



普通高等教育“十二五”规划教材
21世纪大学计算机基础分级教学丛书

C语言 程序设计导学 (第三版)

杜友福 主编



科学出版社

普通高等教育“十二五”规划教材
21 世纪大学计算机基础分级教学丛书

C 语言程序设计导学

(第三版)

杜友福 主编

科学出版社

北 京

版权所有·侵权必究

举报电话:010-64030229;010-64034315;13501151303

内 容 简 介

本书是《C 语言程序设计导学》的第三版,为 C 语言程序设计课程的学习指导和实验教材,既可与《C 语言程序设计(第三版)》配套使用,也可独立使用。本书分为学习指导和实验指导两部分,学习指导部分分为 13 章,每章包括学习要求、内容要点、典型例题分析三个方面的内容;实验指导部分根据 C 语言的教学需要,设计了 11 个实验。学习者通过学习与训练,能够进一步掌握 C 语言的基本概念、C 程序设计的基本方法,并通过上机实验强化程序设计与调试的基本功,理解和掌握 C 语言的基本知识,达到学好和用好 C 语言的目的。

本书特别适合于本、专科非计算机专业的初学者,也可供计算机等级考试者和其他各类学习者使用参考。

图书在版编目(CIP)数据

C 语言程序设计导学/杜友福主编. — 3 版. — 北京:科学出版社,2012.7
普通高等教育“十二五”规划教材. 21 世纪大学计算机基础分级教学丛书
ISBN 978-7-03-035078-7

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

中国版本图书馆 CIP 数据核字(2012)第 148073 号

责任编辑:高 嵘 责任校对:徐文刚

责任印制:彭 超 封面设计:苏 波

科 学 出 版 社 出 版

北京东黄城根北街 16 号

邮政编码:100717

<http://www.sciencep.com>

武汉市首壹印务有限公司印刷

科学出版社发行 各地新华书店经销

开本:787 1092 1/16

2012 年 7 月第 3 版 印张:12

2012 年 7 月第一次印刷 字数:276 000

定价:22.00 元

(如有印装质量问题,我社负责调换)

前 言

C 语言以其独特的功能,受到了广大学生和使用者的喜爱。但又由于 C 语言的语法复杂性和语言灵活性,使得许多初学者学习起来感到有一定的难度,甚至于在整个 C 语言学完后,还不知其所以然。需要掌握哪些知识点,哪些是重点,哪些是难点,为什么会出现这些错误,如何解决这些错误,可能都很茫然。基于此,作者结合多年的教学经验编写了这本《C 语言程序设计导学》,旨在帮助学习者能够学好和用好 C 语言。

本书共分两大部分,第一部分是学习指导,与主教材对应分为 13 章,每章分为三个方面的内容。一是学习要求,给出了需要掌握、了解的内容及基本要求,使学习者有一个明确的方向;二是内容要点,对每一章的知识点进行了归纳和总结,明确了重点,指出了难点,对典型错误的表现、出错的原因进行了分析与解释,使学习者能够做到心中有数,更有效地进行学习;三是典型例题分析,每一道题目指出了要求掌握和了解的知识点。选择题不仅给出了答案,还分析了各种选项对在什么地方、错在何处。填空题在进行比较详尽分析的基础上给出了答案。编程题既给出了程序代码,也对题目所涉及的算法或原理进行了分析。能够使学习者从中去体会 C 语言的基本概念、基本知识、基本方法及程序分析与设计的技巧。第二部分是实验指导,根据 C 语言的教学需要,设计了 11 个实验,每个实验都给出了实验的目的要求、需要预习的内容和具体上机的内容,上机题目有四种:程序验证题、程序填空题、程序改错题和程序设计题。希望学习者通过学习与训练,能够进一步掌握 C 语言的基本概念、C 程序设计的基本方法,并通过上机实验强化程序设计与调试的基本功,理解和掌握 C 语言的基本知识,达到学好和用好 C 语言的目的。

本书由杜友福教授任主编,李新玉副教授和胡必鑫副教授任副主编,第 1、第 2、第 6、第 7、第 8、第 9、第 11、第 12、第 13 章由杜友福编写,第 3、第 4、第 5 章及实验指导由李新玉编写,第 10 章由胡必鑫编写,参加编写的还有周云才教授、周贤善副教授、许新民副教授、付兵副教授、彭元珍副教授、张庆岚副教授以及何黎霞、方琴等教师,全书由杜友福教授负责统稿、定稿。

由于作者水平有限,书中不足之处在所难免,敬请各位读者和专家提出宝贵意见,以帮助作者不断地改进和完善。

杜友福
2011 年 9 月

目 录

学习指导

第 1 章 C 语言程序设计概述	(3)
1.1 学习要求	(3)
1.2 内容要点	(3)
1.3 典型例题分析	(5)
第 2 章 数据类型、运算符与表达式	(10)
2.1 学习要求	(10)
2.2 内容要点	(10)
2.3 典型例题分析	(12)
第 3 章 顺序结构程序设计	(18)
3.1 学习要求	(18)
3.2 内容要点	(18)
3.3 典型例题分析	(20)
第 4 章 选择结构程序设计	(24)
4.1 学习要求	(24)
4.2 内容要点	(24)
4.3 典型例题分析	(27)
第 5 章 循环结构程序设计	(35)
5.1 学习要求	(35)
5.2 内容要点	(35)
5.3 典型例题分析	(39)
第 6 章 数组	(50)
6.1 学习要求	(50)
6.2 内容要点	(50)
6.3 典型例题分析	(53)
第 7 章 函数	(60)
7.1 学习要求	(60)
7.2 内容要点	(60)
7.3 典型例题分析	(63)
第 8 章 用户标识符的作用域和存储类	(70)
8.1 学习要求	(70)

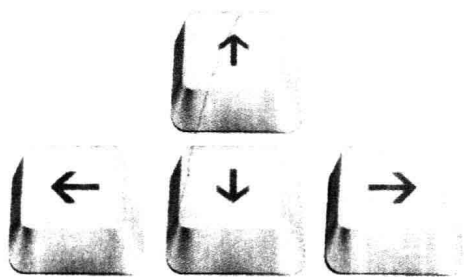
8.2	内容要点	(70)
8.3	典型例题分析	(72)
第9章	编译预处理	(77)
9.1	学习要求	(77)
9.2	内容要点	(77)
9.3	典型例题分析	(79)
第10章	指针	(82)
10.1	学习要求	(82)
10.2	内容要点	(82)
10.3	典型例题分析	(85)
第11章	结构体、共用体 and 用户定义类型	(105)
11.1	学习要求	(105)
11.2	内容要点	(105)
11.3	典型例题分析	(111)
第12章	位运算	(125)
12.1	学习要求	(125)
12.2	内容要点	(125)
12.3	典型例题分析	(126)
第13章	文件	(131)
13.1	学习要求	(131)
13.2	内容要点	(131)
13.3	典型例题分析	(133)

实验指导

实验一	顺序结构程序设计	(143)
实验二	选择结构程序设计	(147)
实验三	循环结构程序设计(一)	(150)
实验四	循环结构程序设计(二)	(154)
实验五	数组(一)	(158)
实验六	数组(二)	(161)
实验七	函数(一)	(164)
实验八	函数(二)	(167)
实验九	指针	(171)
实验十	结构体	(175)
实验十一	文件	(178)
附录	在 Visual C++6.0 集成开发环境下运行 C 程序	(182)

➡ 学习指导

XUE XI ZHI DAO





第 1 章

C 语言程序设计概述

1.1 学习要求

掌握程序及程序设计的基本概念、程序的运行过程、算法及其特点、算法的描述及其表示方法、C 语言程序的基本结构。

熟悉设计程序时应遵循的基本原则。

了解利用计算机解决实际问题的过程、结构化程序设计方法,并在以后的程序设计中应用和逐步熟练掌握。

基本要求:能将日常生活中解决一些简单问题的步骤用算法描述,掌握 N-S 图的算法表示方法。

1.2 内容要点

【知识点】

(1) 程序就是用计算机能够识别的语言所描述的解决某个特定问题的方法和步骤,是由一组相关的指令组成的。

程序 = 数据结构 + 算法

(2) 程序设计就是利用计算机解决问题的全过程。通常是对问题进行分析并建立数学模型;在此基础上设计出相应的数据结构和算法;然后用某种程序设计语言编写相应的程序代码;最后调试程序,使之运行后能产生预期的结果。

(3) 程序设计语言可分为机器语言、汇编语言、高级语言和面向对象的高级语言四类。

(4) 高级语言程序通常要经过编辑、编译后方能在计算机上运行,少数语言以解释方式执行。

(5) 设计程序时应遵循的基本原则包括:正确性、可靠性、简明性、有效性、可维护性和可移植性。

(6) 计算机算法是为计算机解题设计的有明确意义的运算步骤的有限集合。算法应具有有穷性、确定性、有效性、有输入输出等特点。

(7) 算法可以用自然语言、伪代码、传统流程图、结构化流程图等方法表示。

(8) 结构化程序设计方法要求在设计程序时只使用顺序、选择、循环三种基本结构。这三种基本结构有以下共同特点:只有一个入口、只有一个出口、结构内的每一部分都有

机会被执行到、结构内不存在“死循环”(无终止的循环)。

(9) 结构化程序设计方法的基本思路是,把一个复杂问题的求解过程分阶段进行,每个阶段处理的问题都控制在人们容易理解和处理的范围内。结构化程序设计方法的基本原则是:自顶向下、逐步细化、模块化设计、结构化编码。

(10) 函数是 C 程序的基本单位。一个函数由函数的首部和函数体两个部分组成。函数的首部描述该函数的外观(即该函数的名字,该函数需要外部提供哪些信息,执行完毕后能为外部提供哪些结果等),函数体则定义该函数的内涵(即该函数将对什么数据执行何种操作)。

(11) 一个 C 源程序至少包含一个函数(即 main 函数)。一个 C 程序总是从 main 函数开始执行并在该函数结束。main 函数可以放在程序的任意位置。

(12) C 程序书写格式自由,一行内可以写多个语句,一个语句也可以写在多行上。分号是 C 语句的必要组成部分。

(13) C 程序的输入和输出操作是通过调用库函数 scanf 和 printf 来完成的。

(14) C 程序中可以用“/* */”对程序的任何部分作注释;也可以用“//.....”作为一个单独的注释行。注释对程序的执行没有任何影响。

【重点】

(1) 算法的特点及简单算法的设计。

(2) C 程序的基本结构。

【难点】

(1) 算法的设计思想。

(2) 用传统流程图和结构化流程图表示算法。

【典型错误分析】

1. 算法设计中的逻辑错误

对初学者来说,最常见的逻辑错误包括:忘记对计数器、总数等变量进行初始化;选用的用于控制输入循环结束的标记值也是一个合法的输入数据值;一个作为除数的变量的值为 0 等。在用 C 语言编程实现算法后,编译程序往往不能发现这类错误,但程序运行时的结果可能是不正确的。

排除源程序中的错误(包括语法错误和逻辑错误)是进行程序设计所必须具备的能力,是对所学知识的综合检验过程,需要清醒的头脑,严谨细致的工作作风,扎实的基础知识以及逻辑分析能力。对于初学者,调试程序时应注意以下两点:

(1) C 编译程序尽管能够指出大多数程序的语法错误,但不可能完全准确地指出错误发生的源程序行,有时应当综观上下文、参考出错信息进行综合分析判断,不要死抠所报的出错行号。

(2) 程序中一处发生的错误会产生若干连带的错误信息,即若干条错误信息可能只有第一条是真正的错误,改掉这一错误,所有其他出错信息将随之消失。因此查错纠错应从最前面的出错点开始,纠正某个错误后,若不能发现其他错误,则应再次编译源程序,按

照新的提示信息查找错误。

2. 分号使用错误

分号是 C 语句的组成部分,初学者往往容易遗漏分号而造成语句的不完整,导致编译程序对 C 语句无法正确理解。

3. 注释使用错误

正确的注释形式是 `/* 注释内容 */`。初学者容易犯以下错误:忘记了注释的结束符 `*/`,或者注释用 `/*` 开头,或用 `/*` 结束,或者在注释符号中插入了空格,即写成 `/* ..... */` 的形式。

C 编译程序对于注释部分(起始符 `/*` 与结束符 `*/` 之间)是不进行任何处理的,也就是说,编译时,一旦遇到 `/*` 符号就对其后的任何符号不予处理,直到遇到 `*/` 符号为止。而对于其他部分则作为 C 的语句进行处理。因此注释符号使用错误会导致编译程序对程序的错误翻译,从而出现语法错误。

1.3 典型例题分析

一、选择题

例 1.1 C 编译程序是_____

- A. C 程序的机器语言版本
- B. 一组机器语言指令
- C. 将 C 源程序翻译成目标程序的程序
- D. 由制造厂家提供的一套应用软件

解 本题的关键是掌握编译程序的作用。

C 语言编写的源程序要执行,必须经过以下步骤:用编辑程序将源程序输入并存放在计算机的存储设备中,然后通过 C 编译程序将源程序翻译成计算机能识别的目标程序,最后通过连接装配程序将目标指令模块组合成完整的可执行程序才能在计算机上运行。C 编译程序是系统软件的一部分,并非应用软件。C 程序的机器语言版本称为目标程序。因此选项 A、D 均不正确。虽说能够运行的程序都是一组机器语言指令,但选项 B 是笼统的,不准确的。

因此,正确答案为选项 C。

例 1.2 以下叙述中正确的是_____。

- A. C 程序由主函数组成
- B. C 程序由函数组成
- C. C 程序由函数和过程组成
- D. C 程序中的注释由 `/*` 开头,由 `*/` 结束

解 本题的关键是掌握 C 程序的基本构成。

在初学阶段,由于程序较小,可能只需 `main()` 函数即可,但解决规模较大的实际问题的最好办法是把程序分成较小的模块,每一模块完成一个独立的功能,这样便于管理和调试。C 语言的程序模块就是函数,因此读者应该明确且牢记:C 程序是由函数组成的。每个 C 语言的源程序可以包含多个函数,但必须有一个且只能有一个主函数。因此,选项 A 是片面的。在 C 程序中不存在“过程”这一程序结构,故选项 C 也是错误的。

程序设计中可以对源程序进行必要的注释,以便于阅读和维护。C 程序中的注释行

由 / * 开头,由 * / 结束,“/”和“*”必须紧接着,在这两个字符之间不得插入空格。在选项 D 中“/”和“*”之间都插有空格,故选项 D 也是错误的。

因此,正确答案为选项 B。

例 1.3 C 函数由_____构成。

- A. 函数首部和函数体
- B. 执行语句和非执行语句
- C. 自变量和因变量
- D. 定义域和值域

解 本题的关键是掌握 C 函数的组成。

从数学角度来说,一个函数有自变量和因变量,也有其定义域和值域,但从 C 语言的角度来说,函数分为函数首部和函数体两个部分,函数的首部描述该函数的外观,函数体则定义该函数的内涵,各种语句用来在函数体中定义各种数据及对这些数据的操作。

因此,正确答案为选项 A。

例 1.4 一个 C 程序的执行是_____。

- A. 从 main() 函数开始,直到 main() 函数结束
- B. 从第一个函数开始,直到最后一个函数结束
- C. 从第一个语句开始,直到最后一个语句结束
- D. 从 main() 函数开始,直到最后一个函数结束

解 本题的关键是掌握 C 程序的执行过程。

C 程序中必须要有一个而且也只能有一个 main() 函数,不论 main() 函数处于程序中的什么位置,C 程序执行时总是从 main() 函数开始。C 程序的执行与函数位置无关。故选项 B 是不正确的。

C 程序中除了顺序结构的语句执行与排列顺序一致外,选择结构和循环结构都会改变语句的执行顺序。因而选项 C 是不正确的。

如果程序中有多个函数,其他函数要通过调用才能执行。而且被调用函数在执行完毕后,必须返回到调用函数处。故选项 D 也不正确。

因此,正确答案为选项 A。

二、填空题

例 1.5 C 语言源程序文件的后缀是_____,经过编译后生成文件的后缀是_____,经过连接后生成文件的后缀是_____。

解 文件的后缀即文件的扩展名,主要用来表示该文件的类型。一般情况下,C 语言源程序文件的扩展名为 C;C 语言源程序经过编译生成目标程序,其扩展名为 OBJ;目标程序必须经过连接生成可执行程序才能运行,其扩展名为 EXE。

因此,本题答案:C,OBJ,EXE。

例 1.6 在事先不知道需要连续输入多少个数据的情况下,可用一个_____值控制输入循环的终止。

解 这是一道关于算法设计方面的题。算法设计中,经常会在某种条件下重复执行某些操作,这种情况称为循环。例如,要连续输入多个数据,就可以用循环来实现,循环次数有时是固定数,有时是通过条件来控制的。就本题来说,由于不知道需要连续输入多少个数据,此时通常采用的办法是设置一个标记值(flag)来控制循环结束,当输入值为标记值时表示所有数据已输入完毕。显然该标记值应该小心选择,其值不应该在合法输入数

据的范围内。类似的问题在循环中经常会遇到。

因此,本题答案:标记。

例 1.7 结构化程序设计方法使用的三种基本结构是_____、_____和_____。

解 这是一道基本概念题。结构化程序设计方法要求程序中只使用顺序、选择、循环三种基本结构。这三种基本结构构成的算法不存在无规律的转向,属于结构化的算法,而且已经证明,由以上三种基本结构构成的算法可以解决任何复杂的问题。

因此,本题答案:顺序、选择、循环。

三、阅读题

例 1.8 下述算法用于求 $1 * 2 * 3 * 4 * 5$ 。设两个变量,一个变量 p 代表被乘数(同时用于存放每一步骤的当前所得乘积),一个变量 i 代表乘数。算法如下:

S1:使 $p=1$

S2:使 $i=1$

S3:使 $p * i$,乘积仍放在变量 p 中,可表示为 $p * i \rightarrow p$

S4:使 i 的值加 1,即 $i+1 \rightarrow i$

S5:如果 i 不大于 5,返回重新执行步骤 S3 以及其后的步骤 S4 和 S5,否则,算法结束。最后得到 p 的值就是 $5!$ 的值

如何改动使其成为求 $1 * 3 * 5 * 7 * 9 * 11$ 。如果要求 $1+2+\dots+100$,应如何修改?

解 首先分析一下上述算法中每一步的作用。步骤 S1、S2 两步是对变量赋初值,即设置初始乘积为 1,第一个乘数为 1。S3 是进行乘法运算,并将结果存放到当前所得乘积 p 中。S4 通过改变 i 的值得到下一个乘数。S5 判断是否所有的运算已执行完毕。

求 $1 * 3 * 5 * 7 * 9 * 11$ 与求 $1 * 2 * 3 * 4 * 5$ 的差别主要在于,下一个乘数比前一个乘数多 2,且最后一个乘数是 11。通过以上分析可知,只需作很少的改动即可:

S1: $1 \rightarrow p$

S2: $1 \rightarrow i$

S3: $p * i \rightarrow p$

S4: $i+2 \rightarrow i$

S5:若 $i \leq 11$,返回 S3;否则,结束

可以看出,用这种方法表示的算法具有通用性、灵活性。S3 到 S5 组成一个循环,在实现算法时,要反复多次执行 S3、S4、S5 等步骤,直到某一时刻,执行 S5 步骤时经过判断,乘数 i 已超过规定的数值而不返回 S3 步骤为止。此时算法结束,变量 p 的值就是所求结果。

如果对这个算法适当修改实现求累加和,只需改动以下几点:

(1) 使 p 的初值为 0(为什么?请读者考虑)。

(2) S3 中乘法运算改为加法运算。

(3) S5 中改为与 100 比较。

由于计算机是高速进行运算的自动机器,可以高速地、不厌其烦地进行重复性操作。计算机中多次重复执行同类操作称为循环(或迭代)。所有计算机高级语言中都有实现循环的语句。因此,上述算法不仅是正确的,而且是计算机能实现的较好的算法。

四、算法设计题

例 1.9 开发一个程序求全班学生某门课程的平均成绩。要求程序每次运行时班级人数可为任意。

解 采用结构化程序设计方法(“自顶向下、逐步求精”的方法)来解决这个问题并用伪代码来表示求精过程中得到的算法。为使初学者比较容易读懂,伪代码采用中文来表示。首先,顶层描述程序的整个功能,然后将其分解为一系列较小的任务,并按照执行顺序列出这些任务。每一步的求精都是一个完整的算法说明,只是每层的细节在变化。重复上述求精过程直到表示每一任务的伪代码能够转化为程序设计语言的程序段为止。

对于上述问题,顶层可用一句伪代码来表示:

求未知人数班级的平均成绩

首先考虑需要使用的数据量(变量)。显然,本例中需要使用的数据量有:总成绩(记为 total),成绩个数计数器(记为 counter),每次输入的成绩(记为 score),全班平均成绩(记为 average)。

对数据进行分析后,再来考虑程序应该如何完成上述功能。许多程序可从逻辑上分为三个阶段:①初始化阶段(对程序变量初始化);②处理阶段(输入数据值并对程序中的变量作相应调整);③终止阶段(计算和打印最终结果)。

本程序也不例外,因此可以将顶层分解为以下三个小任务:

初始化变量;

输入、求和及对输入的成绩计数;

计算并打印班级平均成绩;

对初始化阶段,由于每次输入的成绩值 score 和平均成绩 average 会被后面的读入过程覆盖,故不需要初始化。做如下求精:

初始化总成绩 total 为 0;

初始化计数器 counter 为 0;

由于本例中成绩个数是不定的,可以使用一个专门的值来指示数据输入的结束,这个值称为“标记值”(flag)。显然标记值必须小心选择,使它有别于准备接收的成绩值。因为考试成绩通常是非负的,可以选择 -1 作为标记值。因此可以让程序重复读入成绩值,当遇到标记值时结束读入。每读入一个成绩值,就将其加到总成绩 total 中,计数器加 1。

因此第二步可求精如下:

读入第一个成绩放到 score 中;

当 score 不等于标记值时,重复以下工作:

```
{  
    将 score 加到 total 中;  
    计数器值 counter 加 1;  
    读入下一个数据放到 score 中;  
}
```

显然,只有读入有效的成绩值后,才能求出平均值。因此第三步可求精如下:

如果计数器值 counter 非零,那么执行以下工作:

```
{  
    将 total 除以 counter 的结果放到 average 中;
```

```
    打印 average;
}
否则执行
打印信息"没有输入成绩"
```

经过上述求精后,所有分解的任务已可以用 C 语言描述出来。如此实现的程序结构清晰、简单明了。本例的 C 语言程序如下:

```
/*用标记值控制循环求全班平均成绩*/
#include <stdio.h>
void main()
{
    float average,total;
    int counter,score;
    /*初始化*/
    total=0.0;
    counter=0;
    /*处理阶段*/
    printf("输入成绩,-1表示输入结束");          /*屏幕提示*/
    scanf("%d",&score);
    while(score!=-1)
    {
        total=total+score;
        counter++;
        printf("输入下一个成绩,-1表示输入结束");/*屏幕提示*/
        scanf("%d",&score);
    }
    /*终止阶段*/
    if(counter!=0)
    {
        average=total/counter;
        printf("全班平均成绩为%.2f\n",average);
    }
    else
        printf("没有输入成绩\n");
}
```



第 2 章

数据类型、运算符与表达式

2.1 学习要求

掌握数据类型的概念及其分类、运算符(算术运算符、赋值运算符、逗号运算符、自增(减)运算符)的作用及用法以及由各种运算符组成的相应表达式的使用。

熟悉运算规则(运算符的优先级和结合性)、表达式计算中的类型转换规则。

了解各种类型的数据在计算机中的存储形式。

基本要求:能正确地将数学公式用表达式表示,能分析各种表达式的运算结果。

2.2 内容要点

【知识点】

(1) C 语言的基本数据类型包括整型(基本整型、短整型、长整型、无符号基本整型、无符号短整型、无符号长整型)、实型(单精度实型、双精度实型)和字符型。

(2) C 语言中的常量可以是字面形式,也可以用一个标识符来代表(称为符号常量)。符号常量必须先定义,后使用,其值在其作用域内不能改变,也不能被赋值。

(3) 在 C 语言中,要求对所有用到的变量做强制定义(即指定变量的数据类型),也就是“先定义,后使用”。变量在计算机的内存中占据一定的存储空间,在相应的存储空间中存放着变量的值。每一变量用一个变量名来标识,变量占据的存储空间中存放的值称为变量值。

(4) 标识符由字母、下划线、数字三种字符组成,且第一个字符必须是字母或下划线。C 语言中有三类标识符:关键字、预定义标识符、用户标识符。用户标识符一般用来给变量、函数、文件等命名。用户标识符的取名应做到“见名知义”,以增加程序的可读性。习惯上用大写字母来标识符号常量,用小写字母来标识变量或函数。

(5) 整型数据在内存存放是精确的,实型数据在内存存放存在着误差,字符型数据所存放的是字符的 ASCII 代码值。各种数据类型的取值是有一定范围的,超出取值范围称为“溢出”。

(6) 字符常量是用一对单引号括起来的单个字符。C 语言提供“转义字符”主要用来表示一些控制动作或特殊字符。

(7) 字符串常量是用一对双引号括起来的由任意多个字符组成的字符序列。每一个

字符串的结尾用字符'\0'作为字符串的结束标志。

(8) C 语言中的运算处理功能都是由运算符来实现的。在使用 C 语言的运算符时, 应该注意运算符的优先级和结合性。优先级用于决定运算中不同级别运算符的运算顺序, 结合性则决定同一优先级的运算符之间的运算顺序。

(9) 用运算符将运算对象(常量、变量、函数)按一定的规则连接起来的式子称为表达式。表达式都有一个值, 即运算结果。

(10) C 语言中的算术运算具有封闭性的特点, 它要求参加运算的两个运算量是同一数据类型, 而运算结果也是该数据类型。如果两个运算量的数据类型不一致, 则系统会进行数据类型转换, 将其中的一个低级别类型的数据向另一个高级别类型的数据转换, 然后才进行相应的算术运算, 运算的结果为其中高级别类型的数据。可以使用强制类型转换运算符对数据类型进行转换。

(11) 赋值运算符就是数学中的等于号“=”, 由赋值运算符将一个变量和一个表达式连接起来的式子称为赋值表达式, 其作用就是将赋值运算符右边的表达式的值赋给左边的变量。赋值运算时, 系统自动将赋值运算符右边的数据转换成赋值运算符左边变量的类型进行存储。

(12) 自增、自减运算符的作用是使变量的值自动加、减 1。 $++i$ 是先使 i 加 1, 再使用 i 的值, 而 $i++$ 是先使用 i 的值, 再使 i 加 1。

(13) C 语言中逗号是一种运算符, 由逗号运算符将若干个独立的表达式连接起来构成逗号表达式, 逗号表达式的值是最后的那个表达式的值。但使用逗号表达式的作用主要是可以求出逗号表达式中的每个表达式的值。

(14) sizeof 运算符用于求得一个变量或某种类型的量在计算机内存中所占的字节数。

【重点】

- (1) 标识符的正确使用。
- (2) 各种基本数据类型的使用。
- (3) 算术运算符的优先级和结合性。
- (4) 表达式计算中的类型转换。

【难点】

- (1) 各种类型数据在计算机内存中的存储形式。
- (2) 自增自减运算。

【典型错误分析】

1. 标识符中大小写字母用错

C 语言中的标识符是严格区分大小写的, 如果大小写字母用错, 编译程序会将其作为不同的标识符对待。

2. 赋值运算符的左边出现表达式

赋值运算是将运算符右边表达式的值存放于左边变量的存储空间中, 因此赋值运算符的左边只能是变量, 不允许出现表达式。虽然赋值号用的是数学中的等于号, 但赋值运算的意义不同于数学中的等式。如数学中 $a=a+1$ 是矛盾的, 而在 C 语言中是一合法且