C语言程序设计

吴晓峰
信息科学与工程学院

教学人员

任课教师：吴晓峰
办公室：物理楼218B
Email： xiaofengwu@fudan.edu.cn
电话： 65643757(0)

助教： 顾怡炜
联系方式：物理楼603室
Email： ywgu15@fudan.edu.cn
电话： 18817301078

课件访问：elearning.fudan.edu.cn

授课方式(3＋2课时)

♦课堂讲授(3课时)
时间：周四下午6～8节
地点： HGX303
课堂纪律：参照复旦大学教务处制定的有关条例
♦上机
时间： 周五下午8～9节，
地点： 计算中心三楼2号机房 (2课时)
♦作业 每周布置一次，次周上课前交作业
(过时不候！)
♦考评 70%期末考试+30%平时(作业、上机等)

课程性质：专业基础课(必修课)

♦课程介绍－－教学大纲

《程序设计》是信息技术类专业本科生的一门必修课程，本课程以C语言为程序的描述语言。学习本课程并不要求大家能用C设计一个很复杂的东西，而是要求学生掌握结构化设计的编程思想，对编程不再恐惧陌生。通过本课程的学习，了解程序设计的基本原理、技巧和方法。结合上机实践，让同学具有独立构造算法、开发程序以及调试程序能力。学完本课程后对后继语言的学习将会有很大的帮助！

教学内容：

1、程序设计基础
2、基本数据及其运算和输入输出
3、结构化程序开发
4、数组、字符串、指针及其应用
5、函数
6、存储类型和编译预处理
/ 作用域规则和编译预处理命令
7、结构和链表
8、数据文件处理技术

通过本课程，学习编制“结构合理、风格良好”的程序。为学习与计算机相关课程，打下坚实的基础。

教学目标：

♦掌握程序设计语言的基本知识
♦ 熟悉C语言的上机操作环境
♦结合上机实践，进一步帮助学生提高独立构造算法、开发程序以及调试程序的能力

学习方法

- ♦自主学习，c单词一定要记住
- ♦一定要理解程序，重视上机实践

上机要求

- ♦禁做与课程学习无关的事情（如上网聊天、游戏等）
- ♦认真完成上机实践课题。

8

教学参考资料（含教材）

- ♦夏宽理、王春森编著，《程序设计》
  复旦大学出版社
- ♦谭浩强《C程序设计》清华大学出版社
- ♦其他同类书籍

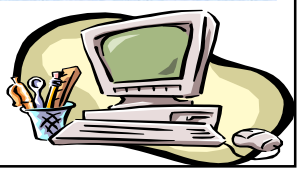
9

学习程序设计是一件很辛苦的事情，要有足够的耐心和细心。学习方法：

理解→ 实践→理解→实践→.....

掌握多种编程语言，

将会照亮您的一生。

课程信息

一年级	程序设计
二年级	数据结构和算法设计、Web应用基础
三年级	计算机体系结构、微机原理、嵌入式系统、数字信号处理
四年级	

第1章 程序设计基础

●要求：

1）了解C语言的发展过程与特点；
2）掌握C语言程序的结构和书写格式；
3）熟悉C语言程序的上机调试过程。

12

1.1 程序设计基础

1.1.1 计算机入门（自学p1~2）

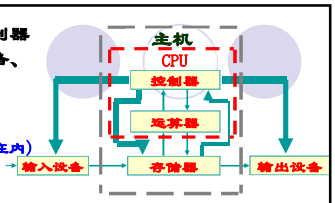
计算机是能对用离散符号表示的数据或信息可编程，自动进行处理的电子装置。

♦Hardware（硬件）

计算机系统由硬件和软件两大部分组成. 硬件(Hardware，又可称：计算机硬件，计算机硬件系统)，计算机系统中实际物理装置的总称。例如： 计算机主板，CPU和内存，外存(硬盘，光盘，U盘等)，键盘和鼠标，显像管显示器，液晶显示器，输入光笔，打印机，扫描仪，数码照相机等等.

13

硬件系统包括五大部件：控制器、运算器、存储器、输入设备、输出设备。



- 控制器(Controller)
  控制计算机各部件(包括CPU在内)高效地协调工作。
- 运算器(Operator)
  对数据进行算术和逻辑运算。控制器和运算器合称为中央处理器(CPU：Central ProcessUnit)。
- 存储器(Storage)
  计算机内最主要的记忆装置。用于保存程序和数据信息。
- 输入设备(Input Device)
  输入设备(Input Device)指具有向计算机系统输入信息功能的设备。常用的输入设备有：键盘、鼠标器、扫描仪、光笔、条形码读入器等。
- 输出设备(Output Device)
  指能从计算机系统中送出信息的设备。常用的输出设备有：显示器、打印机、绘图仪等。

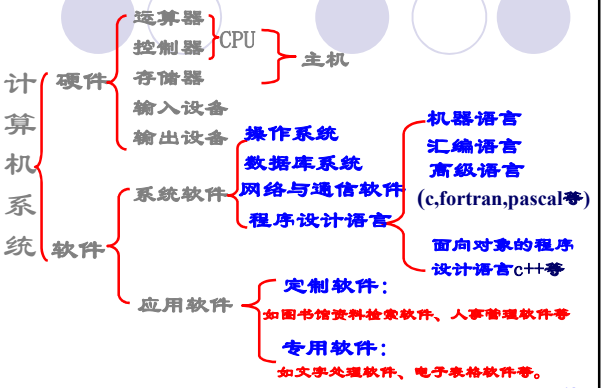
14

♦ Software（软件）

计算机软件是所有计算机程序和相关文件的总称．它由系统软件(如：操作系统，计算机语言，数据库系统.....)和应用软件(如：工具软件，杀毒软件，用户应用软件)构成．如果没有软件，计算机是一台“裸机”，是什么也不能干的，有了软件，才能灵动起来，成为一台真正的“电脑”。计算机其实由软件和硬件构成，硬件是可以看到的，是物质基础，软件则是它的思想灵魂。所有的软件，都是用计算机语言编写的。

15

计算机系统



16

1.2 C语言的发展与特点

在C语言诞生以前，系统软件主要是用汇编语言编写的。由于汇编语言程序依赖于计算机硬件，其可读性和可移植性都很差；但一般的高级语言又难以实现对计算机硬件的直接操作（这正是汇编语言的优势），于是人们盼望有一种兼有汇编语言和高级语言特性的新语言。

一、C语言的发展历史

C是当今最流行的程序设计语言之一，它的产生过程

◆1963 年，剑桥大学将Algol 60 语言(也称为A 语言，C 语言的原型)发展成为CPL 语言，并随后将CPL 发展为BCPL 语言。

◆1970 年，贝尔实验室的汤普森将BCPL发展为B 语言,主要用于UNIX操作系统。

肯尼思·汤普森
Kenneth Lane Thompson

18

◆1973 年，贝尔实验室的里奇基于B 语言设计出C 语言。

◆1978 年，贝尔实验室正式发表了C 语言。由柯宁汉和里奇合著了著名“The C Programming Language”。

丹尼斯·里奇
Dennis MacAlistair Ritchie

布赖恩·威尔森·柯宁汉
Brian Wilson Kernighan

◆1983 年，美国国家标准协会(American National Standards Institute) 制定了一个C 语言标准，通常称之为ANSI C。

二、C语言的特点

- 语言简洁、紧凑、灵活（32个关键字，9种控制语句）
- 运算符丰富（34种表达式）
- 数据类型丰富
- 程序设计结构化、模块化
- 生成目标代码质量高
- 可移植性好（较之汇编语言）。
- 可以直接操纵硬件。

1.3 程序设计基本概念（了解几个基本概念）

♦程序

计算机革命起源于一台机器，程序设计语言也源于一台机器。然而计算机并不仅仅是一台机器，它是心智放大器和一种有表述能力的媒体。我们可以利用计算机开发出许多激动人心的工作。

因此，要使计算机能完成人们预定的工作，就必须把要解决的问题编成**计算机能够接收并能执行**的一条条指令，既编排成一系列计算机解题的步骤．计算机执行这个指令序列后，就能完成指定的功能，这样的指令序列就是**程序**。简而之，**程序是指令的序列。**

例一：

从自然语言的角度，程序是对解决某个问题的方法步骤的描述，从计算机的角度，程序是用某种计算机能理解并能执行的计算机语言描述解决问题的方法步骤。

程序的特点是有始有终，每个步骤都能操作，所有步骤执完对应问题就能得到解决。

例如：某个会议的安排如下：

第一项　　宣布会议开始。
第二项　　全体起立唱国歌。
第三项　　宣读嘉奖令。
第四项　　颁发奖励证书。
第五项　　宣布会议结束。

上述步骤是解决了嘉奖问题的程序。

例二：

求一元二次方程$ax^2+bx+c=0(a\neq0)$

设解实数根的步骤如下：

第1步 获得系数a, b, c

第2步 计算$d=b^2-4ac$

第3步 若d>0，计算$x1=(-b+\sqrt{d})/2a, x2=(-b-\sqrt{d})/2a$
输出：有两个实根x1和x2，转**第六步**
否则到**第4步**。

第4步 若d<0，则输出：没有实根，转到**第6步**
否则转到**第5步**。

第5步 计算：x1=x2=-b/2a
输出：有两个相同的实根为x1，转到第6步。

第6步 结束。

程序的性质(p3)

- 目的性：程序有明确的目的，程序运行时能完成赋予它的功能。
- 分步性：程序为完成其复杂的功能，由一系列计算机可执行的步骤组成。
- 有序性：程序的执行步骤有序的，不可随意改变程序步骤的执行顺序。
- 有限性：程序所包含的指令序列是有限的。
- 操作性：可以对某些对象进行操作，完成程序预定的功能。

♦ 程序设计

程序设计就是分析解决问题的方法步骤，并将其记录下来的过程。从自然语言的角度，就是用自然语言的角度来记录，从计算机角度，就是用计算机语言记录下来。简而之，程序设计是：**编写程序的过程。**

上面的例一用自然语言的角度来记录的程序，而例二用计算机语言来记录，例如用c语言来记录，结果如下：

```
#include <stdio.h>
#include <math.h>
void main()
{     float a,b,c,d,x1,x2;          //说明存放数据的变量
      scanf("%f%f%f",&a,&b,&c); //输入三个系数
      d=b*b-4*a*c;       //计算d
      if(d>0)            //若d>0，有两个实根
      {   x1=(-b+sqrt(d))/(2*a);
          x2=(-b-sqrt(d))/(2*a);
          printf("x1=%f,x2=%f\n",x1,x2);//输出两个实根
      }
      else if(d==0)
      {     x1=x2=-b/2*a;
            printf("x1=x2=%f\n",x1); //输出重根
      }
      else printf("没有实根\n"); // d<0
}
```


♦程序设计语言(计算机语言)

人和计算机进行沟通的语言－人用计算机语言编制程序，计算机执行程序。而c程序设计就是通过c语言与计算机通信，并告诉计算机如何工作。

1946年，第一台电子计算机问世，应用领域迅速扩大，软硬件飞速发展，程序设计语言相继问世。计算机发展到今天，程序设计语言多达上千种，它们大致可分为三类：

机器语言
汇编语言　} 低级程序设计语言
高级语言

1.机器语言：

电子计算机所使用的是由“0”和“1”组成的二进制数，二进制是计算机的语言的基础。由“0”和“1”组成的指令序列交由计算机执行，这种语言，就是机器语言。

如：10000000　加
　　10010000　减

优点：运行效率高：占用内存少、执行速度快。

缺点：不直观，不易阅读和记忆，不易查错，与硬件相关，移植性不好，程序调试和维护比较困难。

2. 汇编语言：

为了减轻使用机器语言编程的痛苦，人们进行了一种有益的改进：用一些简洁的英文字母、符号串来替代一个特定的指令的二进制串，比如，用“ADD”代表加法，“SUB”代表减法，这样一来，人们很容易读懂并理解程序在干什么，纠错及维护都变得方便了，这种程序设计语言就称为汇编语言，即第二代计算机语言。然而计算机是不认识这些符号的，这就需要一个专门的程序，专门负责将这些符号翻译成二进制数的机器语言，这种翻译程序被称为汇编程序。

例如：
C=A+B;
用汇编语言写成：

```
LD   RG0, A      /*将A放到寄存器：RG0*/
LD   RG1, B      /*将B放到寄存器：RG1*/
ADD RG0, RG1  /* RG0和RG1相加，结果放RG0 */
ST   RG0, C      /*将RG0 送C */
```

LD：　取数指令，取内存A的数给寄存器
ADD：加法指令，两个寄存器相加后，结果给目标寄存器
ST：　存数指令，寄存器中的数给内存B

优点：运行效率仍十分高，执行速度较快。
　　程序精炼而质量高。

缺点：与硬件相关，移植性不好。

3.高级语言

以更接近于数学语言或人们的自然语言，同时又不依赖于计算机硬件，编出的程序能在所有机器上通用。1954年，第一个完全脱离机器硬件的高级语言—FORTRA N问世，影响较大、使用较普遍的有FORTRAN、BASIC、C、Ada、C++、VC、VB、Delphi、JAVA等。

高级语言主要由语句构成，有一定书写规则，程序员用语句表达要计算机完成的操作。如：C=A+B;

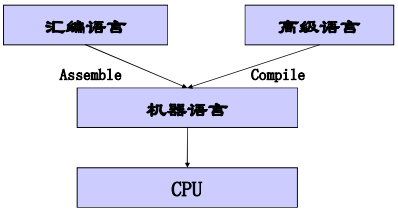
优点：编程效率高，不必考虑硬件，易于编写、理解和维护。

缺点：执行效率低，占用内存多，运行速度较慢。

程序设计语言：将自然语言形式化为有格式的语言.

从前面得知:只有用计算机指令编写的程序能够直接被计算机执行，而其他的指令还需要通过中间的翻译过程。把用高级语言编写的源程序转换成机器语言程序的翻译程序称为

编译器 或 编译程序



32

为什么要学习c语言？

C语言是中级语言，它即有高级语言的特点，有很强的数据处理能力，同时有汇编语言的功能，符合硬件处理要求。C允许对位、字节和地址这些计算机功能中的基本成分进行操作。C语言程序非常容易移植。因此，C语言常被用于系统程序设计软件，也可以作为应用程序设计语言，编写不依赖计算机硬件的应用程序。例如，可编写多维图形和动画软件。

C语言不仅在速度和结构上有它的优势，而且每个C语言系统都提供了专门的函数库，程序员可以根据不同需要对其进行剪裁，以适应各种程序的设计。由于它允许分别编译，所以C语言可使程序员方便地管理大型项目，最大限度地减少重复劳动。

简单地讲，算法（Algorithm）是为解决一个特定问题而采取的确定的、有限的方法和步骤。例如，某个会议的具体安排就是完成“开会”这个问题的算法，求解一元二次方程的求根公式是解决一元二次方程求根的算法。这些具体的方法和步骤，就是解决一个问题的算法。如果一个算法有缺陷，或不适合于某个问题，执行这个算法将不会解决这个问题。不同的算法可能用不同的时间、空间或效率来完成同样的任务。一个算法的优劣可以用空间复杂度与时间复杂度来衡量。

34

♦算法

什么是程序？程序=数据结构+算法。

对于面向对象程序设计，强调的是数据结构，而对于面向过程的程序设计语言如C、Pascal、FOTRAN等语言，主要关注的是算法。掌握算法，也是为面向对象程序设计打下一个扎实的基础。

那么，什么是算法呢？人们使用计算机，就是要利用计算机处理各种不同的问题，而要做到这一点，人们就必须事先对各类问题进行分析，确定解决问题的具体方法和步骤，再编制好一组让计算机执行的指令即程序，交给计算机，让计算机按人们指定的步骤有效地工作。这些具体的方法和步骤，其实就是解决一个问题的算法。根据算法，依据某种规则编写计算机执行的命令序列，就是编制程序，而书写时所应遵守的规则，即为某种语言的语法。

由此可见，程序设计的关键之一，是解题的方法与步骤，是算法。学习高级语言的重点，就是掌握分析问题、解决问题的方法，锻炼分析、分解，最终归纳整理出算法的能力。

所以在高级语言的学习中，一方面应熟练掌握该语言的语法，因为它是算法实现的基础，另一方面必须认识到算法的重要性，加强思维训练，以写出高质量的程序。

下面通过例子来介绍如何设计一个算法：

36

算法分析：例1、求1+2+3+…+100的和。

方法一：第一步：对称地取前后两数相加，即1+100，2+99，3+98，…，50+51，得出两数之和都是101的规律。

第二步：这样的和式共有 $\frac{100}{2}$ 个，即50个

第三步：总结出计算方法：$101\times\frac{100}{2}$;

第四步：计算上式得出结果5050。

注意：算法有始有终。

方法二：直接累加求1+2+3+ …+100的和，虽然也能求得结果，但是要做99次加法，这显然是十分烦琐的。

可见，解决同一问题可能有不同的算法。通过分析可以看出，一个好的算法可以简捷、高效地解决问题。

37

例2、

猴子吃桃问题：有一堆桃子不知总数，猴子第一天吃掉一半，觉得不过瘾，又多吃了一只，第二天照此办理，吃掉剩下桃子的一半另加一个，天天如此，到第十天早上，猴子发现只剩一只桃子了，问这堆桃子原来有多少个？

假设第一天开始时有a_1只桃子，第二天有a_2只，. . .，第9天有a_9只，第10天是a_{10}只，在a_1，a_2，. . .，a_{10}中，只有a_{10}= 1是知道的，现要求a_1，而我们可以看出，a_1，a_2，. . .，a_{10}之间存在一个简单的关系：

$a_9 = 2 * (a_{10} + 1)$

$a_8 = 2 * (a_9 + 1)$

⋮

$a_1 = 2 * (a_2 + 1)$

也就是：$a_i = 2 * (a_{i+1} + 1)$ i=9, 8, 7, 6, . . . , 1

这就是此题的数学模型。

再考察上面从a_9，a_8直至a_1的计算过程，这其实是一个递推过程，这种递推的方法在计算机解题中经常用到。另一方面，这九步运算从形式上完全一样，不同的只是a_i的下标而已。由此，我们引入循环的处理方法，并统一用a_0表示前一天的桃子数，a_1表示后一天的桃子数，将算法改写如下：

1) a_1=1；{第10天的桃子数，a1的初值}
   i = 9。{计数器初值为9}
2) $a_0 = 2 * (a_1 + 1)$。{计算当天的桃子数}
3) a_1= a0。{将当天的桃子数作为下一次计算的初值}
4) i=i-1。
5) 若i > = 1，转2)。
6) 输出a_0的值。

其中（2 ~ 5 ）步为循环。

这就是一个从具体到抽象的过程，具体方法是：

1) 弄清如果由人来做，应该采取哪些步骤。
2) 对这些步骤进行归纳整理，抽象出数学模型。
3) 对其中的重复步骤，通过使用相同变量等方式求得形式的统一，然后简练地用循环解决。

♦算法的描述

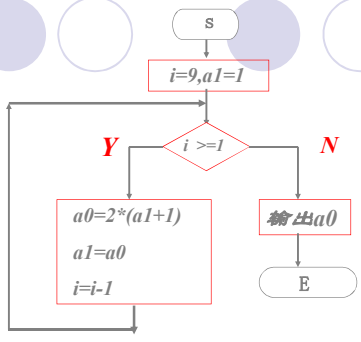
算法可以使用自然语言、伪代码、流程图等多种不同的方法来描述。

1. 流程图：

流程图是一种传统的算法表示法，它利用几何图形的框来代表各种不同性质的操作，用流程线来指示算法的执行方向。

流程图的元素包括：

- 程序的入口（开始，S，Start）和出口（结束，E，End）。用圆角框表示。
- 程序中的处理，用以描述程序中执行某项工作。用方框表示
- 程序中的控制流，用以描述处理的先后顺序。用带箭头的线段表示。
- 程序执行的条件。通常用菱形框表示。



猴子吃桃算法流程图

2. 伪代码

例如，图1.1的算法用伪代码可描述如下：

```
s = 0; k = 1;
do {
    输入x;
    s = s+x;
    k = k+1;
  } while(k < 11);
```

♦数据结构

程序的处理对象是描述客观事物的数据，由于客观事物的多样性，会有不同形式的数据，如整数、实数、字符，以及计算机能够接收、存储和处理的各种各样符号集合。

数据结构是指ADT（抽象数据类型 Abstract Data Type）的物理存放方式的实现。

- ♦基本类型（整型、实型、字符型）
- ♦构造类型（数组、指针、结构）
- ♦复杂的数据结构（栈、队列、树、图）

有些书直接将程序定义为：

程序 = 数据结构 + 算法

44

♦结构化程序设计和结构化控制结构

结构化程序设计由迪克斯特拉(E.W.dijkstra)于1969年提出，是以模块化设计为中心，将待开发的软件系统划分为若干个相互独立的模块，这样使完成每一个模块的工作变单纯而明确，为设计一些较大的软件打下了良好的基础。

由于模块相互独立，因此在设计其中一个模块时，不会受到其它模块的牵连，因而可将原来较为复杂的问题化简为一系列简单模块的设计。模块的独立性还为扩充已有的系统、建立新系统带来了不少的方便，因为我们可以充分利用现有的模块作积木式的扩展。

45

结构化程序设计的基本思想是采用“自顶向下，逐步求精”的程序设计方法和“单入口单出口”的控制结构。

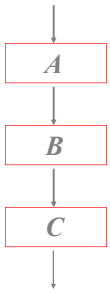
"自顶向下、逐步求精"的程序设计方法是从问题本身开始，经过逐步细化，将解决问题的步骤分解为由基本程序结构模块组成的结构化程序框图；

经过多年的研究，按照结构化程序设计的观点，任何复杂的程序，都可以由顺序结构、选择（分支）结构和循环结构这三种控制结构作为“建筑单元”，基本结构之间可以并列、可以相互嵌套，但不允许交叉，那么这个新构造的程序一定是一个单入口单出口的程序。据此就很容易编写出结构良好、易于调试的程序来。

46

(1) 顺序结构

顺序结构是简单的线性结构，各框按顺序执行。其流程图的基本形态如图所示，语句的执行顺序为：A→B→C。



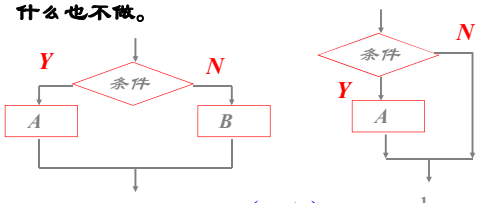
47

(2) 选择（分支）结构

这种结构是对某个给定条件进行判断，条件为真或假时分别执行不同的框的内容。其基本形状有两种，如图 a）、b）所示。

图a)的执行序列为：当条件为真时执行A，否则执行B；

图b)的执行序列为：当条件为真时执行A，否则什么也不做。



a 选择（分支）结构 b

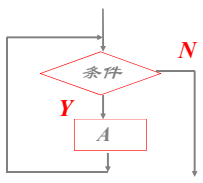
48

(3) 循环结构

循环结构有两种基本形态：while型循环 和do－while型循环。

a. while 型循环

其执行序列为：当条件为真时，反复执行A，一旦条件为假，跳出循环，执行循环紧后的语句。

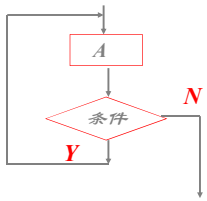


while 型循环

49

b. do-while型循环

执行序列为：首先执行A，再判断条件，条件为真时，一直循环执行A，一旦条件为假，结束循环，执行循环紧后的下一条语句。



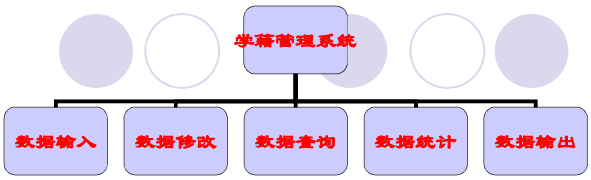
do-while型循环

复杂问题的解决方法

下面我们举一实例，加以具体说明。

例如：学籍管理系统。

就整个系统来看，我们很难马上就写出解决问题的算法及对应的源程序，因为这个系统相对来说比较大、比较复杂，我们不妨把整个系统分解成若干个小问题，减小问题的规模和复杂程度。经过系统分析，整个系统大致包括数据输入、数据修改、数据查询、数据统计和数据输出几个部分。根据不同问题的划分，每一类问题作为一个模块，可以画出整个学籍管理系统的一级模块图，如下图所示。



学籍管理系统的一级模块图

此时的模块图比起最初的感觉要直观一些，每一个子模块要比整个系统要简单一些。这时还可以把一级模块图中某一个子模块再进一步划分。比如：数据输入可分解成学生基本情况输入模块、学生成绩输入模块、学生奖惩情况输入模块等，同样其他一级子模块也可继续细分，这里就不再赘述了。

通过对该例解决问题思路的描述，我们对“自上而下，逐步细化，模块化”的程序设计思想有了进一步的了解。希望读者在今后的程序设计中，不断地学习，不断地实践，逐步掌握这种设计方法。

1.4 程序语言的结构和书写格式

利用计算机解决各种类型，复杂程度各异的问题时，关键是用户需要编写出计算机能够“读懂”的程序，使计算机能够按照程序设计者的意愿去工作。C语言就是一种在计算机上实现程序的描述语言。

1.4.1 几个简单C的程序

任何一种程序设计语言都具有特定的语法规则和规定的表达方法。类似于我们说话、写文章要有主谓宾基本语句部分一样，缺少一部分就不成一句话了。

为了解C语言的基本程序结构，我们先介绍几个简单的C程序。

【例1.1】

```
/*1*/ #include <stdi
/*2*/ void main()
/*3*/{
/*4*/ printf("Hellow World!\n");
/*5*/}
```

编译预处理命令

main()是主函数，每个C程序必须有一个main主函数。main是函数名，不能更改；“()”表示这是一个函数，括号为空表示函数没有参数。void表示函数没有返回值。

表示语句结束

程序的注释部分

用是在屏幕上输出信息

Hellow World!"{"开始，"}"结束。

函数体由一对大括号{}的范围以内的部分组成。本例中主函数内只有一条输出语句。

函数，""内的字符串按原样输出，“\n”是换行符，即输出完后回车换行。

【例1.2】读入两个整数，输出它们的和

```
#include  <stdio.h>
 void main()
 {  /* 变量定义部分 */
 int x, y, sum; /* 定义 x, y, sum */
  /* 以下为语句序列 */
 printf("Input x and y\n");/*提示输入数据*/
 scanf("%d%d", &x, &y); /* 输入x和y的值 */
 sum = x+y ; /* 完成x+y的计算,求sum=x+y */
 printf("x + y = %d\n", sum); /* 输出结果 */
}
```


【例1.3】已知华氏温度0、20、…、200，求对应的摄氏温度。

```
#include  <stdio.h>
void main()
{  double f, c;        /* 变量定义 */
   int lower, upper, step;
   lower = 0;    upper = 200;
   step = 20;    f = lower;
   while (f <= upper) {  /* 循环计算 */
     c = 5.0/9.0 * (f - 32.0);
     printf("\t%7.0f   %7.1f\n", f, c);
     f = f + step;
   }
}
```


【例1.4】输入两个实数，输出它们中的小的数

```
/*功能：由main()函数和1个其它函数min()构成的C语言程序示例*/

#include  <stdio.h>
float min(float a, float b)
{  float temp ;    /* 函数使用的变量的定义 */
   if (a < b) temp = a; else  temp = b;
   return temp ; /* 返回 temp 到调用 min() 函数处 */
}
void main()
{  float x, y, c;        /* 变量定义 */
   printf("输入x和y.\n");
   scanf("%f%f", &x, &y );
   c = min(x, y);       /* 调用函数 min() */
   printf("MIN(%.2f, %.2f)= %.2f\n",x,y,c);
}
```


1.4.2 C程序组成

函数与主函数

函数是C语言程序的基本单位。

程序由一个或多个函数组成,必须有且只能有一个主函数main。例如：前面的三例只有一个主函数main，例4，有两个函数，它们分别是main和min。

程序执行从main开始(位置无关)，在main中结束，其它函数通过调用得以执行。

函数定义

一个函数由函数头和函数体组成。函数头包括函数属性、函数返回值类型、函数名、函数形式参数名，形式参数类型。

函数结构的一般形式：

函数返回值类型 函数名([参数说明表1, ...])
{ 说明和定义部分;
  执行语句序列;
}

可以没有参数说明表，但函数名之后的一对圆括号是必须的。

使用的语法符号约定
[...]：方括号表示可选
... ：省略号表示前面的项可以重复

函数说明

由函数类型（可缺省）、函数名和函数参数表三部分组成，其中函数参数表的格式为：
数据类型形参[,数据类型形参2.....]

【例1.4】中的函数min ()，其函数说明各部分如图所示

函数类型	函数名	函数参数表
float	min	(float a, float b)

函数体

在函数说明部分的下面,是函数头之后用一对花括号括住的部分。函数体用于描述实现函数功能的代码，它又包括：

➢说明和定义部分

说明语句部分由变量定义、自定义类型定义、自定义函数说明、外部变量说明等组成。

➢执行部分

由C语句和控制结构代码组成,用“;”作为语句终止符。

☞注释

/* */为注释，“/*”和“*/”必须成对使用，且“/”和“*”、以及“*”和“/”之间不能有空格，不能嵌套,否则都出错

// 一行注释

不产生编译代码

非法

```
例 /*This is the main/* of example1.1*/  */
```

☞I/O 函数库， stdio.h

例如：前面介绍过的输入输出函数

scanf

printf

☞编译预处理命令

例如：#include <stdio.h>

1.4.3 C程序的书写格式

- 习惯用小写字母，大小写敏感
- 不使用行号
- 可使用空行和空格
- 常用锯齿形书写格式(缩进对齐)

优秀C程序员的编程风格：

- 使用TAB缩进
- { }对齐
- 有足够的注释
- 有合适的空行

```
#include <stdio.h>
void  main( )
{   int a0,a1,i;
    a1=1;
    i = 9;
    while(i>=1)
    {
        a0=2*(a1+1);
        a1=a0;
        i=i-1;
    }
    printf("%d",a0);
}
```

1.5 C语言的词汇、数据类型、常量和变量

1、基本符号：

- 数字10个(0～9)
- 英文字母大、小写各26个(A～Z，a～z)
- 下线字符 "_"
- 特殊符号的字符集 ，主要用来表示运算符。

2、基本词汇

❖常量（常数）：数据

123（整型） 4.56（实型） 'A'（字符型）

❖符号常量：用标识符表示的常量数据

π：PI

ε：EPS

通常用大写字母

❖ 特殊符号：如各种运算符 +, -, *, /, %等

❖关键字(由系统定义,不能重作其它定义,32个)

关键字又称为保留字。它们是C语言中预先规定的具有固定含义的一些单词，如：int表示为整型数据、float表示为单精度实型数据等等。

用户只能按其给定的含义来使用，不能重新定义另作它用。

根据关键字的作用，可分其为数据类型关键字、控制语句关键字、存储类型关键字和其它关键字四类。

（1）数据类型关键字（12个）
char, double, enum, float, int, long, short, signed, struct, union, unsigned, void

（2）控制语句关键字（12个）：
break, case, continue, default, do, else, for, goto, if,return, switch, while

（3）存储类型关键字（4个）：
auto, extern, register, static

（4）其它关键字（4个）：
const, sizeof, typedef, volatile

下面几个虽不属于关键字，但建议把它们看作关键字，不要在程序中随便使用。它们用在C程序的预处理命令行中。

define undef include ifdef
ifndef
endif line elif

❖标识符：

分为系统预定义标识符和用户自定义标识符

（1）系统预定义标识符

这些标识符也是由一些单词所组成，它们的功能和含义是由系统预先定义好的，
如：main代表主函数名、
printf代表输出函数名。
建议用户不要把这些系统预定义标识符另作它用，否则会带来不必要的麻烦。

（2）用户自定义标识符

用户可根据需要自行定义一些标识符，用作为符号名、变量名、数组名、函数名、文件名等等，如例1.2中： x,y sum代表变量名，
其中sum用于存储x,y的和。

用户自定义标识符的命名必须遵守一定的规则。合法的用户自定义标识符应满足以下条件。

标识符命名规则：

1. C语言规定标识符只能由字母、数字和下划线（"_"）三种字符组成，且第一个字符必须为字母或下划线。

例如：
下面列出的是合法的标识符：
John ,A32 , p12_3, B64, a_b_c
下面是不合法的标识符和变量名
￥123, #33, a>b,*p

2. 标识符的长度各个系统不同，建议变量名的长度不要超过8个字符。

3. 标识符不能是关键字

标识符命名的良好习惯——见名知意：

★所谓"见名知意"是指，通过变量名就知道变量值的含义。通常应选择能表示数据含义的英文单词（或缩写）作变量名，或汉语拼音字头作变量名。
如:name/xm（姓名）、sex/xb（性别）、
age/nl（年龄）、salary/gz（工资）。
等，以增加程序的可读性。这是结构化程序的一个特征。

本书在一些简单举例中，为方便起见，仍用单字符的变量名(如a、b、c等)，读者最好不用此方法。

★C语言对大小写字母敏感。

大写字母和小写字母被认为是两个不同的字符(C语言对大小写字母敏感)。因此，sum和SUM，Class和class是两个不同的变量名。一般，变量名用小写字母表示，与人们日常习惯一致，以增加可读性。

★先定义，后使用。

在C语言中，要求对所有用到的变量作强制定义，也就是“先定义，后使用”。

目的是：

1. 每一个变量被指定为一确定类型，在编译时就能为其分配相应的存储单元。如指定a、b为int型编译系统为a和b各分配4个字节，并按整数方式存储数据。

2. 指定每一变量属于一个类型，这就便于在编译时，据此检查该变量所进行的运算是否合法。

例如，整型变量a和b，可以进行求余运算：

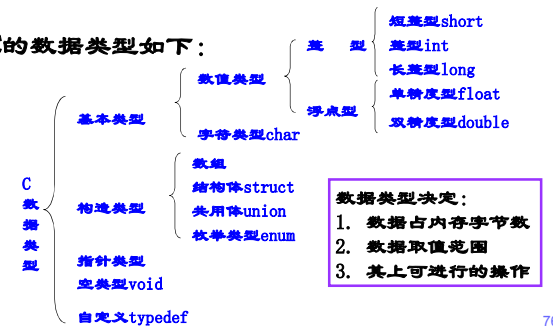
a%b

如果将a、b指定为实型变量，则不允许进行“求余”运算，在编译时会给出有关“出错信息”。

3、C 语言的数据类型

C语言的数据结构是以数据类型形式出现的。主要有基本数据类型、指针类型、构造类型等。

C的数据类型如下：



C语言中数据有常量与变量之分，它们分别属于以上这些类型。

在程序中对用到的所有数据都必须指定其数据类型。

4、常量

★定义：程序运行过程中，其值不能被改变的量称为常量.（即常数）

常量的类型有：

整型常量　如，15、0、－7

浮点型常量　如，5.0、－12.36

字符型常量　如，’A’、’a’

指针常量　如，NULL

字符串常量　如，”ABC”

★符号常量:用标识符代表常量

定义格式: #define 标识符 字符序列

```
#define   PI    3.14159
#define   MAXN  100
```

定义PI代表常量3.14159,此后凡在本文件中出现的PI都代表3.14159，MAXN 都代表100

注意：

★符号常量不同于变量，它的值在其作用域(在本例中为主函数)内不能改变，也不能再被赋值。如再用以下赋值语句给PRICE赋值是错误的。 PRICE=40；

★习惯上，符号常量名用大写，变量名用小写，以示区别。

80

5、变量

在程序运行过程中，其值可以改变的量称为变量。它用标识符来表示(变量名)，

变量在存在期间，在内存中占据一定的存储单元，该存储空间中存放的数据就是变量的值。在程序中，通过变量名来引用变量的值。

每个变量都有唯一的名字，在同一程序块中，不能被重复定义。

81

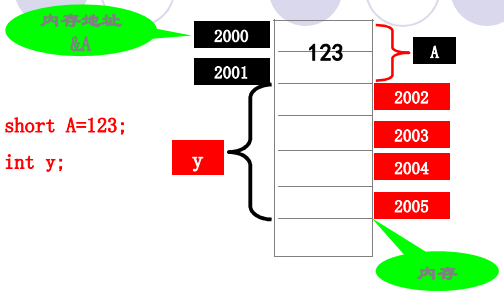
- 变量名

每个变量都必须有一个名字—变量名，变量命

在定义或说明变量时要指出其类型。每一个变量被指定为一确定类型，在编译时就能为其分配相

在程序运行过程中，变量值存储在内存中。在程序中，通过变量名来引用变量的值。变量有两个有用值：一是变量所表示的数据值，另一个是变量的地址值。A=123;&A

82



71

变量的定义

C中的任何一个变量在被引用之前必须定义；

C中可以随时定义变量，不必集中在执行语句之前；

在同一程序块内，不能定义同名变量，不同程序块内可以定义同名变量；变量可以在定义时初始化。未初始化的变量中有默认值或无效值。变量定义的一般形式：

类型名　变量名表；　(变量名：小写字母，见名知义)

决定分配字节数和数的表示范围

类型名：整型 int
实型（浮点型）float ，double
字符型 char

```
int i,j;  i ,j占4个字节
float z;  z占4个字节
int  index=100,big=1000;
```

- 变量名代表内存中的一个存储单元
- 用于存放该变量的值
- 该存储单元的大小由变量的数据类型决定

84

通常，定义了一个变量而未赋初值时，一般变量中存放的是随机值。因此，为使定义的变量有一确切的数值，需给定义的变量赋一初值。所以，C语言也允许在定义变量的同时为其赋初值，其形式为：

类型名 变量名1=常量1，变量名2=常量2，…；

如例1.3函数体中的第一条语句
int lower,upper,step ;
也可改写成 int lower=0,upper=200,step=20 ;
表示定义了一个整型变量lower初值为0,upper初值为200, step初值为20 .
一个变量代表着内存中一个具体的存储单元，用变量名来标识。存储单元中存放的数据称为变量的值，变量的值可以通过赋值的方法获得和改变。

C 语言的基本语句分类

与其它高级语言一样，C语言也是利用函数体中的可执行语句，向计算机系统发出操作命令。

C 语言的语句有以下几种

❖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。例如，int a,b;float x,y;

❖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。
例如，x=x+y;

❖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。
例如，printf(“Hello World\n”);

❖表达式语句
表达式语句由表达式后加一个分号构成。在c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。最典型的表达式语句是，在赋值表达式后加一个分号构成的赋值语句。
例如，“num=5”是一个赋值表达式，而“num=5;”却是一个赋值语句。

❖流程控制语句
用来控制程序执行过程的语句，例如：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。
C语言只有9条控制语句，又可细分为三种：
(1) 选择结构控制语句
if() ~ else ~, switch() ~
(2) 循环结构控制语句
do ~ while(), for() ~, while() ~, break, continue
(3) 其它控制语句
goto, return

❖复合语句
用花括号括住的一组任意语句。

```
void main()
{
  ……
   {……} /*复合语句。注意：右括号后不需要分号*/
  ……
}
```

复合语句的性质：
(1) 在语法上和单一语句相同，即单一语句可以出现的地方，也可以使用复合语句。
(2) 复合语句可以嵌套，即复合语句中也可出现复合语句。

❖空语句：空语句仅由一个分号构成。显然，空语句什么操作也不执行。
例如，下面就是一个空语句：

```
;
```

❖其他语句：包括编译预处理命令，用户自定义类型语句等。
例如，#define N 100