·高等学校计算机基础教育教材精选·

计算基础
(C++语言实现)

赵宏　主编
王恺　副主编

清华大学出版社

·高等学校计算机基础教育教材精选·

计算基础
(C++语言实现)

赵宏　主编
王恺　副主编

清华大学出版社
北京

内 容 简 介

本书将计算基础与C++程序设计语言相结合，内容涵盖了利用计算机求解问题的一些基本原理和方法、C++语言的基础知识以及如何使用C++语言实现算法解决实际应用问题等。本书通过对一些精选问题的求解思路和方法的分析，针对初学者容易出现的错误和困惑的地方给出了大量提示，帮助读者更好地理解使用计算机解决问题的基本方法，初步具备使用C++程序设计语言解决实际问题的能力。

本书是专门为高等院校非计算机专业提高计算思维能力、学习计算机高级语言程序设计课程而编写的教材，面向初学者，不要求读者已熟悉相关的概念和计算机高级程序设计语言方面的背景知识。本书也适合自学者使用。

图书在版编目(CIP)数据

计算基础(C++语言实现)/赵宏主编，王恺副主编. --北京：清华大学出版社，2013.8
高等学校计算机基础教育教材精选
ISBN 978-7-302-32919-0

Ⅰ. ①计… Ⅱ. ①赵… ②王… 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(2013)第136283号

责任编辑：张瑞庆
封面设计：傅瑞学
责任校对：焦丽丽
责任印制：李红英

出版发行：清华大学出版社
网　　址：http://www.tup.com.cn，http://www.wqbook.com
地　　址：北京清华大学学研大厦A座　　**邮　　编**：100084
社 总 机：010-62770175　　**邮　　购**：010-62786544
投稿与读者服务：010-62776969，c-service@tup.tsinghua.edu.cn
质量反馈：010-62772015，zhiliang@tup.tsinghua.edu.cn
课件下载：http://www.tup.com.cn，010-62795954
印 装 者：保定市中画美凯印刷有限公司
经　　销：全国新华书店
开　　本：185mm×260mm　　**印　　张**：17.5　　**字　　数**：406千字
版　　次：2013年8月第1版　　**印　　次**：2013年8月第1次印刷
印　　数：1～2500
定　　价：29.50元

产品编号：054393-01

前言

2006年3月,美国科学基金会计算机与信息科学及工程部主任周以真(Jeannette M. Wing)教授首先提出并定义了“计算思维”(Computational Thinking,CT)的概念:计算思维是运用计算机科学的基础概念进行问题求解、系统设计以及人类行为理解等涵盖计算机科学之广度的一系列思维活动。2011年,图灵奖获得者 Richard M. Karp 提出了“计算透镜”(Computational Lens)理念,其核心是将计算作为一种通用的思维方式,通过这种广义的计算(涉及信息、执行算法、关注复杂度)来描述各类自然过程和社会过程,从而解决各个学科的问题。

在美国,“计算思维”的提出得到了美国教育界和科学界的广泛支持,并对“计算思维”所发挥的作用取得了共识。美国科学基金会启动“大学计算教育振兴的途径”项目并投入巨资进行计算教育的改革,同时还启动了以计算思维为核心的重大基础研究,进一步将计算思维的培养扩展到美国的各个研究领域。

在我国,计算思维的重要性也已引起了科学家和教育界的高度重视。教育部高等学校计算机基础课程教学指导委员会主任委员陈国良院士等积极地倡导把培养学生的“计算思维”能力作为计算机基础教学的核心任务,并由此建设更加完备的计算机基础课程体系和教学内容。

大学计算机课程是面向非计算机专业的计算机相关课程,其教学的基本目标是培养学生具备一定的计算机基础知识,掌握相关的软硬件技术,能利用计算机解决本专业领域中问题的能力。在“计算思维”理念下,大学计算机课程的教学又有了新的内涵和目标,需要进行科学系统的研究。

计算机程序设计相关的基本思想和方法是培养学生像计算机科学家一样去思考问题的有效途径。目前,不仅许多高校计算机专业和多数软件学院的程序设计课程选择了C++作为第一门语言,越来越多的理工科专业也把C++作为公共计算机基础课程。一方面是由于C++是应用最广的面向对象语言,另一方面是由于它有利于初学程序设计的学生学习一般的编程技巧。然而,由于公共计算机基础课的课时有限,近几年的教学实践表明,把C++语言作为高级语言程序设计的教学语言,使非计算机专业的理工科学生在有限的课时内掌握这门程序设计语言,难度很大。而且,目前的教材更多注重的是C++语言的语法和编程技巧,能够将计算机科学的基本思想与程序语言进行有机结合、注重非计算机专业计算思维能力培养的教材还没有看到。将计算思维与程序设计有机结合,目前仍有许多值得探索的地方。

对于在计算机科学领域工作的人来说，“编程”是一项重要的技能，人们经常认为计算机科学就是“编程”，但计算机科学不仅仅是编程。一些能够较好地学习计算机科学的人能够很容易掌握程序设计语言，是由于他们已经理解了计算机科学的一些基本原理。本教材将计算机科学的一些基本计算原理与用C++语言实现相结合。首先，讲解一些求解问题的算法和原理并不涉及“编程”，这样可以帮助读者专心于算法和原理本身；其次，介绍可以用C++来实现这些算法的相关语法和语句；最后，用C++语言实现相应的算法。当然，读者也可以用自己熟悉的其他程序设计语言，如Java、C#等来写程序实现算法。

本书由南开大学信息技术科学学院公共计算机基础教学部的教师结合多年的教学经验和大学计算机课程教学的发展需要编著而成。面向我国高校非计算机专业理工科学生，力争使他们在有限课时内学习一些计算原理，能够用C++程序设计语言实现一些基本算法，同时具有自觉使用计算思维去解决实际问题的能力。赵宏负责第1章至第3章、第7章和第10章的编写并负责全书的统稿和定稿，王恺负责第4章至第6章和第8章及第9章的编写。

在本书的编写过程中，得到了清华大学出版社张瑞庆编审的大力支持，在此表示真诚的感谢！

本书还参考了国内外的一些计算原理的开放课程网站和C++程序设计语言方面的书籍，力求有所突破和创新。由于能力和水平的限制，书中出现的不妥乃至错误之处，恳请读者指正。

作　者

2013年4月于南开园

目录

第1章　绪论

导读

要使用高级程序设计语言编写程序解决实际问题，首先需要了解程序设计与高级语言的基本概念，以及使用高级语言进行程序设计的基本方法。本章主要介绍程序设计的基本概念、步骤和方法；简要介绍C++高级程序设计语言，并通过两个简单的C++程序实例，介绍C++源程序的基本结构和组成元素；最后介绍在Visual C++ 2005集成开发环境下编辑、编译、连接和运行程序的步骤和方法。

1.1　程序设计的基本概念

1.1.1　用计算机求解问题的过程

图1-1是用计算机求解实际问题的流程图。人类使用计算机求解实际问题的基本步骤如下：

① 将实际问题抽象成数学模型。分析问题，从中抽象出处理的对象，用数据的形式对问题加以描述。

② 设计求解问题的算法。对描述问题的数据设计出相应的处理方法，从而达到求解问题的目的。算法需要用某种形式（如自然语言、流程图、伪代码等）表示出来。确定算法是最关键的一步。

③ 编写程序实现算法。将算法翻译成计算机能够读懂的语言，期间还需要调试和测试计算机程序。

④ 运行程序求解问题。通过计算机运行程序，按照所设计的算法对描述问题的数据进行处理，最终得到问题的结果。

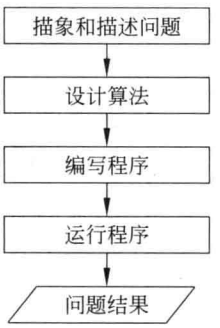

图1-1　用计算机求解问题的流程图

可见，计算机程序是通过计算机程序语言精确描述算法的模型，它的作用是指示计算机进行必要的计算和数据处理从而解决特定的问题。计算机程序涉及两个基本概念——数据和算法。**数据**是程序使用和处理的信息。面对问题，需要找出解决问题的方法，我们把这种能够在有限的步骤内解决问题的过程和方法称为**算法**。

下面通过一个简单的例子来说明如何建立数学模型和确定算法。

【例 1-1】 在高度为 100m 的铁塔上平抛一个物体，初速度为 20m/s，求其运动轨迹（以 0.1s 为时间间隔），直到物体落到地面为止。

问题分析：设坐标原点在塔底，物体初始位置是 x=0，y=100。如果用 v_0 表示初速度，则物体在时刻 t 的坐标是：

$$\begin{cases} x = v_0 t \\ y = 100 - \dfrac{1}{2} g t^2 \end{cases}$$

这两个公式就是问题的数学模型。该问题是求物体的运动轨迹，即每隔 0.1s 计算物体的坐标值。采用的算法是：利用循环结构按以上公式计算每一组 x、y 的值，直到 y=0 为止。

先用自然语言对算法进行描述：

① 定义变量 x、y 和 t，分别表示物体的坐标和时刻，并分别给它们赋初值 0、100 和 0。输出物体的初始坐标(x,y)。

② 利用公式，计算 t=0.1 时物体的坐标 x、y 并输出，然后 t 增加 0.1。

③ 判断：如果 y>0，则重复步骤②，否则结束。

用流程图描述算法，如图 1-2 所示。

用计算机求解问题的过程也称为程序设计的过程，是指设计、编制、调试程序的方法和过程，是寻找算法并用计算机能够理解的语言表达出来的一种活动。程序设计过程涵盖了上述步骤，即明确要解决的问题、将问题抽象成一定的数学模型、找出解决问题的算法、用程序设计语言描述算法、运行程序求解问题。

编程是将所设计的算法转换成计算机能够运行的代码的过程。编写一个程序并让程序运行起来，一般包括编辑、编译、连接和执行等步骤。编程步骤如图 1-3 所示。

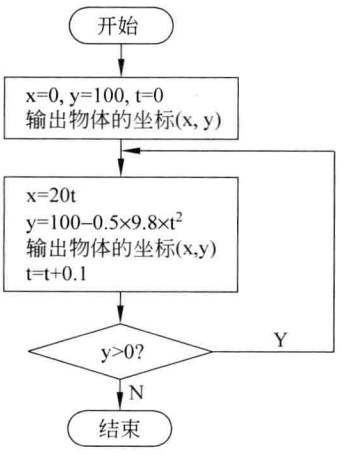

图 1-2　例 1-1 程序流程图

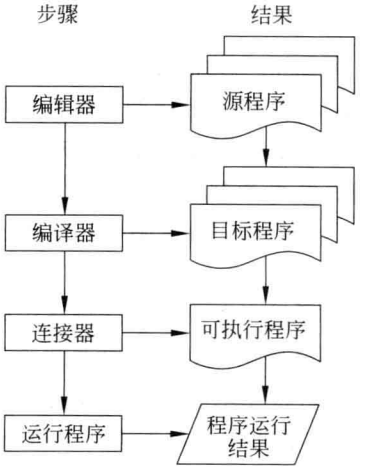

图 1-3　编程步骤

(1) 编写程序——编辑

使用文本编辑器编写程序，并将它保存到文件中，这个文件就是程序的**源代码**，保存源代码的文件称为源文件。源代码又称**源程序**，是程序员使用计算机高级语言（如C++

语言)编写的,计算机不能直接运行源代码。

(2) 编译成目标程序——编译

编译器是一个软件,运行该软件将源代码翻译成计算机能够识别的内部语言——**机器语言**。源文件经过编译后就会生成程序的目标代码文件,目标代码又称**目标程序**。

(3) 连接成可执行程序——连接

运行连接程序,将程序的目标程序和该程序调用的库函数的目标代码以及一些标准的启动代码组合起来,生成存储可执行代码的文件。可执行代码又称**可执行程序**。

(4) 运行程序——执行

运行可执行程序,得到程序的运行结果。

在编程的各个阶段都可能出现问题。在编译阶段发现问题,说明程序的语法存在错误,需要回到编辑步骤,对不符合高级语言的语法错误进行修改;在运行阶段出现问题,可能是语法问题,但更多的是程序算法思想问题,此时需要重新修改源代码,再重复进行编译、连接和执行,直到得到满足要求的程序。

1.1.2 程序设计方法

20 世纪 60 年代末期,随着"软件危机"的出现,程序设计方法的研究开始受到重视。结构化程序设计(Structured Programming, SP)方法是程序设计历史中被最早提出的。20 世纪 70 年代中后期,针对结构化程序设计在进行大型项目设计时所存在的缺陷,又提出了面向对象程序设计(Object Oriented Programming, OOP)方法。30 多年来针对面向对象程序设计方法的大量研究工作,使得它成为目前最重要的程序设计方法。

1. 结构化程序设计

结构化程序设计(SP)方法也称为面向过程的程序设计方法,反映了过程性编程的方法。它根据执行的操作来设计一个程序,简单易学、容易掌握,模块的层次清晰,降低了程序设计的复杂性,程序的可读性强。SP 方法便于多人分工开发和调试,从而有利于提高程序的可靠性。

例如,对于计算圆的面积和周长的问题,结构化程序设计方法的设计思想可用图 1-4 表示。

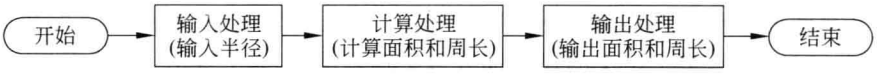

图 1-4 结构化程序方法的设计思想示意图

SP 方法的核心是将程序模块化,主要通过使用顺序、分支(选择)和循环(重复)3 种基本结构,形成具有复杂层次的结构化程序。它采用"自顶向下,逐步求精"的设计思想,其理念是将大型的程序分解成小型和便于管理的任务,如果其中的一项任务仍然较大,就将它分解成更小的任务。程序设计的过程就是将程序划分成为小型的、易于编写的模块的过程。程序的模块功能独立,只使用 3 种基本结构,具有单一出口和入口,增加了模块的独立性,可以像搭积木一样根据需要使用不同的模块。程序员开发程序单元(称为函

数)来表示各个任务模块。图 1-5 是采用结构化程序设计方法设计的程序的结构示意图。

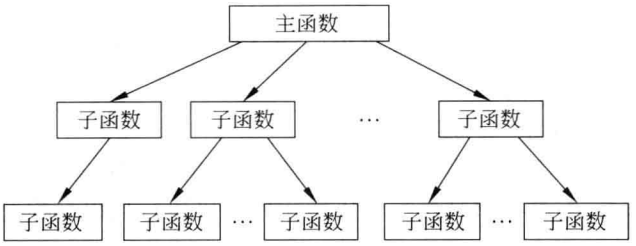

图 1-5　结构化程序结构示意图

到目前为止,仍有许多应用程序的开发采用结构化程序设计技术和方法。即使在目前广为流行的面向对象软件开发中也不能完全脱离结构化程序设计。

2. 面向对象程序设计

面向对象程序设计方法强调的是数据,而不是算法的过程性。根据人们认识世界的观念和方法,把任何事物都看成对象,复杂的对象是由简单的对象以某种方式组成,并认为世界是由各种对象组成的。

面向对象程序设计方法最重要的两个概念是类与对象,图 1-6 是类和对象的关系示意图。

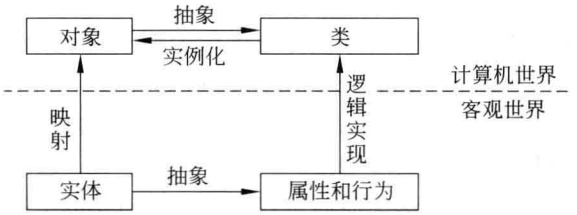

图 1-6　类和对象关系示意图

面向对象程序设计方法中的对象是一个封装了数据和操作这些数据的代码的逻辑实体;类是具有相同类型的对象的抽象,一个对象所包含的所有数据和代码可以通过类来构造。面向对象程序设计方法中的类通常规定了可以使用哪些数据和对这些数据可以执行哪些操作。数据表示对象的静态特征——属性,操作表示对象的动态特性——行为。例如,对于计算圆的面积和周长的问题,可以定义一个描述圆的类。类中定义的数据部分包括圆心的位置和半径;类中定义的操作部分包括输入圆心、输入半径、计算面积、计算周长以及结果输出等。

面向对象程序设计方法首先是设计类,它们准确地表示了程序要处理的东西,然后就可以设计使用这些类的对象的程序。它是自下向上(bottom-top)的编程方法,即从低级组织(如类)到高级组织(如程序)的处理过程。面向对象程序设计这种以对象为中心的设计方式符合人类认识事物和解决问题的思维方式和方法,较好地实现软件工程的 3 个主要目标:重用性、灵活性和扩展性。

面向对象程序设计方法还包含数据抽象、继承、动态绑定、数据封装、多态性、消息传

递等主要概念，本书将在第7章及后续章节作详细介绍。

提示：如果读者对SP和OOP暂时还不能理解，这是很正常的，它不会对下面内容的学习产生任何影响。其实，以后的学习也会遇到这样的情况，例如马上要遇到的C++中的头文件、主函数、cout、cin等。不必担心，先把它们作为一种定式来使用，随着不断地使用和学习的深入，很快就会豁然开朗的。

1.2 高级程序设计语言——C++

计算机本身只会完成几十种或上百种不同的简单动作，每一个动作称为一条指令。计算机设计者为每一个动作采用一个二进制的编码来表示，并为每一个动作设计一种通用的格式，即设计由指令码和内存地址组成的指令。所有指令构成了计算机的指令系统。计算机唯一可以读懂的语言就是计算机的指令，叫做**机器语言**，它被称为低级程序设计语言。如果程序员直接把让计算机完成的任务以指令序列的形式写出来，就是机器语言程序设计。

高级程序设计语言，例如C、C++、Java、C#、Fortran、Python等，是为解决使用机器语言编写程序困难的问题而逐步发展起来的，其语法符合人们的习惯，便于普通用户编程，但计算机却不懂。编译程序、编译系统或编译器就是能够把用高级语言写出的程序翻译为机器语言的指令序列的程序。

有了高级程序设计语言及其编译系统的帮助，人们可以编制出规模更大、结构更复杂的程序。

1. C++语言

20世纪70年代早期，美国Bell实验室为开发UNIX操作系统，在旧语言的基础上，开发了能将低级语言的效率、硬件访问能力和高级语言的通用性、可移植性融合在一起的C语言，C语言直至今天仍然被广泛应用。C++语言在20世纪80年代同样诞生于Bell实验室。C++语言是在C语言的基础上，沿用了C的大多数语法，并引入面向对象的特征，开发出一种过程性与对象性相结合的程序设计语言，1983年取名为C++。1998年国际标准化组织和美国国家标准局制定了C++标准，称为ISO/ANSI C++，也就是平时所称的C++。

C++语言开发的宗旨是使OOP方法和数据抽象成为软件开发者的一种真正实用技术。经过多次的改进和完善，目前的C++具有两方面的特点：第一，C++是C语言的超集，与C语言兼容，这使得许多C代码不经修改就可以经C++编译器进行编译；第二，C++支持面向对象程序设计，被称为真正意义上的面向对象程序设计语言。

2. C++开发工具——Visual C++

Visual C++是由微软公司开发的专门负责开发C++软件的工具，被称为集成开发环境(Integrated Development Environment，IDE)。集成开发环境包括编写和修改源代码的文本编辑器、编译器、连接器、程序调试和运行，以及其他程序开发的辅助功能和工具。通过这个工具，可以大大地提高程序员开发程序的效率。本书采用的IDE是Visual C++

2005，作为学习 C++ 语言和开发程序的工具。

1.3 C++ 程序

在学习 C++ 语言之前，请思考下面的问题。

- 实际问题如何用数据描述？
- 计算机如何存储数据？
- 计算机如何处理数据？
- 如何让计算机知道人们要进行什么操作？

在后面的学习过程中，应该能够逐渐对上述问题给出明确的答案。

下面通过两个简单的 C++ 程序实例，使读者对 C++ 源程序的基本组成和在 Visual C++ 2005 中进行程序编辑、编译、连接和运行步骤有一个初步的了解；并且通过模仿程序实例，能够立即上手编写一些简单的 C++ 程序。

1.3.1 简单 C++ 程序实例

【例 1-2】 编写一个程序，程序的功能是将“大家好!”输出到屏幕上。

完整的程序代码如下：

```
/*
第一个简单 C++程序____Chap1_1.cpp
设计日期:2013 年 4 月
*/
#include<iostream>                    //预处理命令
using namespace std;                  //命名空间
int main()                            //主函数
{
    cout<<"大家好!";                  //输出
    return 0;                         //函数返回
}
```

【例 1-3】 编写一个程序，程序的功能是将用户输入的两个整数求和，并将结果输出到屏幕上。

完整的程序代码如下：

```
//第二个简单 C++程序____Chap1_2.cpp
//设计日期:2013 年 4 月
#include<iostream>                    //预处理命令
using namespace std;                  //命名空间
int main()                            //主函数
{
```

```
    int x,y,result;                          //定义变量
    cout<<"请输入两个整数:";                  //输出
    cin>>x>>y;                               //从键盘输入数据
    result=x+y;                              //进行计算和赋值处理
    cout<<"两数之和为:"<<result<<endl;        //向屏幕输出数据
    return 0;                                //函数返回
}
```

下面,通过分析例 1-2 和例 1-3 两个 C++ 程序的含义,了解 C++ 程序的几个基本组成部分。

1.3.2 C++ 源程序的组成

一个 C++ 程序一般由编译预处理命令、函数、语句、变量、命名空间、输入输出、函数返回和注释等几部分组成。

1. 编译预处理命令

例 1-2 和例 1-3 中的"#include<iostream>"是一个编译预处理命令,它使程序具有了基本的输入输出功能。C++ 的编译预处理命令包括宏定义命令、文件包含命令和条件编译命令,都是以 # 开始。关于编译预处理的内容将在第 5 章详细介绍。

2. 函数

int main()是程序的主函数。一个 C++ 程序一般由多个函数组成。这些函数可以是用户根据需要自己编写的函数——用户自定义函数,也可以是直接使用系统提供的函数——标准库函数。函数体用花括号({})括起来。函数也称为模块,关于函数的内容将在第 5 章详细介绍。

提示:

- 程序都是从主函数 main 开始执行的。所以,任何一个程序必须有且只能有一个主函数 main。
- 例 1-2 和例 1-3 采用的都是 SP 程序设计方法。

3. 语句

语句是函数的基本组成单元。任何一条语句都以分号(;)结束。上面的两个程序都只有一个函数——主函数,该函数由用花括号({})括起来的多条语句组成。语句可以构成顺序、选择和循环 3 种程序控制结构。关于 C++ 的基本语句和程序控制结构的内容将在第 2 章和第 3 章详细介绍。

4. 变量

例 1-3 的语句"int x,y,result;"是变量定义语句,其中,int 表示变量的数据类型是整型。这条语句说明在程序中定义了 3 个整型变量 x、y 和 result。C++ 程序需要将数据放在内存单元中,变量名就是内存单元中数据的标识符,通过变量名来存储和访问相应的数据。关于变量的定义和使用将在第 2 章详细介绍。

5. 命名空间

例 1-2 和例 1-3 中的"using namespace std;"是 using 编译指令,表示使用命名空间