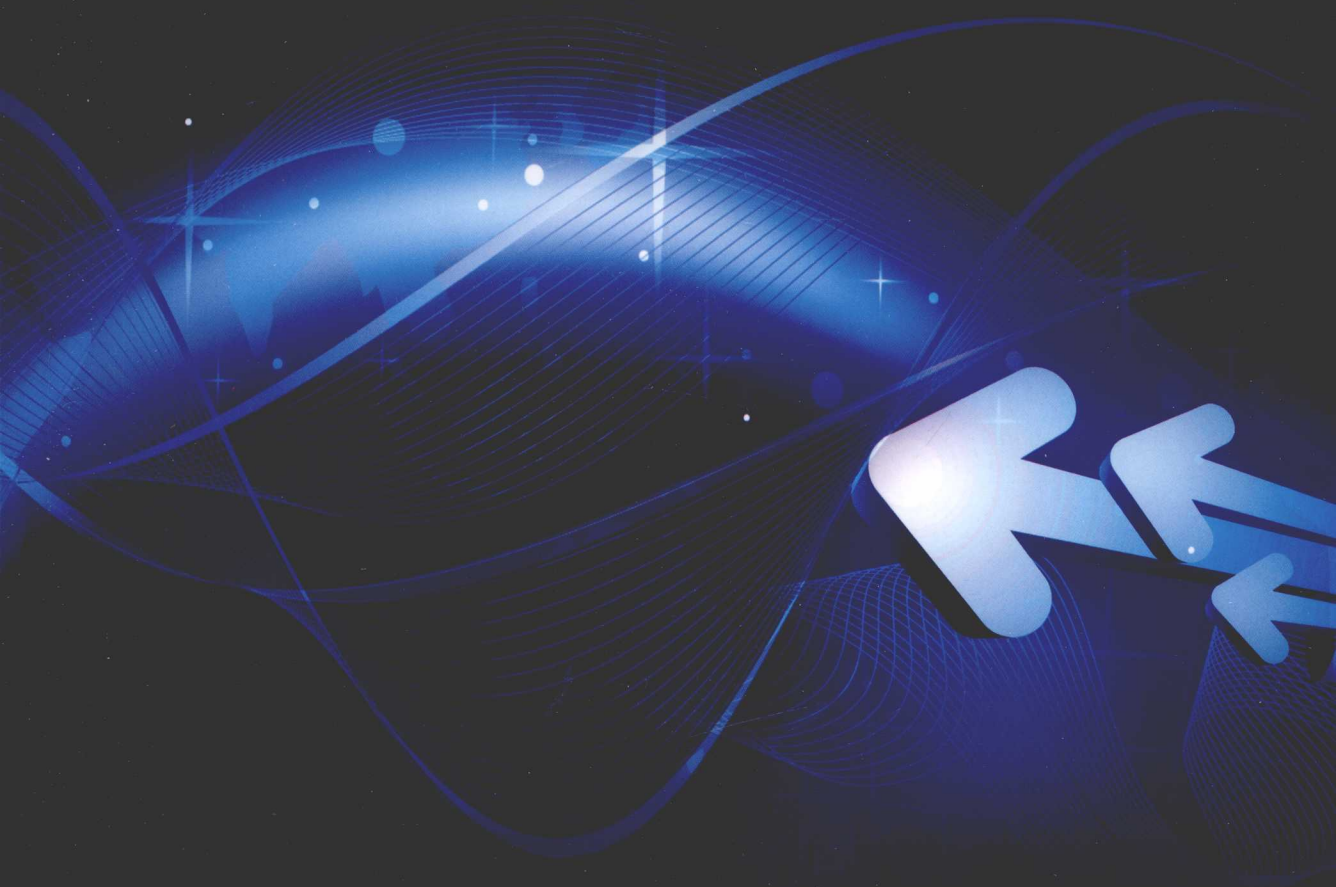
宋桂琴 编著

高级语言程序设计教程

C基础与C++

全国高校计算机应用系列教材



暨南大学出版社
JINAN UNIVERSITY PRESS

宋桂琴　编著

高级语言程序设计教程 C基础与C++

全国高校计算机应用系列教材

暨南大學出版社
JINAN UNIVERSITY PRESS
中国·广州

图书在版编目（CIP）数据

高级语言程序设计教程（C 基础与 C ++）/宋桂琴编著．—广州：暨南大学出版社，2010.9
（全国高校计算机应用系列教材）
ISBN 978-7-81135-540-6

Ⅰ.①高… Ⅱ.①宋… Ⅲ.①C 语言—程序设计—高等学校—教材 Ⅳ.①TP312

中国版本图书馆 CIP 数据核字（2010）第 107466 号

出版发行：暨南大学出版社

地　址：中国广州暨南大学
电　话：总编室（8620）85221601
　　营销部（8620）85225284　85228291　85228292（邮购）
传　真：（8620）85221583（办公室）　85223774（营销部）
邮　编：510630
网　址：http://www.jnupress.com　http://press.jnu.edu.cn

排　版：暨南大学出版社照排中心
印　刷：广州市怡升印刷有限公司

开　本：787mm×1092mm　1/16
印　张：16.25
字　数：425 千
版　次：2010 年 9 月第 1 版
印　次：2010 年 9 月第 1 次
印　数：1—1000 册

定　价：29.80 元

前言

在过去的30年中，C语言以具有丰富的运算符和数据类型、使用灵活、代码重用性高而广受人们的喜爱。C++语言是一种优秀的程序设计语言，它在C语言的基础上发展而来，包含了C语言的所有内容，同时又支持面向对象的程序设计。现在，大多数高校都把C/C++作为主要的教学语言。

本书是作者根据多年的教学经验，在听取同事意见，并对国内外同类著作和教材进行深入的比较研究后写成的。本书先介绍了程序设计的基本概念，接着以Visual C++6.0作为实验开发平台，通过大量精选的案例，全面系统地介绍了C语言基本数据类型、运算符、表达式、数据的输入输出、分支和循环控制结构、数组、函数、指针、结构体、枚举、编译预处理等内容。为使读者在熟悉了C语言的基础上，能轻松地过渡到C++语言的学习，本书在把C语言独立成一个部分进行介绍后，进一步介绍了C++语言，主要包括面向对象的程序设计方法、C++语言中类和对象的概念、继承性与派生类、运算符重载机制、多态性的用法、模板和输入输出流。

学习程序设计一定要有足够的耐心，绝不能浮躁。在设计程序时，要让自己“钻”到程序里，在程序中浸泡自己。同时一定要规范地书写代码，以提高程序的可读性。除了自己要多动手编写代码上机调试外，阅读并评价别人的代码也是提高自己编程能力的一个有效途径。另外，在学习计算机语言时，不能局限于一本参考书，要尽可能多看一些有关书籍，同时自己要多思考、多研究。

本书既可以作为计算机专业本科生程序设计课程的教材，也可以作为学习C/C++语言的入门参考书。

本书在编写过程中特别是在目录、内容的编排上得到同事的极大帮助，在此特向他们表示衷心的感谢。同时，本书参考了大量相关文献，并引用了其中的一些案例和内容，在此，对这些文献的作者表示最真挚的谢意。最后衷心感谢暨南大学出版社为了此书的出版而付出辛勤劳动的各位编辑。

因作者水平有限，书中难免会存在一些不当之处，真心地请求读者批评指正。

编　者

2010年5月

目 录

Ⅲ　面向对象程序设计（C++语言）

Ⅰ　解题和程序设计概要

1　计算机解决问题的初级概念

本章要点

◆日常生活问题的求解过程
◆问题的基本类型
◆计算机解决的问题

学习目标

◆了解问题求解的6个步骤以及问题的基本类型
◆能够用这6个步骤解决问题
◆了解计算机能帮助我们解决哪些问题

1.1　问题求解的一般概念

1.1.1　日常生活问题的求解过程

在生活中我们每天都会遇到各种不同的问题，其中大多都是需要我们做出选择的问题，如电视节目的选择、高考后要报考学校的选择、所学专业的选择、大学毕业后职业的选择等。那么，采用什么样的方法和步骤，才能使选择性问题得到最好的解决呢？我们通常采用以下步骤来求解问题，以确保得到最好的选择结果。

1. 明确问题

解决问题的第一步是明确问题。在解决问题之前应先知道问题是什么，例如，公司经理今天休息，他不知今晚做什么，让我们帮他出个主意。首先应明确我们遇到的问题：如何让经理度过一个漫长的夜晚。

2. 理解问题

接下来，必须搞清问题所涉及的方面，包括问题提出方的知识背景和我们自己的知识背景。为了让经理度过一个美丽的夜晚，我们要弄清他的知识背景：他的性格（喜欢安静还是热闹）、喜欢哪些娱乐方式、经济能力等。我们的知识背景：熟悉哪些娱乐方式、知道我们所在地有哪些特色小吃等。

3. 寻找备选方案

尽可能全面地列出备选方案，可征求不同人的意见，找到不同的解决方案，当然这些方案必须是可行的。例如，今晚我们让经理到太空走一遭的方案是可以的，但显然是不可行的。下面列出备选方案：

（1）看电视。
（2）邀请朋友。
（3）玩游戏。
（4）打高尔夫球。
（5）唱 KTV。
（6）参加朋友聚会。

当再也想不出其他方案时，这个列表就算完成了。

4. 从备选方案列表中找出最好的解决方案

在选择前，首先制定一个评定标准，然后根据这些标准来评价每种方案的利弊。比如那些花钱太多或经理不感兴趣的方案要统统删掉。

5. 列出所选方案的执行步骤

这些步骤必须包含在第二步所确定的知识范围内，然后准备一个步骤列表。

6. 评价解决方案

评价解决方案就是检查它的结果是否正确，是否令用户满意。例如我们可以通过询问经理是否过得开心，来确信制订的方案是否达到预期的效果。

1.1.2　问题的基本类型

解决问题的方案大致可分为两种类型：算法式方案和启发式方案。算法式方案就是指可以通过一系列明确动作来解决问题的方案。用算法式方案解决的问题称为算法式问题。例如，微分问题、计算地球轨迹、对 10000 个名字排序等都属于算法问题。启发式方案是指不能通过直观的步骤来解决问题，而是需要利用相应的知识和经验，通过不断的尝试最终才能解决问题的方案。用启发式方案解决的问题称为启发式问题，如下象棋、打牌等。那么计算机能帮我们解决哪些问题呢？

1.2　计算机解决的问题

计算机并不能解决人类遇到的所有问题。计算机主要处理算法问题，如数学计算过程、包含关系或逻辑处理的问题、反复执行一组数学型或逻辑型指令的问题。而让计算机处理启发式问题就困难些。人工智能这门科学的目标就是研究让计算机尽可能多地帮助人类处理一些启发式问题。下面给出了计算机解决问题的基本过程。

分析问题➡设计算法➡编写程序➡调试程序➡检测结果

计算机要能够工作，必须具备硬件和软件两个方面的条件。硬件是计算机本身的构成部分。计算机硬件的基本构成参见图 1－1。

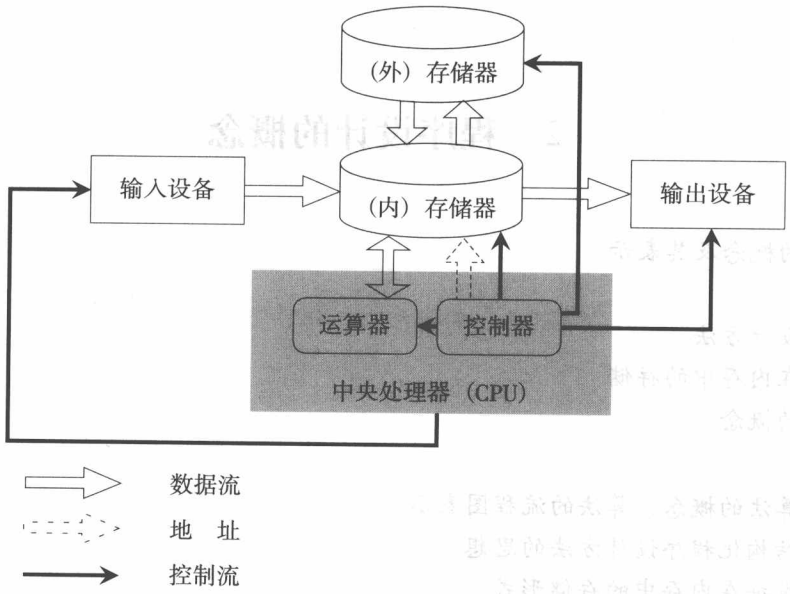


图 1－1　计算机硬件的基本构成

软件是计算机系统中的程序、支持程序运行的数据及相关支持文件。

小结

本章介绍了计算机解决问题的初级概念、解决问题的基本步骤以及计算机可以解决的问题。

习题

1. 列举生活中遇到的算法式问题。
2. 列举生活中遇到的启发式问题。
3. 描述下列问题的知识背景：

（1）开车。

（2）计算教师的月工资。

（3）计算学生大学四年的总学分。

（4）计算超市每天的净利润。

4. 写出解决下列问题的方案步骤（详细列出每一步）。

问题：给出年、月、日，计算该日是该年的第几天。

2 程序设计的概念

本章要点

- ◆算法的概念及其表示
- ◆程序
- ◆程序设计方法
- ◆数据在内存中的存储
- ◆函数的概念

学习目标

- ◆掌握算法的概念、算法的流程图表示
- ◆掌握结构化程序设计方法的思想
- ◆了解数据在内存中的存储形式
- ◆了解变量、常量、数据类型、函数的概念

2.1 算法和程序

2.1.1 算法

计算机解决的任何问题都可以通过按特定顺序执行一系列操作来完成，对操作的描述就是算法（algorithm）。算法包括了执行的操作和执行操作的顺序。例 2－1 给出了一个问题的算法描述。

例 2－1：有 50 个学生，要求将他们之中成绩在 80 分以上者打印出来。

假设，n 表示学生学号，ni 表示第 i 个学生学号；g 表示学生成绩，gi 表示第 i 个学生成绩。

算法描述如下（Si 表示步骤）：

S1：1→i

S2：如果 gi≥80，则打印 ni 和 gi，否则不打印

S3：i＋1→i

S4：若 i≤50，返回 S2，否则，结束。

1. 算法的特征

需注意的是，并不是能“计算”的问题才有算法，其实为解决问题而采取的方法和步骤都叫算法，如剑谱、菜谱、乐谱、数的排序、圆周率的计算方法、图书的检索、火车调度管理、学生成绩查询等。但并不是所有的算法计算机都能执行，计算机可执行的算法应满足下列特征：

（1）有穷性：一个算法应包含有限的操作步骤。

（2）确定性：算法中每一个步骤都应当是确定的，而不能是含糊的、模棱两可的。

（3）有零个或多个输入。

（4）有一个或多个输出。

（5）有效性：算法中每一个步骤都应当能有效地执行，并得到确定的结果。

2. 算法的表示

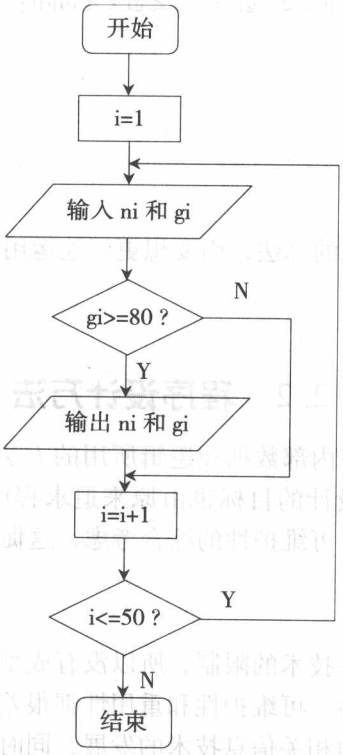


图 2－1 流程图表示的算法

算法可以用自然语言、流程图、伪代码、计算机语言表示。流程图表示的算法，直观形象，易于理解。伪代码使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。图 2－1 是例 2－1 中算法的流程图表示。

如果让计算机帮我们解决问题，我们就必须将解决问题的算法告诉计算机，即把算法用计算机能识别的方式表示出来，在计算机系统中，算法是用程序来表示的。

2.1.2 程序

程序是为完成一项特定任务而用计算机语言编写的一组指令序列。程序清单 2－1 给出了例 2－1 中算法的 C 语言表示。

程序清单 2－1：

```
//*************************************************
//c21.cpp
//*************************************************
  #include <iostream>
  using namespace std;
int main()
   { int i=1, ni, gi;
```

```
do
    { cin >> ni >> gi;
      if( gi >=80)
         cout << " ni = " << ni << " gi = " << gi << endl;
      i = i +1;
    }
while( i <=50);
  return 0;
}
```

计算机用程序描述求解问题的算法，而要想更好地运用计算机实现算法，必须选择合适的程序设计方法。

2.2　程序设计方法

程序设计方法是指组织程序内部数据和逻辑所用的方法。随着计算机应用的不断普及，程序规模随之越来越大，程序设计的目标也由原来追求程序的高效率转变为对程序的可读性、可靠性、移植性、重用性、可维护性的综合考虑，这促进了程序设计方法的发展。

2.2.1　结构化程序设计

早期的程序设计由于受硬件技术的限制，所以没有成型的设计方法，主要依赖个人技术和经验编程，因此程序的可读性、可维护性和重用性都很差。

随着计算机硬件技术和其他相关信息技术的发展，同时也为了能够编写和维护复杂庞大的程序，在 20 世纪 60 年代出现了结构化程序设计方法（structured programming，SP），又称面向过程的程序设计方法。

结构化程序设计强调程序设计风格和程序结构的规范化，提倡清晰的结构。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。具体方法包括：①自顶向下；②逐步细化；③模块化设计；④结构化编码。自顶向下是一种问题分解技术，将复杂的问题分解为一系列复杂性相对较低的子问题，然后逐个解决这些子问题，从而使整个问题得到解决；逐步细化是指对问题按层次进行分解，每一层不断将子问题细化，到了最后一层所有问题都是简单易解决的小问题；模块化设计是指将大程序划分成若干个子程序，每个子程序称为一个模块；结构化编码是指用结构化的计算机语言编写程序代码。

结构化程序设计是对程序的功能进行分解，将数据和对数据的处理过程分开，以过程为中心设计程序。因为用结构化程序设计方法设计的程序耦合度过高，所以这必然降低程序的安全性和重用性。例如，过程 1 和过程 2 都要操作数据 A（如图 2－2 所示），当数据 A 的结构改变时，过程 1 和过程 2 都要进行相应的修改；当过程 1 修改时，可能会引起过程 2 的修改。而且每一种相对于老问题的新方法都要带来额外的开销。

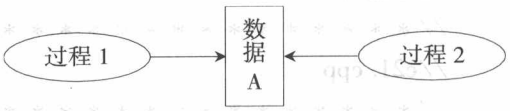


图 2－2　过程 1 和过程 2 操作数据 A

2.2.2 面向对象的程序设计

面向对象的程序设计出现于20世纪80年代中后期。面向对象方法解决问题的思路是：从现实世界中的对象（如人和事物）入手，尽量运用人类的自然思维方式从多方面来构造软件系统。关于面向对象的程序设计我们将在后续章节作进一步介绍。

根据程序设计方法选择合适的程序设计语言，就能编写可高效执行的程序。

2.3 程序设计语言

程序设计语言就是编写程序时使用的语言，是人与计算机交流的工具。计算机语言分为机器语言、低级语言和高级语言。

由计算机硬件系统可以直接识别的二进制指令组成的语言称为机器语言。计算机发展的初期，软件工程师们只能用机器语言来编写程序。这一阶段，在人类的自然语言和计算机编程语言之间存在着巨大的鸿沟。

汇编语言是常用的低级语言，它将机器指令映射为一些可以被人读懂的助记符，如ADD、SUB等。此时编程语言与人类自然语言间的鸿沟略有缩小，但仍与人类的思维相去甚远。同时，它的抽象层次太低，程序员编程时需要考虑大量的机器细节。

高级语言屏蔽了机器的细节，提高了语言的抽象层次，程序中可以采用具有一定含义的数据命名和容易理解的执行语句。C、C++、DELPHI、Ada、ForTran都属于高级语言。

2.4 计算机如何存储数据

2.4.1 内存中数据的存储

程序是用计算机语言编写的。计算机通过运行程序解决问题，在解决问题时，计算机首先将要运行的程序和求解问题所需要的数据存储在内存中。计算机的内存被划分成一个个存储单元，每个存储单元可存8个二进制位（bit），称为一个字节（byte）。在计算机内存中，数据以二进制形式存储，而且不同类型的数据所占的字节数也不同。如果存储的数据是有符号的数（数学中的正负数），就把符号放在最高位存储，“0”表示正数，“1”表示负数。而数值是用数的补码来存储的。例如，一个整数在内存中占4个字节，二进制数00000000000000000000000000001111是正整数15在内存中的存储形式；一个实数在内存中占4个字节，图2-3给出了实数的存储格式。其中指数占8位（包括指数的符号位），尾数占23位，因实数存储为二进制时，小数点前一位必须是1，且不存储，所以内存使用23位空间存储了24位的数。例如，12.65在内存中的存储形式是：01000001010010100110011001100 11。

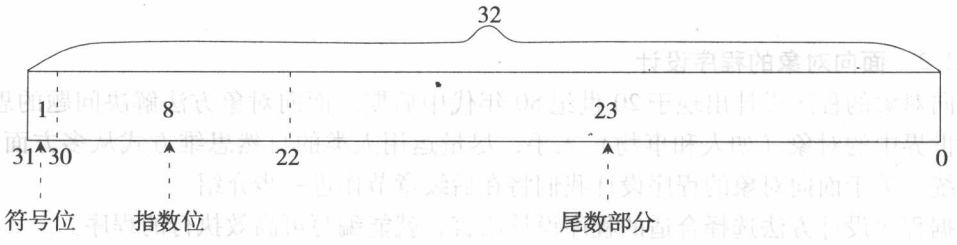


图 2-3　实数存储格式

关于数的原码、反码、补码不属于本书的范围，但学习计算机语言必须了解数在内存中的存储形式，希望读者查找有关参考书，进一步学习原码、反码、补码的知识。

要存储数据，首先要在内存申请到空间，程序是通过变量来申请空间的，我们把程序中用到的数据称为变量。

2.4.2　变量和常量

程序员必须给使用的每一个变量命名，并用这个名字访问它，即计算机使用变量名查找内存中与其对应的变量值。存放在变量所标识的内存单元中的值可以被刷新。那些在问题中值经常变动的数据可以定义成变量，如商品的价格、账户上的余额。

常量是一种特殊的变量，是一种固定的字符或数值，在整个程序运行过程中值都不能被改变。程序中，可以给常量命名，这时，常量就有了内存单元和名字。例如，可以给圆周率 3.1415926 命名为 PI，以后，在程序中，可以通过 PI 访问 3.1415926 这个值。常数是常量的一种表示，可直接使用，如 89、8.7。

常量可方便程序员编写和修改程序中频繁使用的常数值。存放在常量所标识的内存单元中的值不可以被刷新。那些在问题中值不变的数据可以定义成常量，如圆周率的值、一个公司薪水系统中公司的名字。

变量所申请内存单元的大小由所存储的数据的类型决定。一般整数所占的内存单元有 4 个存储单元（即 4 个字节）。

2.4.3　数据类型

程序运行时处理的数据都有自己的数据类型。数据类型是指数据的取值范围及对该数据进行的操作。数据类型决定了数据在计算机中的表示形式和计算机可以对其进行的处理。计算机中的数据类型主要有三种：数值型、字符型、逻辑型。如 C 语言中的整型和浮点型都属于数值型。

2.4.4　函数的概念

函数是问题解决过程中经常用到的基本操作，每种编程语言都有自己的函数。C 语言程序是以函数为单位，每个程序都是由若干个函数构成的；C++ 语言程序以类为单位，而类的行为是通过函数来描述的。

小结

算法就是解决问题的过程（方法和步骤）。算法的特征：有穷性、确定性、有零个或多

个输入、有一个或多个输出、有效性。算法的表示：算法可以用自然语言、流程图、伪代码、计算机语言表示。

程序是为完成一项特定任务而用计算机语言编写的一组指令序列。

程序设计方法是指组织程序内部数据和逻辑所用的方法。目前常用的程序设计方法有结构化程序设计方法和面向对象的程序设计方法两种。

程序设计语言就是编写程序时使用的语言，是人与计算机交流的工具。我们要把算法用计算机语言表示出来，计算机才会执行。

程序是用计算机语言编写的。计算机通过运行程序解决问题，在解决问题时，计算机首先将问题中所需要的数据存储在内存中，要存储数据，首先要在内存申请到空间，程序是通过变量和常量来申请空间的。

数据类型是指数据的取值范围及对该数据可进行的操作，数据类型决定了数据在计算机中的表示形式和计算机可以对其进行的处理。数据类型主要有三种：数值型、字符型、逻辑型。

函数是问题解决过程中经常用到的基本操作。

习题

1. 用流程图表示求解下列问题的算法。

（1）给出年、月、日，计算该日是该年的第几天。

（2）求 1 ~ 100 的和。

2. 给出下列数值在内存中的存储形式：

（1）5　67　-89。

（2）14.56　-9.8。

3. 下列问题需要用到的数据中，哪些应该定义为常量，哪些应该定义为变量。

（1）计算圆的周长和面积。

（2）员工工资 = 基本工资 + 奖金 - 税金 - 罚金，根据公式计算员工每月工资额。

Ⅱ 程序语言基础（C 语言）

3 数据类型与基本输入输出

本章要点

- ◆C/C ++ 语言及 C/C ++ 程序的实现过程
- ◆C 语言的程序结构
- ◆C 语言的标识符、数据类型
- ◆C 语言变量、常量的声明和使用
- ◆C 语言的运算符和表达式

学习目标

- ◆熟悉 C/C ++ 程序的执行过程
- ◆了解 C 语言的程序结构
- ◆掌握 C 语言中的基本数据类型
- ◆学会声明及使用变量和常量
- ◆掌握常用运算符的运算法则和优先级
- ◆能够正确书写 C 语言的表达式

3.1 概 述

3.1.1 C/C ++ 语言

C 语言是 1972 年由美国 AT & T 贝尔实验室的 Dennis Ritchie 设计发明的，并首次在 UNIX 操作系统的 DECPDP - 11 计算机上使用。它由早期的编程语言 BCPL（basic combind programming language）发展演变而来。在 1970 年，贝尔实验室的 Ken Hompson 根据 BCPL 语言设计出较先进的并取名为 B 的语言，最后由 Dennis Ritchie 发明了 C 语言。C 语言是国际上广泛流行的计算机高级语言，程序员既可以用它来编写系统软件，也可以用它来编写应用软件。

C ++ 是 C 语言的扩展，是 20 世纪 80 年代初由贝尔实验室的 Bjarne Stroustrup 开发的，1983 年正式取名为 C ++。

3.1.2 C/C ++ 程序的实现过程

用 C/C ++ 语言编写的程序必须利用 C/C ++ 程序开发环境经过编辑（edit）、预处理（preprocess）、编译（compile）、链接（link）和装入（load）过程，才能被计算机执行