

高等学校计算机应用规划教材

C 语言程序设计教程

王娟勤 主 编
晁晓菲 副主编
成宝国 任国霞 吴 昊
胡秋霞

清华大学出版社

北 京

内 容 简 介

本书从培养学生计算思维的角度出发,以案例为引导,介绍了C程序设计基础、基本数据类型、数据运算、程序的基本结构、数组、指针、函数、结构体、共用体和枚举类型、文件和指针的高级应用等内容。书中提供了大量趣味性实例,并做了深入浅出的分析和总结,有助于引领读者理解编程思维和方法;每章有综合案例,为升华知识提供桥梁;各章的知识结构图,有助于学生理清知识脉络;精选的典型习题,为进一步深化基础知识、提高分析问题和解决问题的能力起到了重要作用。

本书采用导学、易学编写策略,每章安排有内容提示、教学基本要求和总结,正文组织本着知识量最小而收获量最大的原则,突出重点和难点。全书逻辑清晰,层次分明,例题丰富,适合作为高等院校本科各专业公共课教材,也可作为计算机相关工程技术人员、计算机爱好者及各类自学人员的参考书。

本书的电子课件、习题答案和实例源文件可以到<http://www.tupwk.com.cn>网站下载。

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 语言程序设计教程 / 王娟勤 主编. —北京:清华大学出版社, 2017
(高等学校计算机应用规划教材)
ISBN 978-7-302-47984-0

I. ①C… II. ①王… III. ①C 语言—程序设计—高等学校—教材 IV. ①TP312.8

中国版本图书馆 CIP 数据核字(2017)第 202071 号

责任编辑:胡辰浩 李维杰

封面设计:孔祥丰

版式设计:牛静敏

责任校对:牛艳敏

责任印制:

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

印 刷 者:

装 订 者:

经 销: 全国新华书店

开 本: 185mm×260mm

印 张: 20.75

字 数: 518 千字

版 次: 2017 年 8 月第 1 版

印 次: 2017 年 8 月第 1 次印刷

印 数: 1~2500

定 价: 48.00 元

产品编号:

前 言

学生学习数学课程、物理课程并不是要成为数学家或物理学家，而是通过这些课程的训练，养成科学思维的素质和能力，这种认识也同样适用于计算机课程。程序设计基础课程对于培养学生计算思维(美国计算机科学家周以真认为：计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动)起着至关重要的作用。

本书以 C 语言为工具，以现实生活中的案例为引导，说明如何分析问题、利用计算机解决问题的思维方法。引入应用程序开发，由浅入深，逐步启发、引领学生学习编写规模逐渐加大的程序，将计算思维的基本思想方法和魅力逐步展现出来，引导读者实现问题求解思维方式的转换——培养计算思维能力。

本书按知识结构组织为 11 章内容，第 1 章为 C 语言程序设计概述，介绍与程序设计有关的概念，说明 C 语言程序的基本组成结构、C 语言程序设计开发环境与过程；第 2 章为 C 语言基础，介绍 C 语言的基本数据类型，常量和变量，C 语言基本运算的运算符、表达式及应用，数据的输入和输出函数的使用；第 3 章为程序设计基本结构，介绍程序的 3 种基本结构，实现选择和循环结构的语句及其应用，介绍常见问题及解决问题的方法；第 4 章为数组，介绍数组的基本概念、使用及处理数组中数据的常用方法；第 5 章为指针，介绍指针的概念，指针的使用，利用指针处理数据的方法；第 6 章为函数，介绍函数的定义、调用及函数间数据传递的方法，说明变量的存储属性及其使用；第 7 章为结构体、共用体与枚举类型，介绍结构体和共用体的定义、使用和区别，介绍了枚举类型的定义和使用；第 8 章为文件，介绍了文件的基本概念，文件的操作步骤，利用文件实现内存和外存中数据交换的方法；第 9 章为底层程序设计，介绍了位运算的运算符、规则及应用；第 10 章为编译预处理，介绍了编译预处理命令的使用、编写大型程序的方法等；第 11 章为指针的高级应用，介绍了多级指针、main 函数带参数、函数指针、动态内存分配及链表。每章提供适合该章知识点的综合案例，以拓展知识、开阔学生的眼界。

全书在内容组织上突出以下特色：

(1) 结构新颖。根据所介绍知识，每章除了安排具有趣味性的实例以外，还安排了有助学生提高和升华知识点的综合案例，这些案例来自于生活或学习中的应用需求，可以让学生在任务的驱动下，由浅入深，学习和编写规模逐渐加大的程序，在潜移默化中逐步让学生了解、学习计算机如何解决问题，从而掌握利用计算机解决问题的方法。

(2) 提供了很多编程“套路”。从典型的程序实例中总结出“套路”，即解决一类问题的方法，从而让初学者迅速掌握基础编程的方法和算法，具有解决实际问题的能力。

(3) 助学。每章安排有内容提示、教学基本要求，例题从问题分析、算法描述、问题总结、注意事项等方面进行完整论述，每章包含知识总结和习题等内容，有利于教师组织

教学，也有助于学生进行预习与复习。

(4) 易学。本着“知识量最小而收获量最大的原则”，突出主线和重点，分解难点，以循序渐进的方法，力求让学生对于难点部分学的轻松，知识点掌握牢固。

(5) 想学。例题选材知识性、趣味性和经典性相结合，尽力降低枯燥度，增强学生学习的“幸福指数”。

(6) 爱做。习题丰富。每章配有形式多样的习题，尽力吸引学生学后爱做、自觉温故知新。

本书由李书琴主审、王娟勤主编。第 1、第 3 章由吴昊编写，第 2、第 4、第 5、第 9 章由王娟勤编写，第 6 和第 7 章由成宝国编写，第 8 章由任国霞编写，第 10 章由晁晓菲编写，第 11 章由胡秋霞编写，并提供了部分例题和习题。

在本书的编写过程中，孙健敏、陈勇和承担《C 语言程序设计》课程的各位老师，在整本书的规划和编写中，对本书提出了很多宝贵意见并给予了帮助，在此表示最诚挚的感谢。

由于编者水平有限，书中的不足、疏漏之处在所难免，恳请广大读者提出宝贵意见和建议。我们的信箱是 huchenhao@263.net，电话是 010-62796045。

本书的电子课件、习题答案和实例源文件可以到 <http://www.tupwk.com.cn> 网站下载。

编 者

2017 年 4 月

书中图标的意义

本书包含一些强调特定知识点的图标，它们能够直观地标识注意、警告、提示、总结和说明等内容。



需要引起注意的内容。



防止错误，容易出错的地方。



提出问题，或大家感觉有疑问的地方。



重点知识，或有总结内容的地方。



程序或知识点说明、论述。



强调特定知识点的地方。



提升知识深度、难度，提高和进阶的内容。



用于标识使用此方法的优势、优点、好处。



问题的“设计思路”。

目 录

第 1 章 C 语言程序设计概述	1	2.6.1 算术运算	28
1.1 C 语言程序的基本组成	1	2.6.2 关系运算	31
1.2 C 语言程序设计的一般步骤	4	2.6.3 逻辑运算	32
1.3 C 语言程序的上机执行过程	5	2.6.4 赋值运算	33
1.3.1 Code::Blocks	6	2.6.5 逗号运算	35
1.3.2 Visual C++ 6.0	8	2.6.6 条件运算符和条件表达式	35
1.4 C 语言学习方法	12	2.6.7 sizeof 运算符	36
1.4.1 为什么要学 C 语言	12	2.6.8 类型转换	37
1.4.2 如何学习 C 语言	13	2.7 数据的输入/输出	40
1.4.3 C 语言学习资源	13	2.7.1 字符数据的输入/输出	40
1.5 案例：程序的铭牌	13	2.7.2 格式化输出和输入函数	41
本章小结	14	2.8 案例：鸡兔同笼	48
习题	15	本章小结	49
习题	15	习题	50
第 2 章 C 语言基础	17	第 3 章 程序设计基本结构	53
2.1 C 语言的字符集	17	3.1 顺序结构	53
2.2 标识符	17	3.2 条件判断——选择结构	55
2.2.1 保留字	17	3.2.1 if 语句	55
2.2.2 用户自定义标识符	18	3.2.2 switch 语句	60
2.3 数据与 C 语言的数据类型	18	3.3 一遍又一遍——循环结构	62
2.3.1 整型数据类型	19	3.3.1 while 循环语句	62
2.3.2 实型数据类型	21	3.3.2 for 循环语句	64
2.3.3 字符型数据类型	22	3.3.3 do...while 循环语句	66
2.4 常量	23	3.3.4 break 和 continue 语句	68
2.4.1 整型常量	23	3.3.5 三种循环语句的比较	70
2.4.2 实型常量	23	3.4 嵌套	70
2.4.3 字符常量	24	3.5 应用举例	73
2.4.4 字符串常量	25	3.5.1 一般计算问题	73
2.4.5 符号常量	25	3.5.2 穷举法求解问题	76
2.5 变量	26	3.5.3 递推和迭代法求解问题	78
2.6 运算符	28	3.5.4 用嵌套的循环求解问题	81

3.6 案例：基因信息处理	83	5.2.4 指针与二维数组	143
本章小结	86	5.3 字符指针	148
习题	87	5.3.1 字符指针与字符串	148
第 4 章 数组	99	5.3.2 字符指针与字符数组	150
4.1 数组的基本概念	99	5.4 指针数组	151
4.2 一维数组	100	5.4.1 指针数组的定义	151
4.2.1 一维数组的定义	100	5.4.2 指针数组的应用	151
4.2.2 一维数组的引用	101	5.5 案例：括号匹配问题	155
4.2.3 一维数组的初始化	102	本章小结	157
4.2.4 一维数组的应用	102	习题	157
4.3 二维数组	110	第 6 章 函数	161
4.3.1 二维数组的定义	110	6.1 函数的定义与调用	161
4.3.2 二维数组的引用	111	6.1.1 函数的定义	162
4.3.3 二维数组的初始化	111	6.1.2 函数调用	164
4.3.4 二维数组的应用	112	6.2 函数间的数据传递	168
4.5 字符数组	114	6.2.1 通过参数传递	168
4.5.1 字符数组的定义	114	6.2.2 通过函数返回值传递	176
4.5.2 字符数组的初始化	114	6.2.3 函数设计的原则	177
4.5.3 字符数组的引用	115	6.3 变量的作用域和生存周期	178
4.5.4 字符串的输入/输出	115	6.3.1 变量的作用域	178
4.5.5 字符串处理函数	118	6.3.2 变量的生存周期	181
4.5.6 应用举例	120	6.4 函数的嵌套调用和递归调用	186
4.6 案例：抽奖嘉年华	122	6.4.1 函数的嵌套调用	186
本章小结	124	6.4.2 函数的递归调用	188
习题	126	6.5 案例：掷骰子游戏	189
第 5 章 指针	135	本章小结	191
5.1 指针的概念	135	习题	193
5.1.1 内存地址和指针	135	第 7 章 结构体、共用体与枚举类型	199
5.1.2 指针变量的声明	136	7.1 结构体	199
5.1.3 取地址运算符和间接寻址运算符	136	7.1.1 结构体的定义	200
5.1.4 指针的引用	138	7.1.2 结构体变量	201
5.2 指针与数组	139	7.1.3 结构体数组	205
5.2.1 指针的算术运算	140	7.1.4 结构体指针	208
5.2.2 指针的比较运算	141	7.1.5 结构体与函数	211
5.2.3 指针与一维数组	141	7.2 共用体	216
		7.2.1 共用体的定义	216

7.2.2 共用体变量的定义	216	9.3 案例：查看内存单元	270
7.2.3 共用体变量的引用	218	本章小结	271
7.3 枚举类型	220	习题	272
7.3.1 枚举类型的定义	220	第 10 章 编译预处理	273
7.3.2 枚举变量的定义和引用	221	10.1 #include 文件包含命令	273
7.4 typedef 类型定义	223	10.2 #define 宏定义	274
7.5 案例：维护通讯录数据库	224	10.2.1 不带参数的宏定义	275
本章小结	228	10.2.2 带参数的宏	276
习题	229	10.3 条件编译	278
第 8 章 文件	235	10.4 编写大型程序	280
8.1 文件概述	235	10.5 案例：数据压缩和解压缩	281
8.1.1 什么是文件	235	本章小结	284
8.1.2 文本文件和二进制文件	235	习题	285
8.1.3 文件类型指针	236	第 11 章 指针的高级应用	287
8.1.4 文件操作	237	11.1 多级指针	287
8.2 文件的打开与关闭	238	11.2 main 函数带参数	288
8.2.1 打开文件	238	11.3 函数指针	289
8.2.2 关闭文件	240	11.4 动态内存分配	291
8.3 文件的读写操作	240	11.4.1 动态内存分配函数	291
8.3.1 按字符读写文件	241	11.4.2 动态内存空间的使用	293
8.3.2 按格式读写文件	245	11.5 链表	295
8.3.3 按数据块读写文件	247	11.6 案例：简单学生成绩管理 系统	300
8.4 文件的定位	250	本章小结	306
8.5 文件出错检测	251	习题	307
8.6 案例：打字练习程序	252	附录	313
本章小结	254	附录 A 字符与 ASCII 码对照表	313
习题	255	附录 B 运算符和结合性	314
第 9 章 底层程序设计	259	附录 C C 语言中的关键字	315
9.1 位运算符	259	附录 D C 常用库函数	315
9.1.1 按位逻辑运算	260	参考文献	323
9.1.2 移位运算	264		
9.1.3 位运算赋值运算符	266		
9.2 位段	267		

第1章 C语言程序设计概述

 **本章内容提示：**介绍 C 语言程序的基本组成、开发 C 语言程序的一般步骤以及两种 C 语言集成开发环境。让初学者对 C 语言程序的组成结构有大致地了解，并介绍 C 语言的学习方法。

 **教学基本要求：**了解 C 语言程序的基本组成，熟练 C 语言程序的开发过程，能够在 C 语言集成开发环境中完成对例题的编辑、编译、连接和运行，并得到正确的结果，为后续章节的学习打下良好的基础，并可以开发一个简单的 C 程序。

计算机语言是人与计算机交流的一种工具，要编写程序、深入地理解计算机的工作原理，就必须学习和掌握计算机语言。在计算机问世的几十年中，出现了多种计算机语言，总体上可以将计算机语言分为机器语言、汇编语言和高级语言三大类。

高级语言接近人类语言和人们习惯使用的数学用语，不依赖于具体的机器，有严格的语法规则。相对于机器语言和汇编语言，高级语言易学易用，所编写的程序易读易改，通用性更强。

C 语言是世界上广泛流行的计算机高级程序设计语言，它于 1973 年由美国贝尔实验室设计发布。由于 C 语言同时具备高级语言的优点和低级语言的功能，而且拥有很好的可移植性，使它成为程序员最喜欢的编程语言之一。它以功能强大、数据结构丰富、目标代码质量高、程序运行效率高、可移植性好等特点成为众多程序员学习程序设计的首选语言。

1.1 C 语言程序的基本组成

下面通过两个例题来了解 C 语言程序的基本组成。

【例 1-1】在屏幕上输出“Hello,World!”。

```
//example1.1 The first C Program
#include <stdio.h>
void main()
{
    printf("Hello,World!\n");
}
```

程序分析：

(1) **main 函数**——从哪里开始，到哪里结束。main()表示“主函数”，每个 C 语言程序都必须有且只能有一个 main 函数，它是每一个 C 语言程序执行的起始点(入口点)。程序是从

main 函数的第一条语句开始执行，然后顺序执行 main 函数中的其他语句。main 函数执行结束，整个程序结束，而不论 main 函数书写在程序中的任何位置。

void 表示该主函数没有返回值。

用 { 和 } 括起来的是主函数 main 的函数体。main 函数中的所有操作(或语句)都包含在这一对花括号之间，即 main 函数的所有操作都在 main 函数体中。

(2) 如何在屏幕上显示文字——函数调用。本程序的“主函数”main 中只有一条函数调用语句——printf() 是 C 语言的库函数，用于程序中数据的输出(显示在屏幕上)。本例是将一个字符串“Hello,World!\n”输出，即在屏幕上显示“Hello,World!”。

(3) 程序的最小独立单元——语句。每条语句用“;”结束。

(4) 编译器如何认识 printf 函数——#include 预处理命令。C 语言规定，函数在调用前必须声明。printf 函数的声明就包含在头文件“stdio.h”中，头文件是每一个 C 程序必不可少的组成部分，因为一个 C 程序至少要包含输入或输出函数。

(5) 程序里的说明书——注释。“/* example1.1 The first C Program*/”为注释，注释是为了改善程序的可读性(提示、解释作用)，在编译、运行时不起作用(编译时会跳过注释，目标代码中不会包含注释)。注释可以放在程序的任何位置，并允许占用多行，只是需要注意“/*”、“*/”匹配，不要嵌套注释。

也可以使用“//”作为注释符，一个“//”只能注释一行，而“/*”和“*/”配对可以注释多行。

注释与软件的文档同等重要，要养成书写注释的良好习惯，这对软件的维护相当重要。因为程序是要给别人看的，自己也许还会看自己几年前编写的程序，清晰的注释有助于读者理解算法和程序的思路。

在程序开发过程中，还可以用注释帮助调试程序，即暂时屏蔽一些不需要运行的语句，以后可以方便地恢复。

【例 1-2】求两个数的较大值。

```
#include <stdio.h>
int max(int x,int y)          /* 计算两数中较大的数 */
{int z;                      /* 声明部分，定义变量 */
    if(x>y)
        z=x;
    else
        z=y;
    return z;                /* 将 z 值返回，通过 max 带回调用处 */
}
void main()
{int a,b,c;                  /*声明部分，定义变量*/
    scanf("%d,%d",&a,&b);
    c=max(a,b);              /* 调用 max，将调用结果赋给 c */
    printf("max=%d",c);
}
```

程序分析:

(1) 本程序包括两个函数。其中，主函数 main 仍然是整个程序执行的起点，函数 max

计算两数中较大的数。

(2) 主函数 main 调用 scanf 函数, 获得两个整数, 分别存入 a、b 两个变量中, 然后调用函数 max, 获得两个数中较大的数, 并赋给变量 c。最后输出变量 c 的值(结果)。

(3) max 是用户自定义的函数, int max(int x,int y)是函数入口, 表示此函数运行时需要获得两个整数值, 数据处理结束后会返回一个整数值。

(4) 函数 max 同样也用{和}将函数体括起来。max 的函数体是函数功能的具体实现, 它从参数表获得数据, 将处理后得到的结果存储于 z 中, 然后将 z 返回调用函数 main。

(5) 本例表明函数除了调用库函数外, 还可以调用用户自定义的函数。



综合上述两个例子, 我们对 C 语言程序的基本组成和程序结构有了一个初步了解:

(1) C 程序由函数构成(函数是 C 程序的基本单位), 所有的 C 语言程序都由一个或多个函数构成。其中, main 函数必须有且只能有一个。

(2) 被调用的函数可以是系统提供的库函数, 也可以是用户根据需要自己设计编写的函数。程序的全部工作由各个函数完成。编写 C 程序就是编写一个个的函数。

(3) main 函数(主函数)是每个程序执行的起始点。无论 main 函数在程序中的哪个位置, 一个 C 程序总是从 main 函数开始执行, 并且也是从 main 函数结束。

(4) 一个函数由函数首部和函数体两部分组成:

① 函数首部: 一个函数的第一行, 由函数类型、函数名、参数类型和参数名组成。一般格式为:

函数类型 函数名(参数类型及参数列表)

② 函数体: 函数首部下方用一对花括号括起来的部分。函数体一般包括声明和执行两部分。

例如函数 max:

函数类型	函数名称	参数类型	参数名	参数类型	参数名	
int	max	(int	x,	int	y)	/*函数首部*/
{						/*函数开始*/
int z;						/*声明部分,定义变量*/
if(x>y)						
z=x;						
else						
z=y;						/*函数体,执行部分*/
return z;						
}						/*函数结束*/

(5) C 程序书写格式自由, 一行可以写几条语句, 一条语句也可以写在多行上。每条语句的最后必须以一个分号“;”表示语句的结束。

(6) 可以使用“/* */”和“//”对 C 程序中的任何部分进行注释。注释可以提

高程序的可读性，使用注释是编程人员必须养成的良好习惯。

一个较大的系统往往由多人合作开发，程序文档、注释是其中重要的交流工具。

(7) C 语言本身不提供输入/输出语句，输入/输出操作是通过调用库函数 (scanf、printf 等) 来完成的。

输入/输出操作涉及具体计算机硬件，把输入/输出操作放在函数中处理，可以简化 C 语言和 C 编译系统，便于 C 语言在各种计算机上实现。不同的计算机系统需要对函数库中的函数做不同的处理，以便实现同样或类似的功能。

1.2 C 语言程序设计的一般步骤

计算机之所以能够产生如此大的影响，其原因不仅在于人们发明了机器本身，更重要的是，人们为计算机开发出了不计其数的能够指挥计算机完成各种工作的程序。正是这些功能丰富的程序给了计算机无尽的生命力，是程序设计工作者智慧的结晶。

所谓程序，是指用计算机语言描述的、为解决某一问题、满足一定语法规则的语句序列，而程序设计就是用某种程序语言编写解决这些问题的步骤的过程。

要设计出一个程序，首先要明确要处理问题中的数据结构(即数据和数据之间的关系)；其次要描述出对问题的处理方法和步骤(即算法)。因此，数据结构与算法是程序设计过程中密切相关的两个方面。著名计算机科学家 Niklaus Wirth 教授提出了关于程序的著名公式(沃思公式)：程序=数据结构+算法。这个公式说明了程序设计的主要任务。

那么，如何用 C 语言进行程序设计呢？一般包含以下步骤：

1. 分析问题

使用计算机解决具体问题时，首先要对问题进行充分的分析，确定问题是什么，解决问题的步骤又是什么。针对所要解决的问题，找出已知的数据和条件，确定所需的输入、处理及输出对象。

2. 确定数据结构和算法

在分析求解问题的基础上，确定程序中数据的类型和组织存储形式，即确定存放数据的结构。针对问题的分析和确定的数据结构，选择合适的算法加以实现。注意，这里所说的“算法”泛指解决某一问题的方法和步骤。

3. 编写与编辑源程序

根据确定的数据结构和算法，用 C 语言把该数据结构和算法严格地用符合 C 语言语法规则的代码描述出来，也就是编写出源程序代码。将源程序输入计算机，并以约定的扩展名“.c”的文本文件形式保存在外存上，例如 file1.c、t.c 等。

用于编辑源程序的软件是编辑程序。编辑程序是提供给用户书写程序的软件环境，可用来输入和修改源程序。

4. 编译、连接与运行

编译是把 C 语言源程序翻译成计算机能识别的二进制指令形式的目标程序。编译过程由编译程序完成。编译程序自动对源程序进行句法和语法检查,当发现错误时,将错误的类型和所在的位置显示出来,提供给用户,以帮助用户修改源程序中的错误。如果未发现句法和语法错误,对目标代码进行优化后生成目标程序。目标程序的文件扩展名是“.obj”。

目标程序尽管计算机能够读懂,但仍不能执行,这是因为其中还缺少一些内容(如库函数、其他目标程序、各种资源等的二进制程序),需要把它们与目标程序“组合起来”,这个过程称为连接。经过连接之后,就生成了可执行程序,可执行程序的扩展名为“.exe”。连接过程是由连接程序(也称链接程序或装配程序)完成的。

运行程序是指将可执行程序投入运行,得到程序处理的结果。如果程序运行结果与预测结果不一致,必须重新回到第 3 步(也有可能从第 1 步开始重新分析问题),对程序进行编辑修改、编译和运行,直到得到正确的结果为止。

与编译、连接不同,运行可以脱离语言处理环境,直接在操作系统环境下进行。

5. 写出程序的文档

程序是提供给用户使用的,如同正式的产品应当提供产品说明书一样,正式提供给用户使用的程序,必须向用户提供程序说明书。内容应包括:程序名称、程序功能、运行环境、程序的装入和启动、需要输入的数据以及使用注意事项等,为程序的使用、维护做好基础工作。

很多软件能够完成从第 3 和第 4 步的全部过程,这种软件称为 C 语言集成开发系统,如 CodeBlocks、Visual C++ 等。在集成开发系统中开发 C 程序的过程可以用图 1-1 表示。

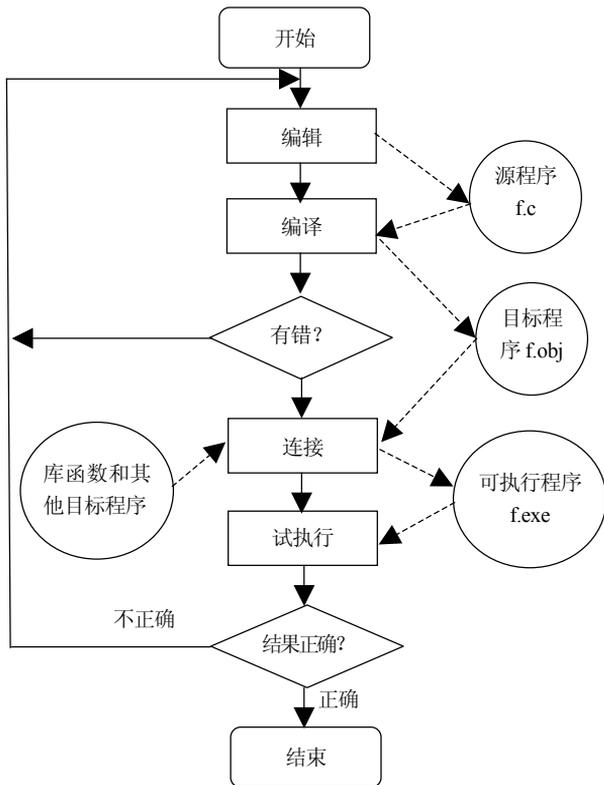


图 1-1 开发 C 程序的步骤

1.3 C 语言程序的上机执行过程

如前所述,编写出 C 程序仅仅是程序设计工作中的一个环节,编写出来的程序需要在计算机上进行调试运行,直到得到正确的运行结果为止。上机调试运行程序一般要经过 4 个步

骤：编辑、编译、连接和运行，才能得到程序的运行结果。编辑、编译、连接和运行通常是在 C 语言集成开发环境中完成的。

1.3.1 Code::Blocks

Code::Blocks 是一个免费、开源、跨平台的 C/C++ IDE 软件，支持 Windows、Linux、Mac OS X 等系统。该软件小巧灵活，具有支持跨平台、代码语法高亮、自动格式化、国际化等功能。

1. 编辑 C 源程序代码

从网上下载 Code::Blocks 免费安装软件，如 Code::Blocks 13.12 版本。先安装 Code::Blocks 软件，启动 Code::Blocks，主界面如图 1-2 所示。

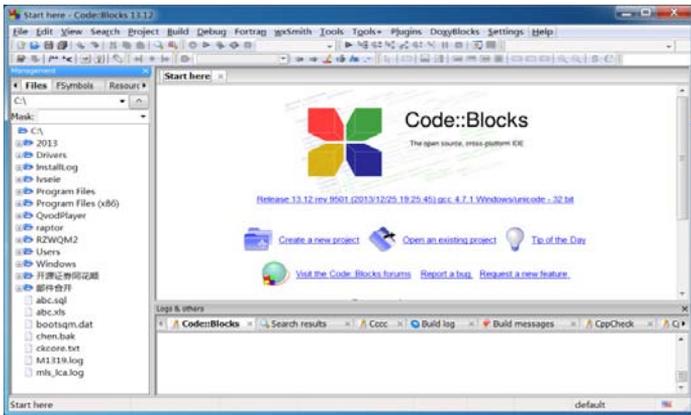


图 1-2 Code::Blocks 主界面

选择菜单“File”→“New”→“File...”，出现“New from template”对话框，选择“C/C++ source”，如图 1-3 所示。

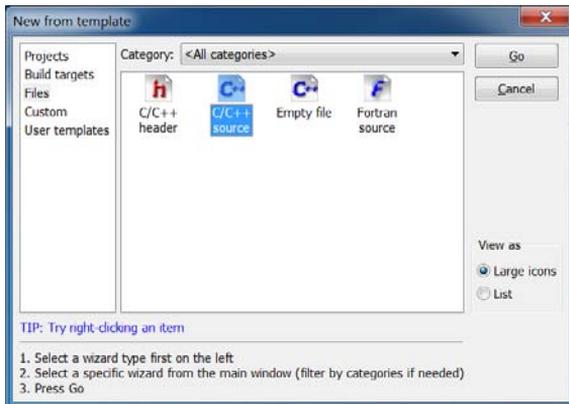


图 1-3 文件类型选择对话框

单击“Go”，进入欢迎界面，这个界面可以跳过，然后进入图 1-4 所示对话框。程序员可以选择用 C 还是 C++，这里选择 C，单击“Next”，进入如图 1-5 所示的文件路径设置对话框。



图 1-4 编程语言选择对话框

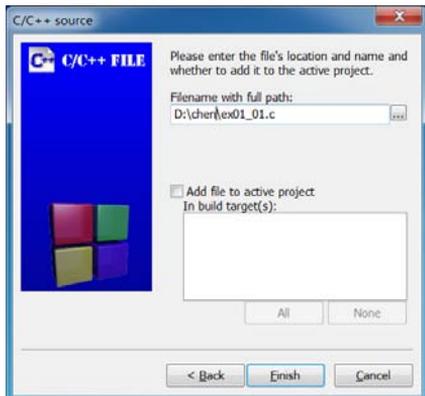


图 1-5 文件路径设置对话框

输入或选择源程序文件的完整描述(含盘符、路径、文件名和扩展名“.c”),单击“Finish”。进入 Code::Blocks 编辑主界面,如图 1-6 所示,就可以正式录入源程序文件了。

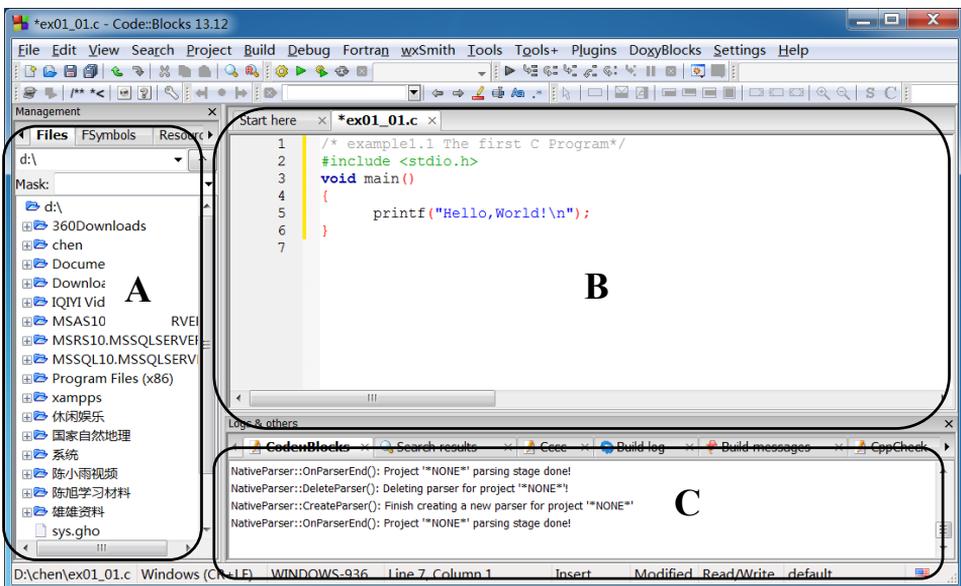


图 1-6 Code::Blocks 编辑主界面

窗口说明:

- A. 文件列表窗口,可以列出指定外存上的所有文件夹及文件。
- B. 源程序编辑窗口,可以对源程序文件进行编辑。
- C. 调试信息输出窗口,用于输出程序编译、连接过程的各种提示信息。

2. 编译、连接和运行程序

在源程序编辑窗口中录入源代码,保存后在工具栏上单击  按钮,完成源程序的编译和连接。如果没有错误,在调试信息输出窗口中会显示生成“.exe”文件。单击工具栏上的  按钮,运行.exe 文件,若程序正确,将会出现如图 1-7 所示的显示结果。

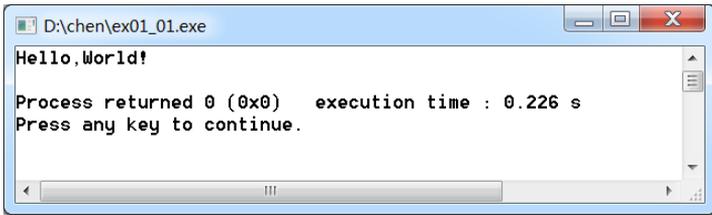


图 1-7 例 1-1 的运行结果

1.3.2 Visual C++ 6.0

Visual C++ 6.0 是 Microsoft 公司推出的运行在 Windows 操作系统中的交互式、可视化集成开发软件，集程序的编辑、编译、连接、运行等功能于一体，为编程人员提供了一个既完整又方便的开发平台。它不仅支持 C++ 语言，也支持 C 语言，是 C++ 语言和 C 语言的集成开发环境。下面简单介绍在 Visual C++ 6.0 环境下开发 C 语言源程序的过程。

1. 编辑 C 源程序代码

启动 Visual C++ 6.0(简称 VC)集成开发环境，在如图 1-8 所示的 VC 主窗口中，选择“文件”→“新建”命令，弹出“新建”对话框。选择“新建”对话框中的“文件”选项卡，并选择“C++ Source File”选项，建立 C 语言源程序，如图 1-9 所示。

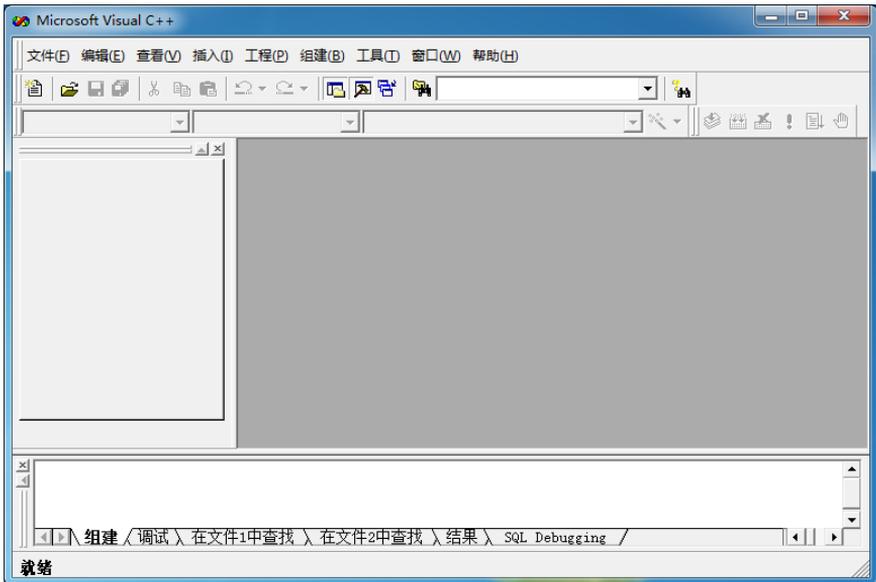


图 1-8 VC 主窗口

在对话框右半部分的“文件名”文本框中输入新文件的名称，如“1-1.c”。对文件命名时，应根据程序的功能进行命名，尽量做到“见名知意”。



图 1-9 “新建”对话框

在“位置”文本框中输入或选择新文件的存储位置。

单击“确定”按钮，进入 VC 主窗口，在编辑窗口中看到光标闪烁时，可输入和修改源程序，如图 1-10 所示。

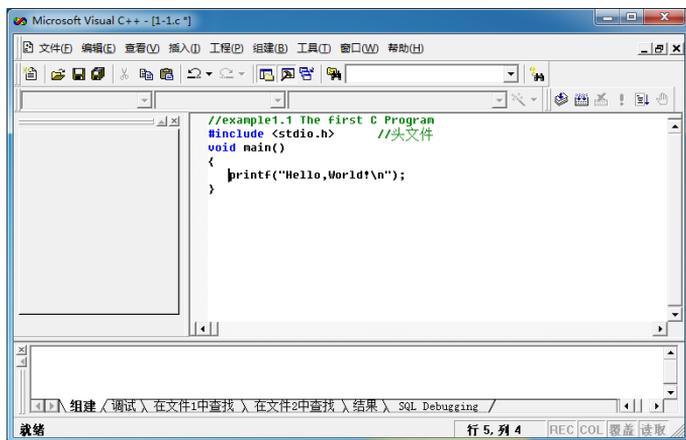


图 1-10 C 源程序编辑窗口

选择“文件”→“保存”命令，或单击工具栏上的“保存”按钮，或按快捷键 Ctrl+S，保存源程序文件。

2. 编译、连接和运行程序

在 VC 系统中，文件在工程的管理之下，而工程则在工作区的管理之下。在编辑 C 源程序代码时没有进行“新建工作区，在工作区下建立工程，在工程下建立 C 源程序代码”的过程。在对 C 源程序代码进行编译时，可为源代码建立默认的工作区和工程。

1) 编译

单击工具栏上的“编译”按钮或选择“组建”菜单栏中的“编译[1-1.c]”命令，使用编译系统对文件“1-1.c”进行编译，弹出图 1-11 所示对话框。单击“是”按钮，表示同意由 VC 系统建立一个默认的工作区和一个工程，并对源程序进行编译；若单击“否”按钮，将不会对源程序进行编译。

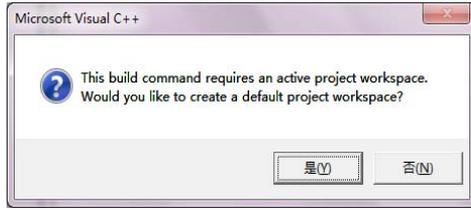


图 1-11 工作区创建对话框

若编译成功，则在 VC 输出窗口中显示“0 error(s),0 warning(s)”，表示既没有致命错误(error)，也没有警告(warning)。编译系统生成一个目标文件 1-1.obj，之后可以进行程序的连接与执行，VC 运行编译后的窗口如图 1-12 所示。

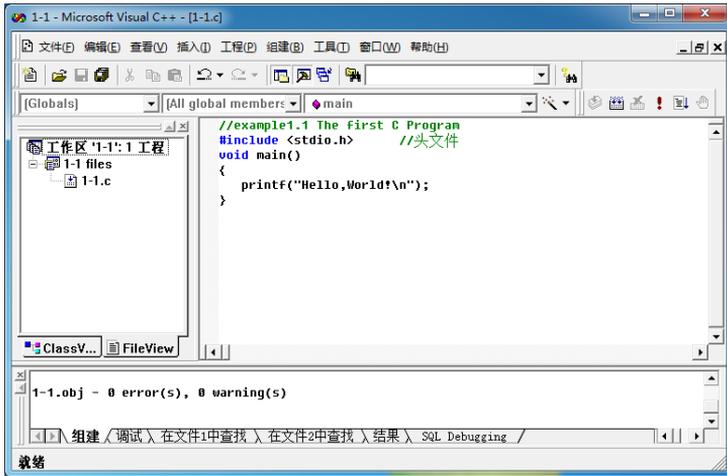


图 1-12 编译后的 VC 窗口

若编译有错，则在输出窗口中显示错误信息。按 F4 功能键或双击错误提示行，在源程序出错行的左侧出现标记，并且光标被定位到此行。此时，应检查标记所在行(可能会是标记前的一行或多行)的程序代码，找出错误的原因并改正，然后再编译。若出现错误，则再修改，直到编译通过为止。

2) 连接

单击工具栏上的“组建”按钮或选择“组建”菜单下的“组建[1-1.exe]”命令，或按快捷键 F7 对目标文件 1-1.obj 进行连接。若连接成功，则生成一个可执行文件 1-1.exe。

3) 运行

单击工具栏上的“执行”按钮或选择“组建”菜单下的“执行[1-1.exe]”命令，或按快捷键 Ctrl+F5，运行后产生结果输出窗口。

程序执行完毕后，单击“文件”菜单下的“保存”，对源程序进行保存，也可以用 Ctrl+S 快捷键来保存文件。



警告：

程序执行、保存完毕后，单击“文件”菜单下的“关闭工作空间”(即关闭本程序的工作区)，如图 1-13 所示，才能编写下一个程序，否则新建的程序会被添加到第一个程序的工程中，两个程序都无法执行。

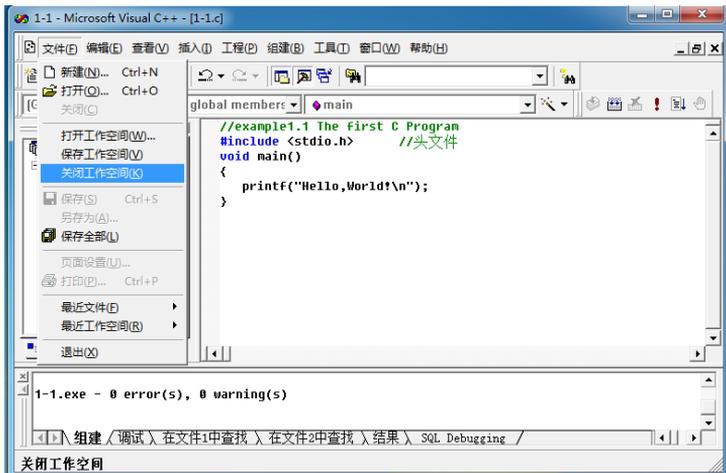


图 1-13 关闭工作空间

工程工作区就像一个“容器”，由它来“盛放”相关工程的所有相关信息。创建默认工程工作区之后，系统将创建一个相应的工作区文件(.dsw)，用来存放与该工作区相关的信息；另外还将创建的其他几个相关文件是：工程文件(.dsp)以及选择信息文件(.opt)等。

开发 C 语言源程序时，也可以采用“创建工程工作区”→“创建工程”→“在工程下创建 C 源程序”的步骤进行，下面简略说明过程：

(1) 在图 1-8 所示的 VC 主窗口中选择菜单“文件”→“新建”，出现如图 1-9 所示的“新建”对话框。选择“工程”选项卡下的“Win32 Console Application”，在右上处的“位置”文本框和“工程名称”文本框中填入工程在磁盘上的存放位置(目录或文件夹位置)以及工程的名字，单击“确定”按钮进入下一个选择界面。这个界面主要询问用户想要构成一个什么类型的工程，其界面如图 1-14 所示。



图 1-14 工程创建界面

(2) 选中“一个空工程”将生成一个空的工程，工程内不包括任何东西。若选中“一个简单的程序”，将生成包含一个空的 main 函数和一个头文件的工程。选中“一个"Hello, World!"程序”与选中“一个简单的程序”没有什么本质上的区别，区别在于所建工程的 main 函数中

是否包含显示“Hello, World!”字符串的输出语句。选中“一个支持 MFC 的程序”，可以利用 VC 提供的类库来进行编程。

(3) 若在步骤(2)中选择建立“一个空工程”，则选择菜单“工程”→“增加到工程”→“新建”选项，在出现的对话框的“文件”标签(选项卡)中，选择“C++ Source File”。在右中处的“文件名”文本框中为将要生成的文件取一个名字，例如取名为 1-1.c。然后单击“确定”按钮，进入源程序的编辑窗口，输入 C 源程序代码，编译、连接、执行即可。

1.4 C 语言学习方法

1.4.1 为什么要学 C 语言

C 语言带给学生的最大好处就是它会为学生打开一扇了解计算机的窗口，在几乎做任何事情都离不开计算机的今天，越了解计算机也就意味着越能利用好计算机。

美国卡内基·梅隆大学计算机科学系前系主任周以真教授在 2006 年发表了一篇著名的文章——*Computational Thinking*，文中谈到“计算机科学的教授应当为大学生开一门称为‘怎么像计算机科学家一样思维’的课，面向非专业的，而不仅仅是计算机科学专业的学生”，这是因为“机器学习已经改变了统计学……计算生物学正在改变着生物学家的思考方式。类似地，计算博弈理论正改变着经济学家的思考方式，纳米计算改变着化学家的思考方式，量子计算改变着物理学家的思考方式”，所以“计算思维代表着一种普遍的认识和一类普适的技能，每一个人，不仅仅是计算机科学家，都应热心于它的学习和运用”。对于不太可能成为专职程序员的学生而言，通过学习编程，了解什么是抽象、递归、复用、折中等计算思维，能在各行各业中更有效地利用计算机工具解决复杂问题。

面向过程的程序设计语言历史上出现过很多种，如 Basic、Pascal、FORTRAN 等。因为种种原因，用的人越来越少，学习的人也就越来越少了。

进入 Windows 时代，出现了很多面向对象的可视化编程语言，如 VC、Java、VB 等，但这些语言的重要基础之一还是面向过程的知识，初学者往往会先被它们华丽便捷的界面设计所吸引，对最基础的面向过程程序设计方法却无法深入学习。学习 C 语言，将可以更准确地定位于计算机思维训练，更有效地理解计算机的工作原理。

C 语言被广泛应用于设备驱动程序、高性能的实时中间件、嵌入式领域、并发程序设计等，也是 IT 行业交流、笔试、面试时最常见的语言。

在过去的 40 多年里，C 语言已成为最重要、最流行的编程语言之一。它的成长归功于使用过的人都对它很满意。更重要的是很多流行语言、新生语言都借鉴了它的思想、语法，如 C++、Java、C# 等。

掌握了 C 语言，其他类似语言不学自通。掌握了 C 语言后，再去学习其他面向过程的语言，最多一个星期就能学会，因为“万变不离其宗”，只是语法上有些许改变，而思想却没有更改。

C 语言久经考验，有现成的大量优秀代码和资料。这就使读者能在过去程序的基础上，

快速和高效地编写新的算法和函数。C 语言是一个开源组织的语言，在全球著名的开源组织网站和国内的一些论坛中，能找到任何想要的开源代码。C 语言使用者众多，讨论者也就众多，有数不尽的资料可以学习和使用。

1.4.2 如何学习 C 语言

C 语言相对其他的程序设计语言而言，程序员发挥空间最大、运行效率最高，给了程序员无限制的自由，这些优点也正是它让 C 语言初学者又爱又恨的原因。如何才能真正地学好 C 语言呢？

(1) 多练、多读优秀代码，光看教材例题是肯定学不好 C 语言的，只有在熟练掌握例题、理解例题的基础上，多编写程序，通过编写和调试程序，积累丰富的编写和调试程序的经验，才能透过 C 语言窥探计算机底层原理，掌握基本的程序设计思维。

(2) 多总结，学习完 C 语言的语法规则和基本算法后，对所学习的知识内容和编写的程序及时进行总结，理清知识的脉络，总结出规律，以便在编写新程序时做到游刃有余。

(3) 多交流，与同学交流，与老师交流，通过网络与网友交流。

1.4.3 C 语言学习资源

因为学习 C 语言的人很多，所以任何问题在网络上都可以找到解决方案。有许多网站既拥有大量的经典案例，又具有很高的人气。以下网站是值得大家交流的好去处。

1) www.csdn.net

全球最大的中文 IT 社区，分为多个版块，免费注册用户后，可以在相应的版块中咨询关于计算机的任何问题。

2) www.cyuyan.com.cn

C 语言网，主要针对 C/C++ 用户的一个交流平台。

3) <https://sourceforge.net>

SourceForge.net 又称 SF.net，是开源软件开发者进行开发管理的集中式场所，也是全球最大的开源软件开发平台和仓库。

SourceForge.net 上拥有大量非常优秀的开源软件，事实上，这些软件完全可以代替一些商业软件。



进阶提高

1.5 案例：程序的铭牌

每个程序都应该包含识别该程序的信息，如程序名、功能、编写目的、编写日期、作者等文档说明(铭牌)，以帮助阅读者了解该程序的大致信息。一般将这类信息放在程序开头并以程序注释的形式出现。例如，对例 1-1 加入该程序的铭牌。

```

/*****
*   Name:example1-1.c   *
*   Purpose:print the message of welcome   *

```

```

* Author:WangJuanqin *
* Date:2016.8.15 *
*****/
#include <stdio.h>
void main()
{
    printf("Hello,World!\n");
}

```

读者可以尝试为自己以后的程序设计一款帅气的铭牌模板，用程序彰显自己的智慧和设计。

本章小结

本章介绍了 C 语言程序的基本组成、开发流程、两种 C 语言程序集成开发环境和 C 语言程序设计的学习方法，本章教学涉及的有关知识的结构导图如图 1-15 所示。

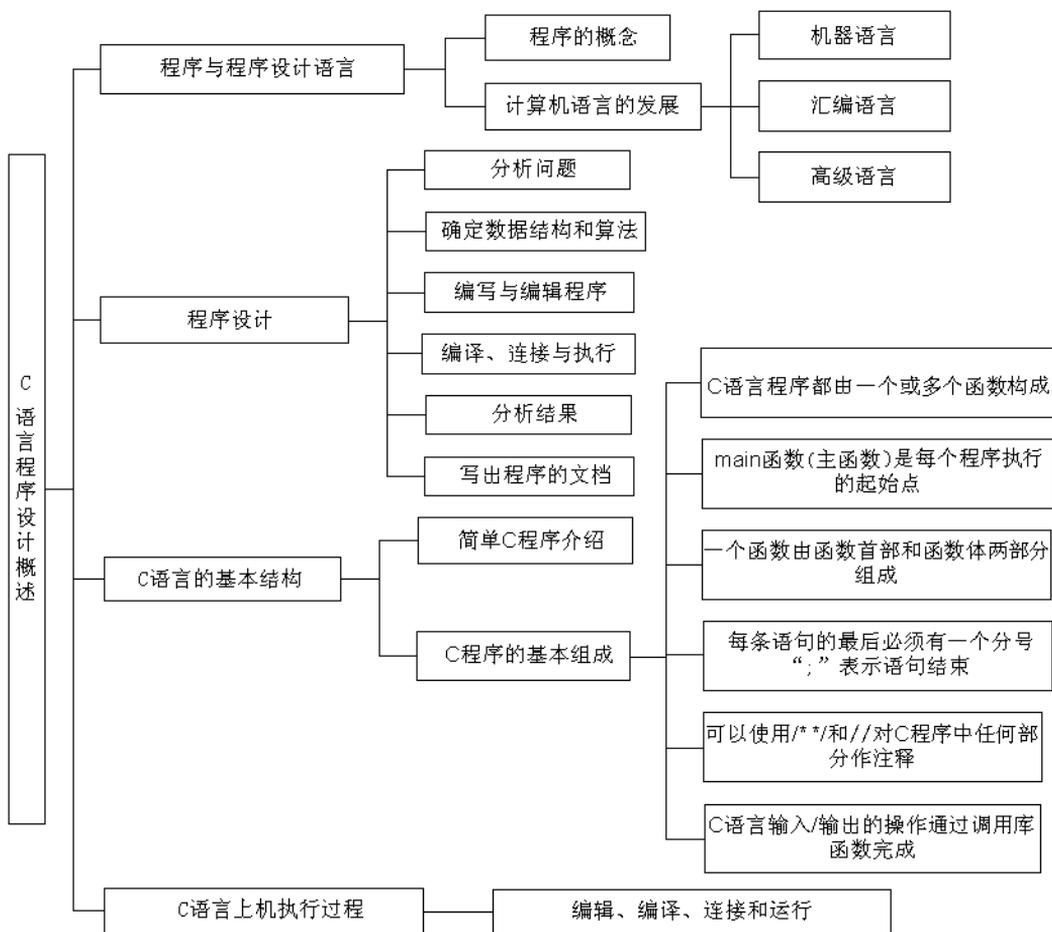


图 1-15 C 语言程序概述知识导图

习 题

一、单选题

1. C语言源程序由()组成。
A. 函数 B. main 函数 C. 子程序 D. 过程
2. 以下关于C语言源程序执行的叙述中, 正确的是()。
A. C语言源程序的执行总是从第一个函数开始, 在最后一个函数中结束。
B. C语言源程序的执行总是从 main 函数开始, 在 main 函数中结束。
C. C语言源程序的执行总是从 main 函数开始, 在最后一个函数中结束。
D. C语言源程序的执行总是从第一个函数开始, 在 main 函数中结束。
3. 以下说法中正确的是()。
A. 一个C函数中只允许一对花括号。
B. 在C语言程序中, 要调用的函数必须在 main 函数中定义。
C. C语言不提供输入/输出语句。
D. C语言程序中的 main 函数必须放在程序的开始部分。
4. C语言集成开发系统提供了C程序的编辑、编译、连接和运行环境, 以下可以不在该环境下进行的是()。
A. 编辑和编译 B. 编译和连接 C. 连接和运行 D. 编辑和运行
5. 下面描述中不正确的是()。
A. C程序的函数体由一系列语句和注释组成。
B. 注释内容不能单独写在一行上。
C. C程序的函数说明部分包括函数名、函数类型、形式参数等的定义和说明。
D. scanf 和 printf 是标准库函数而不是输入和输出语句。
6. 下面描述中正确的是()。
A. 主函数中的花括号必须有, 而子函数中的花括号是可有可无的。
B. 一个C程序行只能写一条语句。
C. 主函数是程序启动时唯一的入口。
D. 函数体包含了函数说明部分。

二、填空题

1. 为解决某一实际问题而设计的语句序列就是_____。
2. 在程序设计中, 把解决问题的方法和有限的步骤称作为_____。
3. 一个完整的C程序至少要有有一个而且只能有一个_____函数。
4. 高级语言源程序代码必须经过翻译才能够运行, C语言采用的翻译方式是_____方式。
5. 开发一个C程序要经过编辑、_____、连接和运行4个步骤, 才能得到运行结果。

6. 目标程序文件的扩展名是_____。
7. 程序连接过程是将目标程序、_____或其他目标程序、各种资源等的二进制程序连接装配成可执行文件。
8. 因为源程序是_____类型的文件,所以它可以用具有文本编辑功能的任何编辑程序完成编辑。

三、编程题

1. 在屏幕上输出：
金
木
水
火
土
2. 编写程序,从键盘输入两个整数,分别求这两个数字的和、差、积和高,并输出结果。
3. 编写程序,从键盘输入梯形的上底、下底和高,求该梯形的面积并输出结果。