

精品课程·名师讲堂丛书

C 语言程序设计 辅导讲案

——主讲教材《C 语言程序设计》
(清华·第二版)

夏清国 主编

西北工业大学出版社

【内容简介】 本书主讲教材为谭浩强编著的《C 语言程序设计》(第二版,清华大学出版社)。全书共 15 讲,每讲内容包括本讲内容聚焦、典型例题、课后作业三个部分。书后有三个附录:主讲教材《C 语言程序设计》各章习题精选详解,课程考试真题,课后作业和课程考试真题参考答案。

本书可作为大学生自学指导、考研人员系统复习、任课教师教学的参考用书,也可作为考研辅导班的教材。

图书在版编目(CIP)数据

C 语言程序设计辅导讲案/夏清国主编. —西安:西北工业大学出版社,2008.2

(精品课程·名师讲堂丛书)

ISBN 978-7-5612-2336-9

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

中国版本图书馆 CIP 数据核字(2008)第 001992 号

出版发行:西北工业大学出版社

通信地址:西安市友谊西路 127 号 邮编:710072

电 话:(029)88493844 88491757

网 址:www.nwpup.com

印 刷 者:陕西天元印务有限公司

开 本:850 mm×1 168 mm 1/32

印 张:9.375

字 数:308 千字

版 次:2008 年 2 月第 1 版 2008 年 2 月第 1 次印刷

定 价:15.00 元

前 言

本书是西北工业大学出版社为落实和推动教育部实施的“精品课程建设工程”和“教育质量工程”，而精心策划和组织编写的“精品课程·名师讲堂”丛书之一，主讲教材为谭浩强编著的《C 语言程序设计》（第二版，清华大学出版社），兼顾其他版本的 C 语言程序设计教材。全书共 15 讲，每讲内容包括：本讲内容聚焦——内容要点精讲，知识结构图解，重点、难点点击；典型例题；课后作业。典型例题均为精选的、具有代表性的题目，并进行详细分析解答；课后作业均选自教材每章后的配套习题。书后有三个附录：主讲教材《C 语言程序设计》各章习题精选详解、课程考试真题、课后作业和课程考试真题参考答案。

本书倾注了编者对 C 语言程序设计课程教学内容和教学方法苦心研究的成果，对 C 语言程序设计课程内容深度、新度、广度、量度的理解和把握，对长期教学实践经验的凝结与升华，对国内外知名大学优秀教学成果的学习与借鉴，以及对各类考试重点、热点、难点规律的认识与探索。

本书从指导课程教学、学习和考试的角度，通过对大量涉及内容广、类型多、技巧性强的习题的解答，揭示了 C 语言程序设计的解题方法、解题规律和解题技巧，这对于提高读者分析问题的能力，理解基本概念和理论，开拓解题思路，全面提高 C 语言编程素质，会收到良好的效果。本书可作为大学生自学、任课教师教学的参考用书，也可作为考研辅导班的教材。

本书由夏清国、姚群、王宇翔编写，书中所有程序由夏翼整理并进行调试，夏清国负责全书的组织与定稿。在此书的编写与出版过程中，西北工业大学计算机学院和西北工业大学出版社给予了大力支持和帮助，在此谨表诚挚谢意。

编者

2007 年 11 月

目 录

第 1 讲	C 语言概述	1
1.1	本讲内容聚焦	1
1.2	典型例题	3
1.3	课后作业	4
第 2 讲	程序的灵魂——算法	5
2.1	本讲内容聚焦	5
2.2	典型例题	9
2.3	课后作业	12
第 3 讲	数据类型、运算符与表达式	14
3.1	本讲内容聚焦	14
3.2	典型例题	25
3.3	课后作业	28
第 4 讲	最简单的 C 语言程序设计——顺序程序设计	29
4.1	本讲内容聚焦	29
4.2	典型例题	35
4.3	课后作业	36
第 5 讲	选择结构程序设计	37
5.1	本讲内容聚焦	37
5.2	典型例题	44
5.3	课后作业	46
第 6 讲	循环控制	47
6.1	本讲内容聚焦	47
6.2	典型例题	51

6.3	课后作业	56
第7讲	数组	57
7.1	本讲内容聚焦	57
7.2	典型例题	66
7.3	课后作业	69
第8讲	函数(一)	70
8.1	本讲内容聚焦	70
8.2	典型例题	81
8.3	课后作业	91
第9讲	函数(二)	92
9.1	本讲内容聚焦	92
9.2	典型例题	97
第10讲	预处理命令	99
10.1	本讲内容聚焦	99
10.2	典型例题	104
10.3	课后作业	108
第11讲	指针(一)	109
11.1	本讲内容聚焦	109
11.2	典型例题	129
11.3	课后作业	134
第12讲	指针(二)	135
12.1	本讲内容聚焦	135
12.2	典型例题	148
12.3	课后作业	151
第13讲	结构体与共同体	152
13.1	本讲内容聚焦	152
13.2	典型例题	172
13.3	课后作业	176

第 14 讲	位运算	177
14.1	本讲内容聚焦	177
14.2	典型例题	182
14.3	课后作业	184
第 15 讲	文件	185
15.1	本讲内容聚焦	185
15.2	典型例题	194
15.3	课后作业	198
附录		199
附录一	主讲教材课后习题精选详解	199
附录二	课程考试真题	240
附录三	课后作业和课程考试真题参考答案	252

第 1 讲

C 语言概述

本讲涵盖了主讲教材第 1 章的内容(2 学时)。

1.1 本讲内容聚焦

4 4 4 4



一、 内容要点精讲

(一) C 语言出现的历史背景

(二) C 语言的特点

C 语言的主要特点:

- (1) 语句简练、紧凑、使用方便、灵活。
- (2) 运算符丰富。
- (3) 数据结构丰富,具有现代化语言的各种数据结构。
- (4) 具有结构化的控制语句。
- (5) 语法限制不太严格,程序设计自由度大。
- (6) 可以进行位操作,能实现汇编语言的大部分功能,可用来写系统软件。
- (7) 生成目标代码质量高,程序执行效率高。
- (8) 程序的可移植性好。

(三) 简单的 C 程序介绍

1. C 程序的组成

(1) C 程序组成。C 程序是由函数构成的,一个 C 源程序至少有一个函数,即 main 函数,也可以由一个 main 函数和若干个其他函数构成。

(2) C 程序中函数之间的关系。构成 C 程序的函数有两大类:一类是主函数,另一类是子函数。在主函数和子函数之间,主函数可以调用子函数,而子函数不能调用主函数;在子函数之间,函数之间可以彼此调用,没有层次之分。

2. 函数的组成

例:求两数之最大值。

```
main( )
{ int  a , b , c ;
  scanf ( "%d,%d" , &a , &b ) ;
  c = max(a , b ) ;
  printf ( "max=%d\n" , c ) ;
}

int max(int x , int y)          /* 自定义函数 */
{ int z ;
  if ( x>y ) z=x ;
  else z=y ;
  return(z) ;
}
```

(1) 函数的首部。函数首部包括函数名、函数类型、函数属性、函数参数(形参)名、参数类型。

(2) 函数体。函数体,即函数首部下面的大括弧{……}内的部分。它由两部分组成:

1) 声明部分。定义函数中所用到的变量,以及对该函数中需要调用的函数进行声明。

2) 执行部分。由若干语句组成。C 程序允许无执行部分,即一个空函数,表明它什么也不做。

3. 几点说明

(1) 一个 C 程序,无论 main 函数在程序中的什么位置,程序的执行都将从 main 函数开始,在 main 函数中结束。

(2) 程序书写格式自由,一行内可写几个语句,一个语句也可以分写在多行上。

(3) 每个语句和数据定义的最后必须有一个分号。分号是 C 语言的必要组成部分。

(4) C 语言本身没有输入输出语句,输入输出操作是由库函数 scanf 和 printf 等来完成的。

(5) 可以用 /* … */对 C 程序中的任何部分进行注释,在执行程序时,系

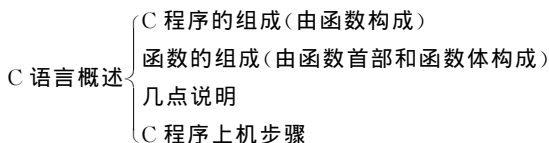
统不会执行注释部分。

(四) C 程序上机步骤

- (1) 调用 C 程序。
- (2) 编辑。
 - 1) 建立新文件或打开一个已有的文件。
 - 2) 编辑源文件。
- (3) 编译源程序。
 - 1) 分步完成。
 - 2) 一次完成。
- (4) 执行程序。
- (5) 退出 Turbo C 集成环境。



二、知识结构图解



1.2 典型例题

例 1.1 输出字符串: This is a C program.

```
main( )
{
    printf("This is a C program. \n");
}
```

【分析】

本例中主函数内只有一个输出语句, printf 是 C 语言中的输出函数。双引号内的字符串按原样输出, “\n”是换行符, 语句最后有一个分号, 表示语句的结束。

解 输出一行信息: This is a C program.

【评注】 C 程序可以没有输入语句, 但必须有输出语句, 否则程序就无意

义了。

例 1.2 求 a,b,c,d 四数之最大值。

```
main( )
{ int  a , b , c , d , m ;
  scanf ( "%d,%d,%d%d" , &a , &b , &c , &d ) ;
  m = max(a , b ) ;
  m = max(m , c ) ;
  m = max(m , d ) ;
  printf ( "max=%d\n" , m ) ;
}

int max(int x , int y)
{ int z ;
  if (x>y) z=x ;
  else z=y ;
  return(z) ;
}
```

【分析】

(1) 本例中包含两个函数:主函数 main 和被调用的子函数 max。max 函数的作用是将 x 和 y 中较大者赋给变量 z。return 语句将 z 的值返回给主调函数 main。

(2) scanf 是 C 语言中的输入函数,其作用是输入 a,b,c,d 的值。scanf 与 printf 都是 C 系统提供的标准输入输出函数。

(3) &a,&b,&c,&d 中的“&”符号表示“取地址”。

(4) scanf 中的“%d”表示以十进制整数形式输入,printf 中的“%d”表示以十进制整数形式输出。

解 若分别给 a,b,c,d 赋值为 8,5,7,10,则解得的结果为 10。

【评注】 C 源文件是由函数构成的,但一个 C 源文件必须有一个,也只能有一个主函数(main 函数)。

1.3 课后作业

习题 1.4,习题 1.5,习题 1.6,习题 1.8。

第 2 讲

程序的灵魂——算法

本讲涵盖了主讲教材第 2 章的内容(2 学时)。

2.1 本讲内容聚焦



一、 内容要点精讲

一个程序应该包括:

- (1) 对数据的描述。指定数据的类型和数据的组织形式,即数据结构。
- (2) 对操作的描述。即操作步骤,也即算法。

故 程序 = 数据结构 + 算法

或 程序 = 算法 + 数据结构 + 程序设计方法 + 语言工具和环境

(一) 算法的概念

1. 算法

算法是一套操作方案,是为解决一个问题而采取的方法和步骤。

2. 算法分类

算法可分为数值运算算法和非数值运算算法两大类。

- (1) 数值运算算法。目的是求数值解。
- (2) 非数值运算算法。包括面十分广泛,主要用于事物管理领域。

(二) 简单算法举例

(三) 算法的特性

- (1) 有穷性。指一个算法应包含有限的操作步骤,而不能是无限的。
- (2) 确定性。指算法中的每一个步骤都应当是确定的。
- (3) 有零个或多个输入。一个算法可以没有输入,也可有多个输入,所谓输入是从外界取得必要的信息。
- (4) 有一个或多个输出。输出即一个算法所得的结果。

(5)有效性。指算法中的每一个步骤都应当有效地执行,并得到确定的结果。

(四)怎样表示一个算法

1. 用自然语言表示算法

用语言表示各种操作。

2. 用流程图表示法

用一些图框表示各种操作。用图形表示算法,直观形象,易于理解。

3. 三种基本结构及流程图

(1)顺序结构。顺序结构是最简单的一种结构,其执行过程是按事先排好的顺序进行。顺序结构流程图如图 2.1 所示。

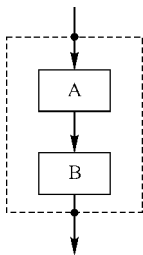


图 2.1

(2)选择结构。选择结构也称分支结构。此结构中包含一个判断框,根据给定的条件 p 是否成立而选择执行 A 框或 B 框。选择结构流程图如图 2.2、图 2.3 所示。

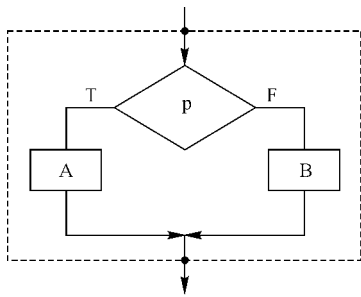


图 2.2

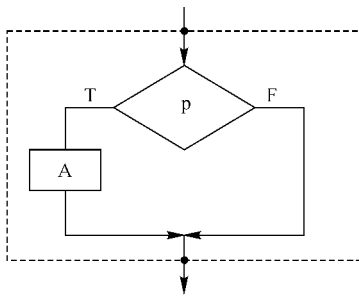


图 2.3

(3)循环结构。

1)当型循环结构。当型循环结构的功能是当给定条件 p_1 成立时,执行 A 框操作,执行完 A 后,再判断条件 p_1 是否成立,如果仍然成立,再执行 A 框,如此反复执行 A 框,直到某一次 p_1 条件不成立为止。当型循环结构流程图如图 2.4(a)所示。

2)直到型循环结构。直到型循环结构的功能是先执行 A 框,然后判断给定 p_2 条件是否成立,如果 p_2 条件不成立,再执行 A 框,然后再对 p_2 条件作判

断,如果 p_2 条件仍然不成立,再执行 A 框,如此反复执行 A 框,直到给定的 p_2 条件成立为止。直到型循环结构流程图如图 2.4(b)所示。

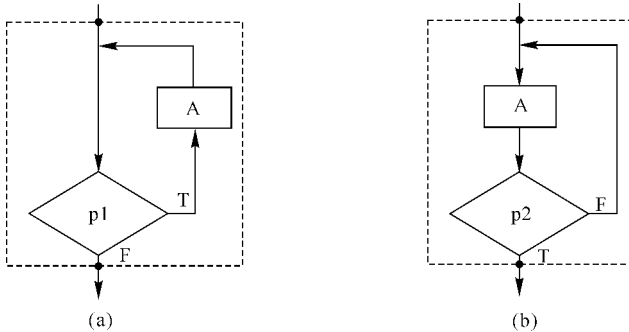


图 2.4

(4) 三种基本结构的共同特点。①只有一个入口;②只有一个出口;③结构内的每一部分都有机会被执行到;④结构内不存在“死循环”(无法终止的循环)。

4. 用 N-S 流程图表示算法

(1) N-S 流程图的流程图符号。

1) 顺序结构。顺序结构的 N-S 流程图符号如图 2.5 所示。

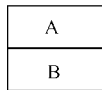


图 2.5

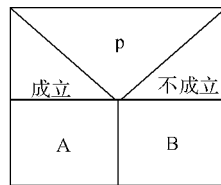


图 2.6

2) 选择结构。选择结构的 N-S 流程图符号如图 2.6 所示。当 p 条件成立时执行 A 操作, p 条件不成立时执行 B 操作。

3) 循环结构。当型循环结构:当型循环结构的 N-S 流程图符号如图 2.7 所示。当 p_1 条件成立时反复执行 A 操作,直到 p_1 条件不成立为止。

直到型循环结构:直到型循环结构的 N-S 流程图符号如图 2.8 所示。先执行 A 操作,然后判断 p_2 条件, p_2 条件成立再执行 A 操作,当 p_2 条件不成立时终止循环。

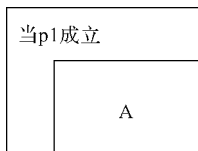


图 2.7

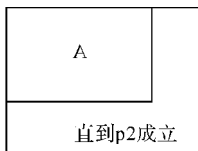


图 2.8

(2) 几点说明。

- 1) 各种结构的 N-S 流程图符号是一个整体,它代表一个基本结构。
- 2) 一个结构化的算法是由一些基本结构顺序组成的;在基本结构之间不存在向前或向后的跳转,流程的转移只存在于一个基本结构范围之内。

5. 用伪代码表示算法

伪代码是用介于自然语言和计算机语言之间的文字和符号来描述算法。它如同一篇文章,自上而下地写出来。每一行(或几行)表示一个基本操作。

6. 用计算机语言表示算法

用计算机语言编写程序描述算法。这时必须严格遵守所用语言的语法规则,这是和伪代码不同的。

(五)结构化程序设计方法

结构化程序设计方法:

(1) 自顶向下、逐步细化的方法。自顶向下、逐步细化的方法是将一个完整的问题分解成若干相对独立的问题,只要这些问题能分别得到正确的解决,整个问题也就解决了。子问题又可进一步分解为若干个子问题,这样可一直重复下去,直到每个问题都已简单到使作者认为可以直接将各子问题表达为文字语句为止。对每步分解,都要做出分解方法的决策。不同的决策会导致不同的解法。

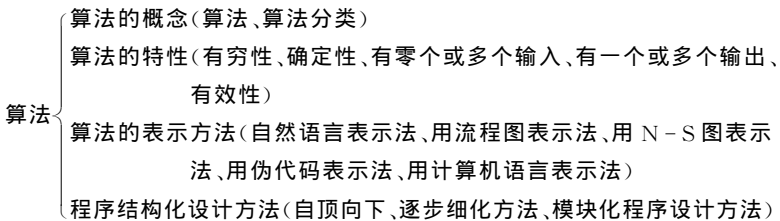
(2) 模块化程序设计方法。模块化程序设计方法是指在程序设计中,将一个复杂的算法(或程序)分解成若干个相对独立、功能单一的模块,以便利用这些模块适当地组合成所需要的全局算法(或程序)。这里所说的模块,是与通常

子算法、子程序或过程极为相似的一个重要概念,这是一个可供调用(即让其他模块调去使用)的相对独立的某操作块(或程序段),每个模块必须是由三种基本结构组成的结构化模块。

(3) 结构化编码。结构化编码是指在设计好一个结构化的算法后,还要善于进行结构化编码。即用高级语言语句正确地实现三种基本结构。



二、 知识结构图解



2.2 典型例题

例 2.1 分别用(1)自然语言,(2)流程图,(3)N-S图,(4)伪代码,(5)计算机语言表示出求 5! 的算法。

【分析】 $5! = 1 \times 2 \times 3 \times 4 \times 5$, 求解过程如下:

步骤 1: 求 1×2 , 得到结果 2;

步骤 2: 用步骤 1 得到的乘积 2 乘以 3(即 $1 \times 2 \times 3$), 得到结果 6;

步骤 3: 将 6 再乘以 4(即 $1 \times 2 \times 3 \times 4$), 得到 24;

步骤 4: 将 24 再乘以 5(即 $1 \times 2 \times 3 \times 4 \times 5$), 得到 120。这是最终结果。

解 根据上面的分析,可以设 2 个变量,分别代表被乘数和乘数,并将每一步骤的乘积存放在被乘数变量中。假设 t 为被乘数, i 为乘数,用循环算法求结果,算法表示如下:

(1) 自然语言表示法。算法如下:

S1: 使 $t = 1$

S2: 使 $i = 2$

S3: 使 $t \times i$, 乘积仍存放在变量 t 中,可表示为 $t \times i \Rightarrow t$

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 以及后面的步骤 S4 和 S5; 否则, 算法结束。

- (2) 流程图表示法。流程图见图 2.9。
- (3) N-S 图表示法。N-S 图见图 2.10。
- (4) 伪代码表示法。算法如下:

开始

置 t 的初值为 1

置 i 的初值为 2

当 $i \leq 5$, 执行下面操作:

使 $t = t \times i$

使 $i = i + 1$

(循环体到此结束)

打印 t 的值

结束

- (5) 计算机语言表示法。用 C 语言表示如下:

```
main()
{int i,t;
  t = 1;
  i = 2;
  while(i<=5)
  {t = t*i;
   i = i+1;
  }
  printf("%d",t);
}
```

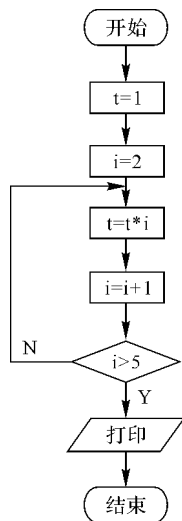


图 2.9



图 2.10

【评注】 注意五种表达方式的特点与区别。

例 2.2 确定 $1 \sim n$ 之间的素数。(1) 用流程图表示求解的算法; (2) 用 N-S 图表示求解的算法。

【分析】 所谓素数, 是指除了 1 和该数本身之外, 不能被其他任何整数整除的数, 如 13 是素数, 因为它不能被 2, 3, 4, 12 整除。