

计算机及应用专业
计算机信息管理专业

专科

全国高等教育自学考试

高级语言程序设计习题详解

黄明 梁旭 周丽梅 编著



机械工业出版社
CHINA MACHINE PRESS

44
12

TP312-44
H862:12

全国高等教育自学考试

高级语言程序 设计习题详解

(计算机及应用专业、计算机信息管理专业 专科)

黄明 梁旭 周丽梅 编著



机械工业出版社

本书是根据“全国自学考试（计算机及应用专业、计算机信息管理专业 专科）高级语言程序设计考试大纲”以及历年考题编写的。全书共分4部分：第1部分是笔试应试指南；第2部分是笔试题解；第3部分是模拟试卷及参考答案；最后是附录，包括考试大纲和2002年上半年试卷及参考答案。

本书紧扣考试大纲，内容取舍得当，叙述通俗易懂，附有大量与考试题型类似的习题及答案，以检查读者对考点的掌握程度。

本书适用于准备参加全国自学考试（计算机及应用专业、计算机信息管理专业 专科）的考生，也可作为大专院校和培训班的教学参考书。

图书在版编目（CIP）数据

高级语言程序设计习题详解/黄明等编著. —北京：机械工业出版社，2004.4
（全国高等教育自学考试）

ISBN 7-111-14263-2

I. 高... II. 黄... III. 高级语言—程序设计—高等教育—自学考试—解
题 IV. TP312-44

中国版本图书馆CIP数据核字（2004）第026214号

机械工业出版社（北京市百万庄大街22号 邮政编码100037）

策 划：胡毓坚

责任编辑：孙 业

责任印制：洪汉军

北京中加印刷有限公司印刷·新华书店北京发行所发行

2004年5月第1版·第1次印刷

787mm×1092mm 1/16·12印张·293千字

0001—5000册

定价：19.00元

凡购本图书，如有缺页、倒页、脱页，由本社发行部调换

本社购书热线电话（010）68993821、88379646

封面无防伪标均为盗版

出版说明

全国高等教育自学考试指导委员会推出面向社会的高等教育自学考试, 经过 10 多年的实践, 已建立起一整套较为完善的规章制度和操作规程, 考试组织严密规范, 考试纪律严格; 坚持考试标准, 实行教考分离, 确保了毕业生的质量。它为没有机会进入高等学校的中国公民提供了接受高等教育的机会, 并以严格的国家考试保证了毕业生的质量, 获得了普遍赞誉。国家自考中心于 2002 年开始执行新的考试计划。新计划中开设的专业共 224 个, 其中专科 141 个占 63%, 独立本科段 61 个占 27%, 专本衔接专业 22 个占 10%。为帮助、指导广大自考生深入理解计算机及相关专业考试的基本概念, 灵活运用基本知识, 掌握解题方法和技巧, 熟悉考试模式, 进一步提高应试能力和计算机水平, 特编写了以下专业的基础课与专业课主要课程的习题详解。

- ◇ 计算机及应用专业 独立本科段
- ◇ 计算机信息管理专业 独立本科段
- ◇ 计算机网络专业 独立本科段
- ◇ 计算机及应用专业 专科

丛书特点:

1. 以 2002 年最新考试大纲为基准

本丛书是根据 2002 年最新考试大纲, 为参加全国高等教育自学考试考生编写的一套习题详解教材。

2. 例题反映了历届考试中的难度和水平

书中对大量的例题进行了分析, 所选例题都是在对最近几年考题深入研究的基础上精心筛选的, 从深度和广度上反映了历届考试中的难度和水平。

3. 作者经验丰富

本丛书的作者都是多年从事全国高等教育自学考试辅导的高等院校的教师。

读者对象:

- ◇ 准备参加全国高等教育自学考试的考生。
- ◇ 计算机及相关专业的本专科生。

即 刻 出 版

前言

自学考试是对自学者进行以学历考试为主的国家高等教育学历考试。本书是为帮助和指导广大考生深入理解自学考试的基本概念，灵活运用基本知识，掌握解题方法和技巧，熟悉考试模式，进一步提高应试能力和计算机水平而编写的。

全书共分 4 部分，即笔试应试指南、笔试题解、模拟试卷及参考答案和附录。书中所选例题均是在对历年真题深入研究的基础上精心筛选的，从深度和广度上反映了考试的难度和水平。模拟试卷的题型分配与真题一致，这些题目是考试指导教师的多年积累，在辅导班中多次实际使用过。

书中附录给出了“全国自学考试（计算机及应用专业、计算机信息管理专业 专科）高级语言程序设计考试大纲”，以及“2002 年上半年全国自学考试高级语言程序设计试卷及参考答案”。

本书由大连铁道学院黄明、梁旭、周丽梅编写。

由于编者水平有限，编写时间仓促，书中错误和不妥之处在所难免，请读者和专家批评指正。

读者在使用本书的过程中如有问题，可通过 E-mail 与我们联系：

dlhm@263.net

编 者

目 录

出版说明

前言

第 1 部分 笔试应试指南

- 1.1 笔试应试策略 2
- 1.2 笔试考点归纳 3
 - 1.2.1 概述 3
 - 1.2.2 基本数据类型和运算 6
 - 1.2.3 顺序结构、选择结构和循环结构的
程序设计 14
 - 1.2.4 数组 19
 - 1.2.5 指针 22
 - 1.2.6 函数 25
 - 1.2.7 结构型、共用型和枚举型 30
 - 1.2.8 文件 34
 - 1.2.9 编译预处理与带参数的主函数 38

第 2 部分 笔试题解

- 2.1 概述 42
 - 2.1.1 单项选择题 42
 - 2.1.2 填空题 43
 - 2.1.3 习题 44
- 2.2 基本数据类型和运算 46
 - 2.2.1 单项选择题 46
 - 2.2.2 填空题 48
 - 2.2.3 习题 50
- 2.3 顺序结构、选择结构和循环结构的
程序设计 54
 - 2.3.1 单项选择题 54
 - 2.3.2 填空题 58
 - 2.3.3 程序分析题 59
 - 2.3.4 程序填空题 62
 - 2.3.5 程序设计题 62
 - 2.3.6 习题 65

2.4	数组	69
2.4.1	单项选择题	69
2.4.2	填空题	73
2.4.3	程序分析题	74
2.4.4	程序填充题	77
2.4.5	程序设计题	78
2.4.6	习题	80
2.5	指针	83
2.5.1	单项选择题	83
2.5.2	填空题	85
2.5.3	程序分析题	87
2.5.4	程序填充题	88
2.5.5	程序设计题	89
2.5.6	习题	91
2.6	函数	94
2.6.1	单项选择题	94
2.6.2	填空题	97
2.6.3	程序分析题	98
2.6.4	程序填充题	102
2.6.5	程序设计题	103
2.6.6	习题	105
2.7	结构型、共用型和枚举型	108
2.7.1	单项选择题	108
2.7.2	填空题	111
2.7.3	程序分析题	112
2.7.4	程序填充题	113
2.7.5	程序设计题	114
2.7.6	习题	114
2.8	文件	117
2.8.1	单项选择题	117
2.8.2	填空题	120
2.8.3	程序分析题	120
2.8.4	程序填充题	121
2.8.5	程序设计题	122
2.8.6	习题	123
2.9	编译预处理与带参数的主函数	125
2.9.1	单项选择题	125
2.9.2	填空题	127
2.9.3	程序分析题	129

2.9.4	程序填空题	129
2.9.5	程序设计题	130
2.9.6	习题	130
2.10	习题参考答案	132

第 3 部分 模拟试卷及参考答案

3.1	模拟试卷一及参考答案	146
3.1.1	模拟试卷一	146
3.1.2	模拟试卷一参考答案	152
3.2	模拟试卷二及参考答案	154
3.2.1	模拟试卷二	154
3.2.2	模拟试卷二参考答案	160

附 录

附录 A	全国自学考试（计算机及应用专业、 计算机信息管理专业 专科）高级 语言程序设计考试大纲	164
附录 B	2002 年上半年全国自学考试高级语 言程序设计试卷及参考答案	176
参考文献		184

1.1 笔试应试策略

全国自学考试(计算机及应用专业、计算机信息管理专业 专科)高级语言程序设计考试大纲涵盖了概述,基本数据类型和运算,顺序结构、选择结构和循环结构的程序设计,数组,指针,函数,结构型、共用型和枚举型,文件,编译预处理与带参数的主函数9章内容。使用的教材是由全国高等教育自学考试指导委员会组编,迟成文编著的《高级语言程序设计》,2000年3月由经济科学出版社出版。考生在考试复习的过程中要紧紧围绕大纲的知识点,首先应对大纲涉及的各章基本概念熟练掌握。

第1章为概述。本章属于基础知识,重点是C语言的字符集和保留字;C语言词类和语句的分类;C程序的基本结构,主要了解程序、程序设计、高级语言的概念。约占4分。

第2章为基本数据类型和运算,重点是基本数据类型常量的书写方法和变量的定义、赋值、初始化、使用方法;基本运算符的运算规则及优先级别;表达式的构成规则和计算。难点是变量生存期和作用域;运算符优先级别;混合表达式计算;逻辑值表示方法。约占20分。

第3章为顺序结构、选择结构和循环结构的程序设计。重点是赋值语句的格式与功能;字符和格式输入/输出函数的调用格式与功能;if、if-else和switch语句的格式和功能;while、do-while、for、break、continue、goto语句的格式和功能;顺序结构、三种选择结构、三种循环结构的程序设计方法;Turbo C环境下,建立、编译修改、编译连接、运行C程序的方法。难点是格式输入/输出函数中格式控制符的使用;break和continue语句的正确使用;Turbo C的各种菜单项的功能。约占25分。

第4章为数组。重点是一维数组、二维数组、字符型数组的定义;初始化和数组元素的使用;字符串处理函数的使用。难点是字符串与字符数组的区别;存放若干字符和存放字符串的字符型数组的差别,应掌握有关处理字符串的系统函数的使用方法。约占9分。

第5章为指针。重点是指针变量的定义、初始化和赋值方法;利用指向变量、数组、字符串的指针变量来正确引用变量、数组元素、字符串;取变量、数组、字符串地址的方法。难点是指针的概念;指向多维数组的指针变量和多维数组的地址。通过本章学习,要求掌握地址、指针、指针变量的概念;能正确定义所需类型的指针变量;能正确地将指针变量指向某变量或数组;能正确地利用指针变量来引用所指向的变量或数组;了解指针数组和多级指针的概念。约占9分。

第6章为函数。重点是函数的定义和调用方法;调用函数时数据传递方法;常用系统函数的使用。难点是值传递和地址传递的区别;系统函数的使用。通过本章学习,要求熟练掌握用户函数的结构、设计方法和调用方法;掌握函数调用中数据传递的几种方式;会设计简单的嵌套调用函数;了解递归调用函数、指针型函数的概念;能正确使用书中介绍的各种常用系统函数。约占20分。

第7章为结构型、共用型和枚举型。重点是结构型、共用型、枚举型数据的特点和定义;结构型变量、数组、指针变量的定义、初始化和成员引用方法;共用型和枚举型变量的定义和引用方法。难点是嵌套的结构型数据的处理。考生应掌握用户自定义类型的定义和使用。约占5分。

第8章为文件。重点是缓冲文件系统中“文件型指针”的概念；有关文件操作(打开、关闭、读写、定位、错误检测)的系统函数使用。难点是有关文件处理的各种系统函数的使用，特别是读写函数种类较多、功能相近，造成记忆和选择上的困难，要求能设计对文件进行简单处理的实用程序。约占5分。

第9章为编译预处理与带参数的主函数。重点是无参宏的定义和宏替换；带参宏的定义和宏替换；文件包含处理。难点是有参宏的替换；主函数中参数的引用方法。通过本章学习，要求掌握宏定义和宏替换的一般方法；包含文件的处理方法；了解条件编译的作用和实现方法；了解带参数的主函数设计和运行方法。约占3分。

考生在复习时应根据大纲里提供的考核点和考核要求来进行复习，这样就能抓住重点，进行有效复习；在做练习时，要根据考试的题型进行练习，在掌握基本概念的基础上，掌握一定的解题技巧。高级语言程序设计的考试题型有：单项选择题、填空题、程序分析题、程序填充题和程序设计题等题型。对于不同题型，要采用不同的答题方法。

单项选择题：这种题型可考查考生的理解、推理分析、综合比较能力，评分客观。在答题时，如果可以，直接得出正确答案，对于没有太大把握的试题，可以采用排除法，经过分析比较加以逐步排除错误答案，最终选定正确答案。

填空题：这种题型常用于考核考生观察能力和运用有关公式、原理的能力。在答题时，无论有几个空，回答都应明确、肯定，考生在复习中最好的应对办法是对学科知识中最基本的知识、概念、原理等要牢记，并结合若干概念和知识点，通过分析和推理来获得正确答案。

程序分析题：这种题型常用于考核考生分析问题和解决问题的能力。最常遇到的就是写出程序的运行结果和程序的基本功能。在答题时，要认真阅读每条语句的功能，前后贯穿，尤其要注意程序主题的功能，它将决定输出结果的正确性和整个程序功能的正确性。

程序填充题：这种题型常用于考核考生分析问题和解决问题的能力。最常遇到的就是写出程序的运行结果、程序的基本功能和填写某一语句。在答题时，要逐条理解语句的功能，然后确定程序的算法，根据算法确定应填写的内容。

程序设计题：这种题型常用于考核考生的综合能力。要求考生全面掌握知识，在答题时，可以自己独立设计，即根据题意设计算法，结合题目要求，选择合适的语句来完成算法步骤，特别要注意如果是复合语句，千万不要忘记“{}”和数据的存放。另外还可以套用以前学过的算法并根据题目要求加以修改。

考生在复习过程中，在掌握知识点的同时也应抓住这些题型的特点，这样才能达到好的应试效果。

1.2 笔试考点归纳

1.2.1 概述

1. 程序与程序设计

从自然语言角度来说，程序是对解决某个问题的方法步骤的描述；从计算机角度来说，程序是用某种计算机能理解并执行的计算机语言描述解决问题的方法步骤。

2. 高级语言与C语言

程序设计的关键是将问题及解决问题的算法过程描述出来，设计人员为用户提供了一种

描述算法过程很方便，同时脱离了对机型要求的、能在任何计算机上运行的计算机语言。程序设计人员可以利用这种计算机语言直接写出各种表达式来描述简单的计算过程，利用这种语言提供的各种控制语句可以描述复杂的加工处理过程。专家们将这种语言称为“高级语言”，而将二进制语言和汇编语言统称为“低级语言”。由于高级语言是面向问题和算法过程描述的，所以又将高级语言称为“面向问题的语言”。

C 语言的前身是 ALGOL 语言。1960 年 ALGOL60 版本推出后，很受程序设计人员的欢迎。用 ALGOL60 来描述算法很方便，但是它离计算机硬件系统很远，不适宜用来编写系统程序。1963 年英国剑桥大学在 ALGOL 语言基础上增添了处理硬件的能力，并命名为“CPL(复合程序设计语言)”。CPL 由于规模大，学习和掌握困难，没有流行开来。1967 年剑桥大学的马丁·理查德对 CPL 语言进行了简化，推出“BCPL(基本复合程序设计语言)”。1970 年美国贝尔实验室的肯·汤普逊对 BCPL 进行了进一步的简化，突出了硬件处理能力，并取了“BCPL”的第一个字母“B”作为新语言的名称。同时用 B 语言编写了 UNIX 操作系统程序。1972 年贝尔实验室的布朗·W·卡尼汉和丹尼斯·M·利奇对 B 语言进行了完善和扩充，在保留 B 语言强大的硬件处理能力的基础上，扩充了数据类型，恢复了通用性，并取了“BCPL”的第二个字母作为新语言的名称。此后，两人合作，重写了 UNIX 操作系统。C 语言伴随着 UNIX 操作系统成为一种很受欢迎的计算机语言。

1977 年，为了让 C 语言脱离 UNIX 操作系统，成为在任何计算机上都能运行的通用计算机语言，卡尼汉和利奇(K&R)撰写了《C 程序设计语言》一书，对 C 语言的语法进行了规范化的描述，成为当时的标准。随着微型机的普及，出现了不同的 C 语言版本，为了统一标准，美国标准化协会(ANSI)于 1987 年制定了 C 语言的标准，称为“ANSI C”。通常将 K&R 的标准称为旧标准，将“ANSI C”称为新标准。

3. C 语言的基本词法

在 C 语言程序中允许出现的所有基本字符的组合称为 C 语言的字符集。C 语言的字符集就是 ASCII 字符集，主要分为下列几类：

- 1) 大小写英文字母(52 个)。
- 2) 数字(10 个)。
- 3) 键盘符号(33 个)。
- 4) 转义字符。转义字符是由“反斜杠字符(\)”开始，后跟单个字符或若干个字符组成的，通常用来表示键盘上的控制代码或特殊符号。

在 C 语言的程序中有特殊含义的英语单词称为“保留字”，主要用于构成语句，进行存储类型和数据类型定义。

C 语言规定，标识符是由字母或下划线开头的字母、数字、下划线组成的一串符号，ANSI C 规定标识符长度不得大于 32 个字符，而 PC 机中通常是前 8 个字符有效。

C 语言规定，用户选取的标识符不能是 C 语言规定的保留字。

C 语言的词类主要分为下列几种：

- 1) 常量。在程序运行中其值不发生变化的数据。
- 2) 变量。用来存放程序运行中变化的数据。
- 3) 运算符。用来表示简单加工计算的符号。
- 4) 函数调用。形如“函数名(实际参数表)”的式子，它代表调用指定函数后获得的结果。



5) 表达式。用常量、变量、函数调用、运算符组成的式子,用来表示简单的加工计算。

6) 保留字。在程序或语句中,用来表示特定语法含义的英语单词。

4. C 语言的基本语句

C 语言的语句主要分为以下几种:

1) 数据定义语句。用来定义程序中使用的各种能存放数据的对象的名称和特性。

2) 赋值语句。形如“变量=表达式”的语句,功能是计算表达式的值并赋予变量。

3) 函数调用语句。形如“函数名(实际参数表)”的语句,功能是调用指定函数。

4) 表达式语句。由任何表达式组成的语句。在 C 语言中,赋值和函数调用都是表达式,所以赋值语句和函数调用语句也是一种特殊的表达式语句。

5) 流程控制语句。用来控制程序执行过程的语句,如选择控制语句、循环控制语句、中止语句、继续循环语句、返回语句、无条件转移语句等。

6) 复合语句。用花括号括住的若干个任意语句。

7) 空语句。无任何操作的语句。

8) 其他语句。包括编译预处理命令、类型定义语句等。

5. C 程序的基本结构

C 程序是 C 语句组成的语句序列,通常一个 C 程序包括一个或多个函数,其中必定有一个函数称为主函数,其函数名为 main。

1) C 程序是由若干个函数构成的。每个 C 程序有且仅有一个主函数,该主函数的函数名规定为 main。除主函数外,可以没有其他函数,也可以有一个或多个其他函数。

2) 每个函数(包括主函数)的定义分为两个部分:函数说明部分和函数体。

函数说明部分的格式如下:

返回值的类型 函数名(形式参数 1,形式参数 2,...)

形式参数的说明:

形参类型 形参

函数体的格式如下:

```
{变量定义部分  
实现函数功能的语句串  
}
```

3) C 程序中的语句最后总要有个“分号(;)",作为每个语句的结束。

4) 可以用“/*”和“*/”括住任意字符,称为“注释”。注释可以出现在程序的任何位置上,通常放在一段程序的开始,用于说明该段程序的功能;或者放在某个语句的后面,对该语句进行说明。程序编译时,注释将不参加编译,也不会出现在目标程序中。

5) C 程序的书写格式很灵活,在一行上可以书写多个语句,一个语句也可以写在多行上。在程序清单的任何一处都可以插入空格符号或回车符号。但是,为了程序清单层次分明、便于阅读,通常都采用缩格并对齐的书写方法。

6) C 程序的执行总是从主函数开始,并在主函数中结束。主函数的位置是任意的,可以在程序的开头,可以在程序的结尾,也可以在两个函数之间。其他函数总是通过函数调用语

句来执行。

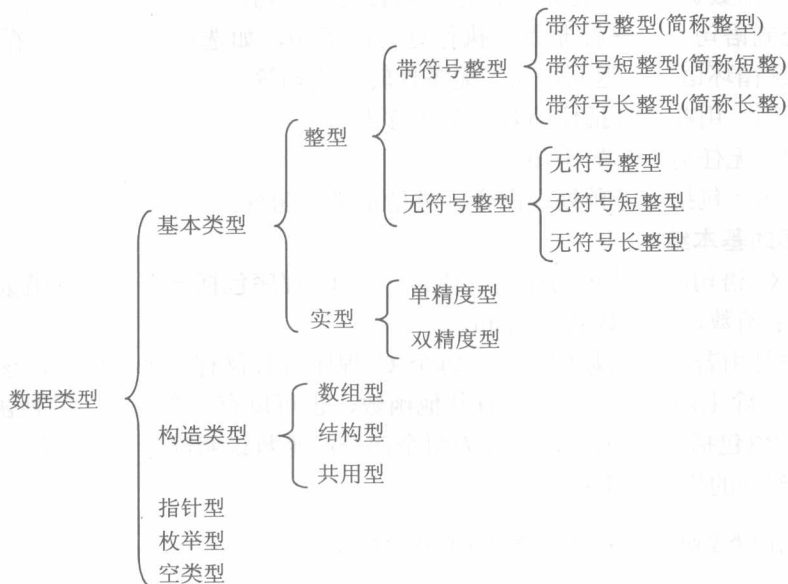
7) 主函数可以调用任何非主函数；任何非主函数都可以相互调用，但是不能调用主函数。

1.2.2 基本数据类型和运算

1. C 语言的数据类型

数据类型是指数据的内在表现形式。通俗地说，数据在加工计算中的特征就是数据的类型。

C 语言中数据类型的分类如下：



2. 常量

常量又叫常数，它是程序运行过程中其值不改变的数据。常量也有数据类型。C 语言规定常量的类型有四种：整型常量、实型常量、字符常量、字符串常量。常量是不需要事先定义的，只要在程序中需要的地方直接写出该常量即可。常量的类型也不需要事先说明，它们的类型是由书写方法自动默认的。

整型常量就是整数，包括正整数和负整数及 0，其数据类型显然是整型。整型常量（短整型常量）在一般微机中占用 2 个字节，长整型常量在计算机中占用 4 个字节。在 C 语言中，整型常量有三种形式的书写方法：

- 1) 十进制整数。通常整数的写法。
- 2) 八进制整数。书写方法是在通常八进制整数的前面加一个数字 0。
- 3) 十六进制整数。书写方法是在通常十六进制整数的前面加 0x。

实型常量也称为“浮点数”，也就是带小数点的实数。实型常量只使用十进制数，它的书写方法有两种：

- 1) 一般形式的实数。它是由整数、小数点、小数三部分组成，其中整数部分或小数部分可以省略。数的正负用前面的“+”（可以省略）号或“-”号来区分。

2) 指数形式的实数。它是由尾数、小写字母 e 或大写字母 E、指数三部分组成。尾数部分可以是十进制整型常量或一般形式的实数，指数部分是十进制的短整型常量。

字符常量是用两个单引号(')前后括住的单个字符来表示。C 语言规定，字母是区分大小写的。由于整型常量在内存中存放的是整数值，如果其值在 0~128 之间，C 语言规定也可以将其看成一个字符常量，对应的字符就是该值作为 ASCII 代码值所对应的那个字符。

字符串常量简称为“字符串”。字符串就是用两个双引号(")前后括住的若干个字符。C 语言规定，字符串中的字母是区分大小写的。一个字符串中所有字符的个数称为该字符串的长度，其中每个转义字符只当做一个字符。虽然在内存中每个字符只占用 1 个字节，但 C 语言规定，每个字符串在内存中占用的字节数等于字符串的长度加 1。其中最后一个字节存放的字符称为“空字符”，其值为 0，书写时常用转义字符“\0”来表示，在 C 语言中称为字符串结束标记。

符号常量是常量的另一种书写方法。为了在程序中明确看出某些常量所表述的对象，可以用特定的标识符来表示这些常量。通常是在程序的开头先定义所有的符号常量，程序中凡是使用这些常量的地方都可以写成对应的标识符。

符号常量的定义格式如下：

```
#define 符号常量 常量
```

其中，符号常量可以按标识符的规则构成，为了和后面将介绍的变量名相区别，建议用大写英文字母；常量可以是任何类型的。该定义必须放在程序的开头，每个定义必须独占一行，其后不跟逗号。

3. 变量

变量是指在程序运行过程中其值可以发生变化的量。

每个变量在使用前都必须定义，定义的内容之一就是说明其数据类型。

定义变量数据类型的语句格式如下：

```
数据类型符 变量名 1,变量名 2,...;
```

变量的初始化是给变量赋值的一种方式。在定义变量的同时给变量赋予初始值就称为变量的初始化。变量的初始化方法很简单，只要在定义变量的语句中，在变量名后跟一个等号和初值即可。初值可以是常量，或由常量组成的表达式。

变量赋初值的语句格式如下：

```
存储类型符 数据类型符 变量名 1=初值 1,变量名 2=初值 2,...;
```

我们知道，在计算机的寄存器和内存中都可以存放数据，而内存中又可以分为一般数据区和堆栈区。我们把变量存放在何处称为变量的存储类型。用户可以通过说明变量的存储类型来选择变量的具体存储地点。

定义变量存储类型的语句格式如下：

```
存储类型符 数据类型符 变量名 1,变量名 2,...;
```

变量的存储类型共分为四种：自动型，寄存器型，静态型，外部参照型。

(1) 自动型。

自动型又称堆栈型。自动型变量是分配在内存的堆栈区，堆栈区内存在程序运行中是重复使用的。当某个函数中定义了自动型变量，C 语言就在堆栈区给该变量分配字节用于存放变量的值。当退出该函数时，C 语言就释放该变量，即从堆栈区中收回分配给该变量的字节，以便重新分配给其他自动型变量。这样做的目的是节省内存。用户对于只在某函数中使用的变量应该将其定义成自动型变量，以便充分利用内存。定义某个或某些变量时，若省略了存储类型符，系统将默认为自动型变量。

(2) 寄存器型。

寄存器型变量是分配在 CPU 的通用寄存器中的。由于 CPU 具有的通用寄存器数量有限，C 程序中允许定义的寄存器型变量一般以 2 个左右为宜。如果定义为寄存器型变量的数目超过所提供的寄存器数目，编译系统自动将超出的变量设为自动型变量。对于占用字节数多的变量，如 long、float、double 类型的变量不能定义为寄存器型变量。

寄存器型变量一般是在函数中定义的，退出该函数后就释放它所占用的寄存器。

(3) 静态型。

静态型变量是分配在内存的数据区中，它们在程序开始运行时就分配了固定的字节，在程序运行过程中不释放。只有程序运行结束后，才释放所占用的内存。

(4) 外部参照型。

C 语言允许将一个源程序清单分放在若干个程序文件中，采用分块编译方法编译生成一个目标程序。其中每个程序文件称为一个“编译单位”。

外部参照型变量是专用于多个编译单位之间传递数据用的。当编译单位甲中要使用在编译单位乙中定义的变量，则编译单位甲就要说明该变量是外部参照型，以便 C 语言编译系统在编译单位甲之外的其他编译单位中寻找该变量的定义。将某个变量说明为其他编译单位定义的变量的方法就是将它说明为“外部参照型”。被说明成外部参照型的变量的存储类型和数据类型都由定义该变量的那个编译单位来说明。

我们把函数（或某个复合语句）内定义的变量称为“内部变量”；把在函数外定义的变量称为“外部变量”。

从变量的生存期来分，我们把生存期覆盖了从定义点到整个程序结束的变量称为具有全局寿命的变量，简称“全局变量”；把生存期只覆盖某个函数（或某个复合语句）的变量称为具有局部寿命的变量，简称“局部变量”。

任何变量的生存期（指全局变量或局部变量）和作用域既取决于它是外部变量还是内部变量，也取决于该变量是何种存储类型。表 1-1 给出了变量的生存期和作用域。

表 1-1 变量的生存期和作用域

变 量	允许的存储类型	生 存 期	作 用 域
外部变量	省略（称无存储类型）	全局变量	定义点到程序结束
	静态型（static）		
内部变量	省略（默认为自动型 auto）	局部变量	定义该变量的函数或复合语句内部
	自动型（auto）		
	寄存器型（register）		
	静态型（static）	全局变量	

4. 运算符

算术运算符包括基本算术运算符和增 1、减 1 运算符，其中基本运算符常简称为算术运算符。

基本算术运算符的运算对象、运算规则与结果、结合性如表 1-2 所示。

表 1-2 基本算术运算符

对象数	名称	运算符	运算规则	运算对象	运算结果	结合性		
单目	正	+	取原值	整型 或 实型	整型 或 实型	自右向左		
	负	-	取负值					
双目	加	+	加法			整型	整型	自左向右
	减	-	减法					
	乘	*	乘法					
	除	/	除法					
	模	%	整数取余					

算术运算符的优先级规定如下：

- 1) 单目基本算术运算符优先于双目基本算术运算符。
- 2) *、/、% 优先于+、-。
- 3) 同级单目基本算术运算符的结合性是自右向左。
- 4) 同级双目基本算术运算符的结合性是自左向右。

增 1 减 1 运算符的运算对象、运算规则与结果、结合性如表 1-3 所示。

表 1-3 增 1 减 1 运算符

对象数	名称	运算符	运算规则	运算对象	运算结果	结合性
单目	增 1(前缀)	++	先加 1，后使用	整型、字符型、指针型变量或数组元素	同运算对象的类型	自右向左
	增 1(后缀)	++	先使用，后加 1			
	减 1(前缀)	--	先减 1，后使用			
	减 1(后缀)	--	先使用，后减 1			

增 1 减 1 运算符的优先级如下：

- 1) 增 1 减 1 运算符优先于双目基本算术运算符。
- 2) 增 1 减 1 运算符和单目基本算术运算符+、-同级别，结合性是自右向左。

关系运算符的运算对象、运算规则与结果、结合性如表 1-4 所示。

表 1-4 关系运算符

对象数	名称	运算符	运算规则	运算对象	运算结果	结合性
双目	小于	<	满足则为真，结果为 1；不满足则为假，结果为 0	整型或实型或字符型等	逻辑值(整型)	自左向右
	小于或等于	<=				
	大于	>				
	大于或等于	>=				
	等于	==				
	不等于	!=				