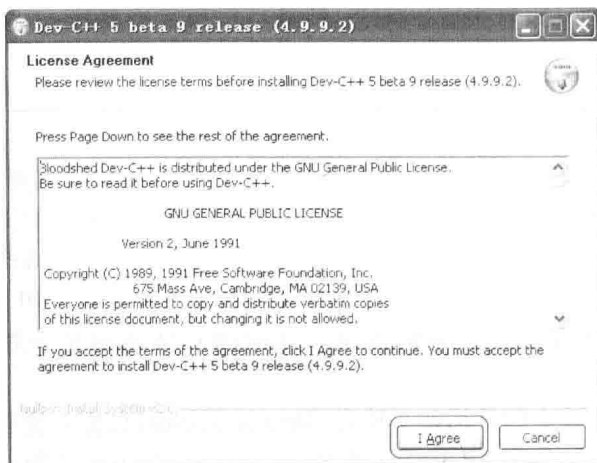


图 1-2 同意 GPL 协议

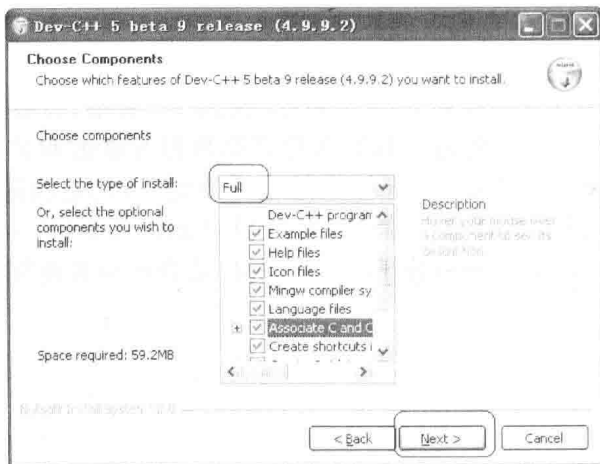


图 1-3 选择所有组件

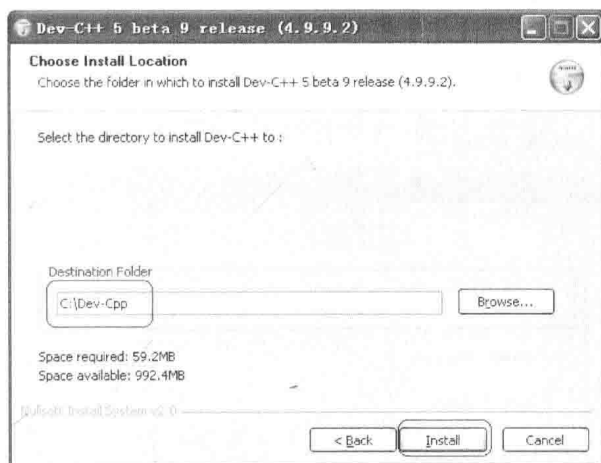


图 1-4 选择安装路径



图 1-5 安装结束

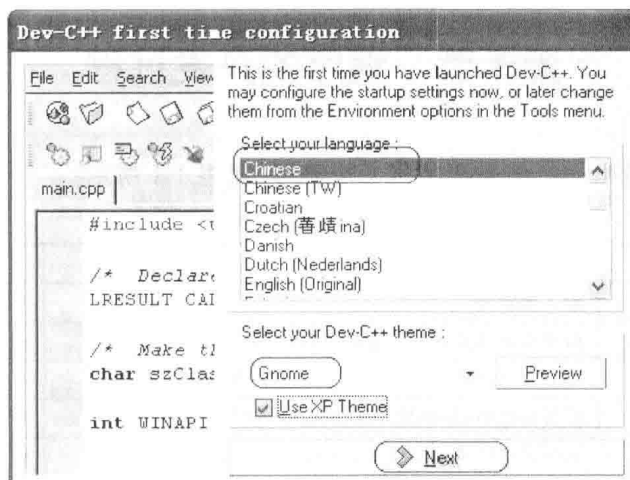


图 1-6 首次运行语言选择

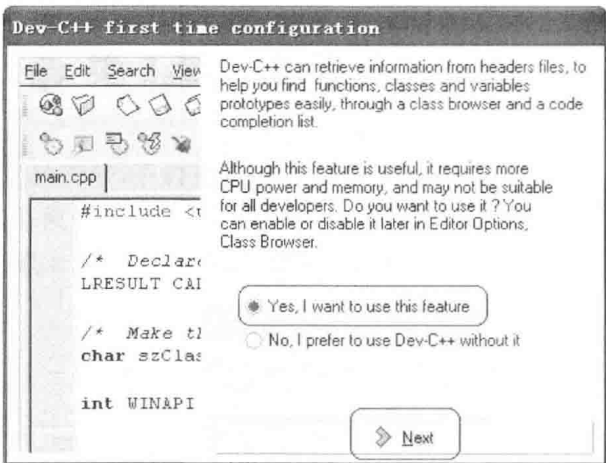


图 1-7 配置头文件查找

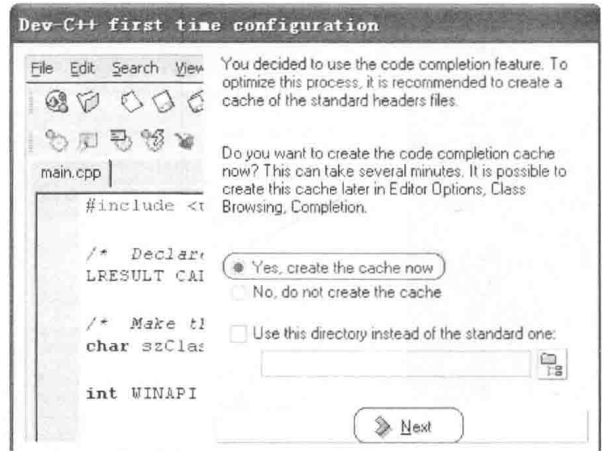


图 1-8 配置头文件缓存

第 3 节 程序的开发过程

C 程序从编写到输出运行结果要经过 6 个阶段,如表 1-1 所示。

表 1-1 C 程序的开发过程

序号	阶段名	说明	结果
1	编辑(edit)	输入源程序	生成. cpp 文件和. h 文件
2 *	预处理(preprocess)	编译器预处理程序产生新的源代码	生成新的源代码
3	编译(compile)	把 C 源代码翻译为机器语言目标代码	生成. obj 文件
4	连接(link)	把目标代码、库函数等连接起来产生可执行文件	生成. exe 文件或. dll 文件
5 *	装入(load)	执行程序时操作系统自动从磁盘读入内存	创建进程
6	执行(execute)	执行程序,产生结果	输出运行结果



以上是一般程序的开发过程。由于预处理和装入阶段是系统自动完成的,所以程序员实际所见的 C 程序开发过程只有编辑、编译、连接和执行 4 个步骤。而 Dev-C++ 更为方便,它的第 6 版将编译和连接合并,只需执行编译就可以直接生成可执行的程序文件。

程序的输入、编译、保存和运行过程分别如图 1-9~图 1-12 所示。

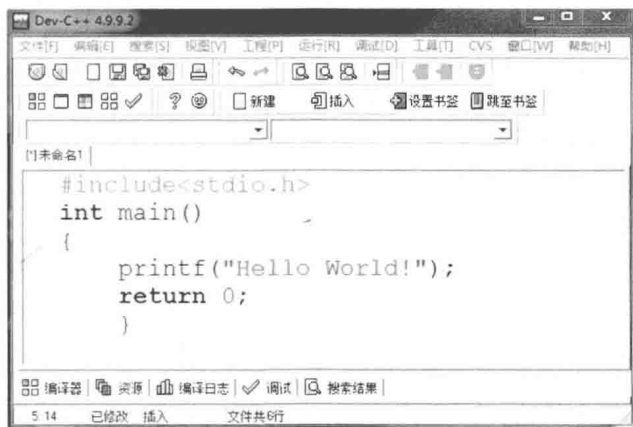


图 1-9 输入源程序



图 1-10 编译源代码



图 1-11 保存源文件

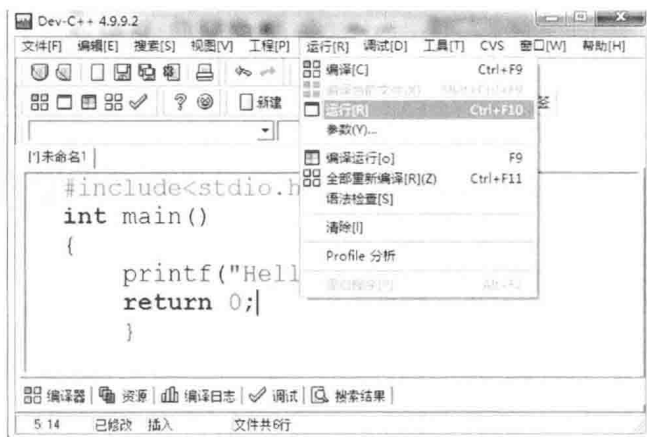


图 1-12 运行程序

第 4 节 程序设计风格

所谓程序设计风格,就是借助好的设计方法编写结构好的程序。程序设计风格反映了一个程序员的基本素养。从初学开始,程序员就应养成良好的编写程序的习惯。就 C 语言而言,程序设计的风格有很多种,并没有一个统一的标准;但每个程序员应该形成自己的编程风格。无论是什么风格,目标只有一个,就是使程序能够清晰、易懂,不造成自己和其他人阅读程序的困难。

【例 1-1】 程序风格示例。

```
#include<stdio.h>

int main()
{
    printf("Hello World!");
    return 0;
}
```

这一程序的作用是输出双引号中的“Hello World!”。该程序的第 4 和第 5 行使用了空格,这些空格可以理解为中文排版的缩进,目的是说明这两条语句的逻辑层次较其他行低;而在第 3 和第 6 行是左对齐,这对对齐的花括号用来描述一个程序的逻辑段落的范围。

注意: 在程序中尖括号<>、圆括号()及花括号{}都是成对出现的。每条语句(在花括号中间的部分)都由分号结尾。

第 5 节 应用实践

本节的目标是使学习者可以最大限度地将自己所学的知识应用到工程中。



Arduino 是目前最为流行的开源硬件。Arduino 是一个开放的硬件平台,包括一个简单、易用的 I/O(输入、输出)电路板,以及一个基于 processing 的软件开发环境。Arduino 既可以被用来开发能够独立运行并具备一定互动性的电子作品,也可以被用来开发与计算机相连接的外围装置,这些装置甚至还能够与运行在计算机上的软件(如 Flash,Max/MSP,Director,Processing 等)进行沟通。通过开放设计的硬件平台和集成开发环境,使用者可以在具备 C 语言知识的基础上对硬件设备进行开发。Arduino 能通过各种各样的传感器来感知环境,并能够控制灯光、马达和读取其他传感器的信号来获得反馈。板子上的微控制器可以通过 Arduino 的编程语言(类似于 C 语言)来编写程序,编译成二进制文件,烧录进微控制器。对 Arduino 的编程是利用 Arduino 编程语言和 Arduino 开发环境来实现的。

Arduino 的硬件设备非常容易获得,主板在淘宝网上的价格不到 20 元人民币,其相关配件也不到 100 元人民币。考虑到语言差异(Arduino 网站是英文的),学习者可以通过必应(www.bing.cn)检索国内的网站获得相应的知识。

推荐 Arduino 就是为了学习者在具备基础知识的基础上能更好地参与工程实践,希望大家利用课余时间积极学习。