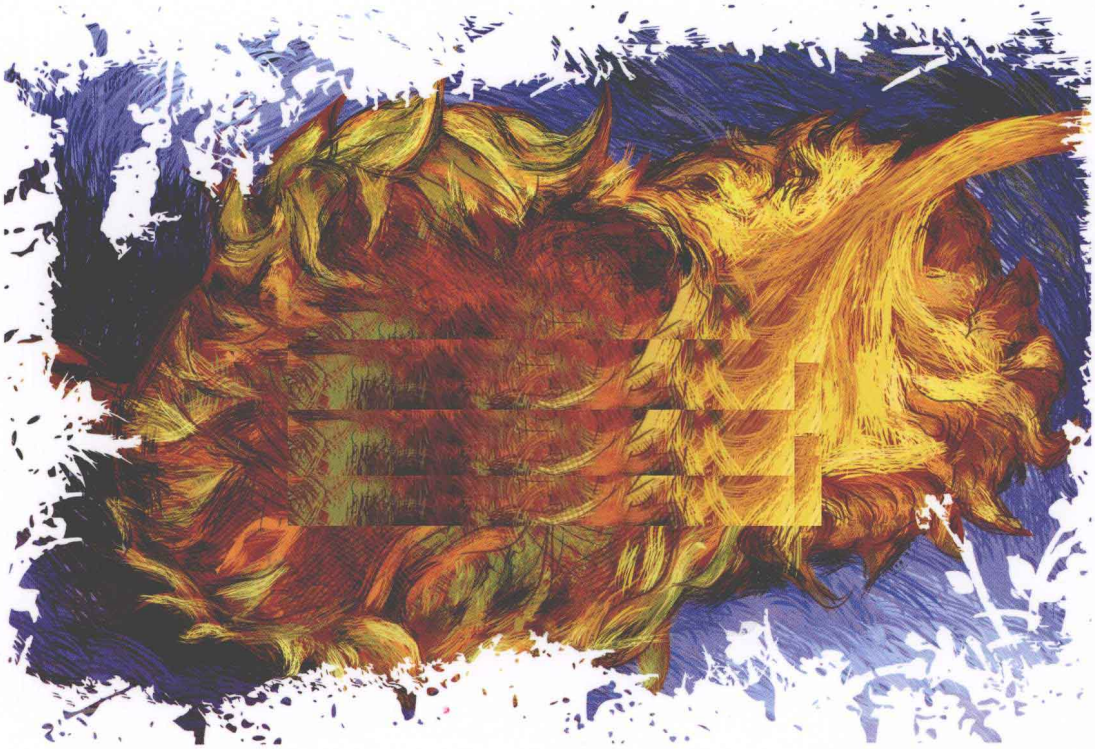
TURING 高等院校计算机教材系列

上海市精品课程

C++程序设计

思想与方法 第2版

翁惠玉 编著



人民邮电出版社
POSTS & TELECOM PRESS

TURING 高等院校计算机教材系列

C++程序设计思想与方法 第2版

翁惠玉 编著

人民邮电出版社
北京

图书在版编目（CIP）数据

C++程序设计 ：思想与方法 / 翁惠玉编著. -- 2版.
-- 北京 ：人民邮电出版社，2012.1
高等院校计算机教材系列
ISBN 978-7-115-26876-1

Ⅰ. ①C… Ⅱ. ①翁… 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(2011)第245440号

内 容 提 要

本书以C++为语言环境，重点讲授程序设计的思想和方法，涉及过程化程序设计和面向对象程序设计，内容包括数据类型、控制结构、数组、指针、数据封装、过程封装、运算符的重载、继承、多态性和异常处理等。第2版秉承以程序设计方法为主、程序设计语言为辅的思想，采用以问题求解引出知识点的方法，强调编程思想和知识的应用，增加了更多的习题和实例，多章都增加了“编程规范与常见错误”小节。结构更加合理，内容更加通俗易懂。

本书旨在使读者通过学习，并经过一定的训练和实践，能够掌握程序设计的方法，并具备良好的程序设计风格。本书可作为各大专院校计算机专业程序设计课程的教材，也可供从事计算机软件开发的科研人员作为参考资料。

高等院校计算机教材系列
C++程序设计：思想与方法（第2版）

◆ 编　　著　翁惠玉
责任编辑　明永玲

◆ 人民邮电出版社出版发行　　北京市崇文区夕照寺街14号
邮编　100061　　电子邮件　315@ptpress.com.cn
网址　http://www.ptpress.com.cn
北京艺辉印刷有限公司印刷

◆ 开本：800×1000　1/16
印张：23
字数：544千字　　2012年1月第2版
印数：5001－8 500册　　2012年1月北京第1次印刷
ISBN 978-7-115-26876-1

定价：49.00元

读者服务热线：(010)51095186转604　印装质量热线：(010)67129223
反盗版热线：(010)67171154

前　言

程序设计是计算机专业十分重要的一门课程，是实践性非常强的一门课程，也应该是一门非常有趣、让学生很有成就感的课程。但在教学过程中，很多学生的反应是：课听懂了，但题不会做，以至于最后丧失了兴趣。我认为主要的问题是教学过程中过分重视程序设计语言本身，过分强调理解语言的语法，而没有把思路放在解决问题的方法上面。因此，2008年我编写了《C++程序设计：思想与方法》，并由人民邮电出版社出版。

我校在教学中一直使用《C++程序设计：思想与方法》，在使用过程中，我发现了一些不足之处，也收集了学生和授课老师对本教材的一些意见和建议。在上海交通大学和人民邮电出版社图灵公司的支持下，我对《C++程序设计：思想与方法》一书进行了修订，并决定出版这一新版。

第2版保持了第1版的写作风格，继续秉承以程序设计方法为主、程序设计语言为辅的思想，采用以问题求解引出知识点的方法，在介绍语言要素的同时，更多地强调编程思想，强调知识的应用，并在以下几个方面做了较大的修改。

(1) 增加了习题量。习题是教材的重要组成部分，第1版的习题部分比较薄弱，所以在第2版中增加了较多的习题。这些习题不仅能帮助学生巩固学到的知识，更重要的是，这些习题能起到拓展知识的作用。

(2) 多章都增加了“编程规范与常见错误”小节。在教学过程中，我发现了一些初学者常犯的错误，于是将这些错误总结起来放在相关章的最后，希望能起到提醒的作用。一个好的程序就像一篇优美的文章，让人赏心悦目。一个好的程序必须符合良好的规范，编程规范旨在指导学生如何使程序更加规范，让学生在开始学程序设计时就养成良好的习惯。

(3) 扩充算法设计和问题求解过程方面的内容，增加了更多的实例。

(4) 对各章内容和文字进行了细致的修改，使结构更加合理、内容更加通俗易懂。

相信第2版会更加符合读者的需求。但由于作者水平有限，本书肯定存在很多不足，敬请读者批评指正。

最后，我要感谢上海交通大学电信学院程序设计课程组的各位老师，我的可爱勤奋的学生们以及关心、爱护和帮助过我的朋友们。

翁惠玉

2011年10月

目　录

试读结束：需要全本请在线购买：www.ertongbook.com

第1章

绪　论

自从第一台计算机问世以来，计算机技术发展得非常迅速，功能不断扩展，性能突飞猛进。特别是微型计算机的出现，使得计算机的应用从早期单纯的数学计算发展到处理各种媒体的信息。计算机本身也从象牙塔进入了千家万户。

计算机系统由硬件和软件两部分组成。硬件是计算机的物理构成，是计算机的物质基础；软件是计算机程序及相关文档，是计算机的灵魂。

1.1　计算机硬件

经典的计算机硬件结构是由计算机的鼻祖冯·诺依曼提出的，因此被称为冯·诺依曼体系结构。冯·诺依曼体系结构主要包括以下3方面的内容。

(1) 计算机的硬件由5大部分组成，即运算器、控制器、存储器、输入设备和输出设备，这些部分通过总线或其他设备互相连接，如图1-1所示。这5大部分协同完成计算任务。在现代计算机系统中，运算器和控制器通常集成在一块称为CPU的芯片上。

(2) 数据的存储与运算采用二进制表示。

(3) 程序和数据一样，存放在存储器中。

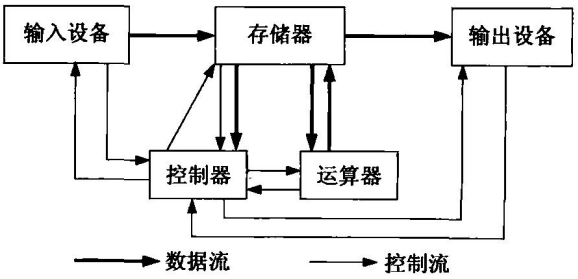


图1-1　计算机硬件系统的组成

运算器是真正执行计算的组件。它在控制器的控制下执行程序中的指令，完成算术运算、逻辑运算和移位运算等。不同厂商生产的机器，由于运算器的设计不同，能够完成的任务也不同，所能执行的指令也不完全一样。每台计算机能完成的指令集合称为这台计算机的指令系统或机器

语言。运算器由算术逻辑单元（ALU）和寄存器组成。算术逻辑单元完成相应的运算，寄存器用来暂存参加运算的数据和中间结果。

控制器用于协调机器其余部分的工作，是计算机的“神经中枢”。控制器依次读入程序的每条指令，分析指令，指挥各其他部分共同完成指令要求的任务。控制器由程序计数器（PC）、指令寄存器（IR）、指令译码器（ID）、时序控制电路及微操作控制电路等组成。程序计数器用来对程序中的指令进行计数，使控制器能依次读取指令；指令寄存器暂存正在执行的指令；指令译码器用来识别指令的功能，分析指令的操作要求；时序控制电路用来生成时序信号，以协调在指令执行周期中各部件的工作；控制电路用来实现各种操作命令。

存储器用来存储数据和程序。存储器可分为主存储器和外存储器。主存储器又称为内存，用来存放正在运行的程序和数据，具有存取速度快，可直接与运算器、控制器交换信息等特点，但其容量一般不大，而且一旦断电，信息将全部丢失。外存储器（包括硬盘、光盘等）用来存放长期保存的数据，其特点是存储容量大、成本低，但它不能直接与运算器、控制器交换信息，需要时可成批地与内存交换信息。

存储器内最小的存储单元是位（也称比特，bit）。它可以存放二进制数的一位，即一个0或一个1。bit是一个组合词，它是英语binary digit中的字母的组合。通常8个位组成一个字节（byte，即B）。字节是大部分计算机分配存储器时的最小单位。存储器的容量也是用字节来表示的，如某台计算机的内存为1G，则表明这台计算机的内存是1GB。

输入/输出设备又称外围设备，它是外部和计算机交换信息的渠道。输入设备用于输入程序、数据、操作命令、图形、图像和声音等信息。常用的输入设备有键盘、鼠标、扫描仪、光笔及语音输入装置等。输出设备用于显示或打印程序、运算结果、文字、图形、图像等，也可以播放声音和视频等信息。常用的输出设备有显示器、打印机、绘图仪及声音播放装置等。

事实上，计算机的工作过程与我们日常生活中的工作过程完全一致。当要求你做一道四则运算题时，你会通过某个途径获取这道题并记录下来。例如，你会把这道题目抄到自己的本子上，那么本子就是主存储器，你的笔就是输入装置。要计算这道题目，你会根据先乘除后加减的原则找出里面的乘除部分，在草稿纸上计算，结果写回本子上，再执行加减得到最后结果写回本子。在此过程中，你的大脑就是CPU，先做乘除后做加减的过程就是程序，草稿纸就是运算器中的寄存器，把答案交给老师的过程就是输出过程。

1.2 计算机软件

计算机硬件是有形的实体，可以从商店里买到。但如果计算机只有硬件，那么，它只能成为一个装饰品。计算机之所以有魅力是因为它有七十二般变化，可以根据我们的要求变换不同的角色。一会儿是计算器，一会儿是字典，一会儿是CD播放机，一会儿又成了一架照相机。要做到这些，必须有各种软件的支持。硬件相当于计算机的“躯体”，而软件相当于计算机的“思想”和处理问题的能力。一个人可能面临各种要处理的问题，他必须学习相关的知识；计算机需要解决各种问题，它需要安装各种软件。

试读结束：需要全本请在线购买： www.ertongbook.com

软件可以分为系统软件和应用软件。系统软件居于计算机系统中最靠近硬件的部分，它将计算机的用户与硬件隔离。系统软件与具体的应用无关，但其他软件都要通过系统软件才能发挥作用。操作系统就是典型的系统软件。应用软件是为了支持某一应用而开发的软件，如字处理软件和财务软件等。

1.3 程序设计

要让计算机能够完成某个任务，必须有相应的软件，而软件中最主要的部分就是程序。*程序*是计算机完成某个任务所需要的指令集合。

程序设计就是教会计算机去完成某一特定的任务，即设计出完成某个任务的程序。它包括3个过程：第一步是设想计算机是如何一步一步地完成这个任务的，这一阶段称为*算法设计*；第二步是用计算机认识的语言描述这个完成任务的过程，这一阶段称为*编码*；第三步是检验编好的程序是否正确完成了给定的任务，这一阶段称为*调试*。

1.3.1 算法设计

算法设计就是要设计一个使用计算机提供的基本动作来解决某一问题的方案，是程序设计的灵魂。算法设计的难点在于计算机提供的基本功能非常简单，而人们要它完成的任务却非常复杂。算法设计必须将复杂的工作分解成一个个简单的、计算机能够完成的基本动作。

解决问题的方案要成为一个算法，必须用清楚的、明确的形式来表达，以使人们能够理解其中的每一个步骤，无二义性。算法中的每一个步骤必须有效，必须是计算机可以执行的操作。例如，若某一算法包含“用π的精确值与r相乘”这样的操作，则这个方案就不是有效的，因为计算机无法算出π的精确值。而且，算法不能无休止地运行下去，必须在有限的时间内给出一个答案。综上所述，算法必须具有以下3个特点：

(1) 表述清楚、明确，无二义性；

(2) 有效性，即每一个步骤都切实可行；

(3) 有限性，即可在有限步骤后得到结果。

有些问题非常简单，一下子就可以想到相应的算法，没遇到什么大的麻烦就已写出一个解决该问题的程序；而当问题变得很复杂时，就需要更多的思考才能想出解决它的算法。与所要解决的问题一样，各种算法的复杂性也千差万别。大多数情况下，一个特定的问题可以有多个不同的解决方案（即算法），在编写程序之前需要考虑许多潜在的解决方案，最终选择一个合适的方案。

算法可以用不同的方法表示。常用的有自然语言、传统的流程图、结构化流程图、伪代码和PAD图等方法。本书主要采用伪代码的方法。所谓的*伪代码*就是介于自然语言和程序设计语言之间的一种表示方法，通常采用程序设计语言的控制结构，用自然语言表示基本的处理。例如，要设计一个算法打印1到100之间的数的平方表，用伪代码表示如下：

```
for ( i=1; i<=100; ++i)
    输出i和i的平方;
```

1.3.2 编码

编码就是用计算机认识的语言来表示算法。计算机能够理解的语言称为*程序设计语言*。

程序设计语言是人和计算机进行交流时采用的语言。随着计算机的发展，人类与计算机交互的语言也在进步，从早期的由二进制表示的机器语言发展到了如今的类似于英语的高级语言。

计算机刚出现时，人类和计算机交互的语言只有*机器语言*。每种计算机都有自己的机器语言。机器语言中的每条语句都是一个二进制的位串。

机器语言是由计算机硬件识别并直接执行的语言。机器语言能够提供的功能是由计算机硬件设计所决定的，因而能提供的功能非常简单，否则会导致计算机的硬件设计和制造过于复杂。不同的计算机由于硬件设计的不同，它们的机器语言也是不一样的。机器语言之所以必须由0、1组成是因为计算机内部的电路都是开关电路。0和1正好对应于开和关两种状态。

每条机器语言的指令一般分成两个部分，如下所示：

操作码	操作数

其中，操作码指出了运算的种类，如加、减和移位等。操作数指出参加运算的数据值或数据值的存储地址，如内存地址或寄存器的编号。

机器指令根据其功能一般可以分成控制指令、算术运算指令、逻辑运算指令、数据传送指令和输入输出指令。

用机器语言书写的程序很难阅读和理解，容易出错。而且程序员在用机器语言编程序时还必须了解机器的很多硬件细节。例如，有几类寄存器，每类寄存器有多少个，每个寄存器长度是多少等。由于不同的计算机有不同的机器语言，在一台计算机上的程序无法在另外一台不同类型的计算机上运行，这将引起大量的重复劳动。

为了克服机器语言的缺点，人们采用了与机器指令意义相近的英文缩写作为助记符，于是在20世纪50年代出现了*汇编语言*。汇编语言是符号化的机器语言，即将机器语言的每条指令符号化，采用一些带有启发性的文字串，如ADD（加）、SUB（减）、MOV（传送）和LOAD（取）。常数和地址也可以规定用一些符号写在程序中。

与机器语言相比，汇编语言的含义比较直观，使程序的编写更加方便，阅读和理解也更加容易。但计算机只“认识”由0、1组成的机器语言，并不“认识”由字符组成的汇编语言，不能直接理解和执行汇编语言写的程序。必须将每一条汇编语言的指令翻译成机器语言的指令后计算机才能理解。为此，人们创造了一种称为*汇编程序*的程序，让它充当汇编语言程序到机器语言程序的翻译，将汇编语言写的程序翻译成机器语言写的程序。

汇编语言解决了机器语言的可读性的问题，但没有解决机器语言的可移植性的问题。而且汇编语言的指令与机器语言的指令基本上是一一对应的，因此汇编语言提供的基本功能与机器语言是一致的，都是一些非常基本的功能，所以用汇编语言编写程序还是很困难的。

高级语言的出现是计算机程序设计语言的一大进步，FORTRAN、COBOL、BASIC和C++等都是高级语言。

高级语言是一种与机器的指令系统无关、表达形式更接近于科学计算的程序设计语言，从而

更容易被人们掌握。程序员只要熟悉简单的几个英文单词、代数表达式以及规定的几个语句格式就可以方便地编写程序，同时不需要知道机器的硬件环境。

由于高级语言是独立于机器硬件环境的一种语言，因而有较好的可移植性，减少了程序员的重复劳动。同时，高级语言提供的功能比机器语言强得多，因而解决问题更加容易。

1.3.3 程序的编译和调试

为了让用高级语言编写的程序能够在不同的计算机系统上运行，首先必须将程序翻译成该计算机特有的机器语言。例如，要在Macintosh机器上运行一个C++的程序，就需要运行一个特殊的程序将C++语言写的程序转换成Macintosh的机器语言；如果在IBM的PC机上运行该程序，则需要使用另一个翻译程序将该C++程序翻译成IBM的PC机上的机器语言。在高级语言和机器语言之间执行这种翻译任务的程序叫作*编译器*。

在大部分计算机系统上，运行程序之前需要先输入程序并将其保存在一个*文件*中。文件是存储在计算机外存中的信息集合的统称。每个文件都必须有一个文件名，通常用句点将文件名分成两部分，如myprog.cpp。前一部分由文件的创建者指定，通常选择一个能反映文件内容的字符串；后一部分称为*扩展名*，表示文件的类型，如扩展名“.c”表示文件的内容是C语言编写的程序，“.cpp”表示文件的内容是C++语言编写的程序。包含程序文本的文件称为*源文件*。

输入文件或修改文件内容的过程称为文件的*编辑*。各个计算机系统的编辑过程差异很大，不可能用一种统一的方式来描述，因此在编辑源文件之前，必须先熟悉所用的机器上的编辑方法。

一旦有了源文件，下一步就是使用编译器将源文件翻译成计算机可直接读懂的形式。这个过程也因机器而异，但在大多数情况下，编译器将源文件翻译成中间文件，这种中间文件称为*目标文件*。目标文件由特定的计算机系统的机器语言组成。但目标文件不能直接运行，这是因为在现代程序设计中，程序员在编程序时往往会用到系统已经做好的工具或其他程序员做好的工具。程序运行时会用到这些工具的代码。于是需要将目标文件和这些工具的目标文件放在一起，这个过程称为*连接*。连接以后的代码称为一个*可执行文件*。这是能直接在某台计算机上运行的程序。系统提供的工具或用户自己写的一些工具程序存放在一个*库*中。由一个源文件到一个可执行文件的转换过程见图1-2。

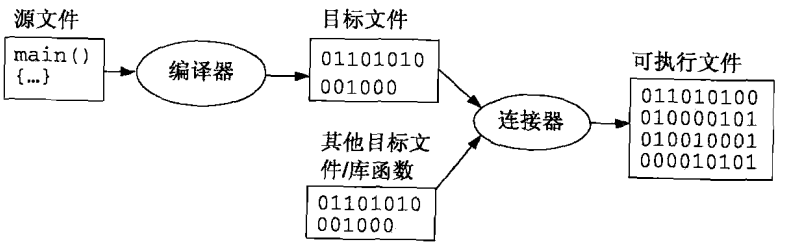


图1-2 编译过程

在编译过程中，编译器会找出源文件中的语法错误和词法错误。程序员可根据编译器输出的出错信息来修改源文件，直到编译器生成正确的目标代码。

语法错误还不是最令人沮丧的。往往程序运行失败不是因为编写的程序包含语法错误，而是程序合乎语法却给出了不正确的答案或者根本没给出答案。检查程序便会发现程序中存在一些逻辑错误，程序员称这种错误为bug。找出并改正这种逻辑错误的过程称为调试（debug），它是程序设计过程中一个重要的环节。调试一般需要运行程序，通过观察程序的阶段性结果来找出错误的位置和原因。

逻辑错误非常难以察觉。有时程序员非常确信程序的算法是正确的，但随后却发现它不能正确处理以前忽略了的一些情况；或者也许在程序的某个地方做了一个特殊的假定，但随后却忘记了；又或者可能犯了一个非常愚蠢的错误。

程序的调试及测试只能发现程序中的错误，而不能证明程序是正确的。因此，在程序的使用过程中可能会不断发现程序中的错误。在使用时发现错误并改正错误的过程称为程序的维护。

小结

本章主要介绍了程序设计所需的下列基础知识和基本概念。

- 计算机系统包括软件和硬件：硬件是计算机的躯壳，软件是计算机的灵魂。
- 计算机硬件主要有5大部分组成：运算器、控制器、存储器、输入设备和输出设备。
- 程序设计语言的发展经过了机器语言、汇编语言和高级语言3个阶段。
- 程序设计包括算法设计、编码、编译、调试及运行维护5个阶段。

习题

1. 简述冯·诺依曼计算机的组成及工作过程。
2. 简述寄存器、主存储器和外存储器的异同点。
3. 所有的计算机能够执行的指令都是相同的吗？
4. 投入正式运行的程序就是完全正确的程序吗？
5. 为什么需要编译？为什么需要连接？
6. 调试的作用是什么？如何进行程序调试？
7. 试列出一些常用的系统软件和应用软件。
8. 为什么在不同生产厂商生产的计算机上运行C++程序需要使用不同的编译器。
9. 什么是源文件？什么是目标文件？为什么目标文件不能直接运行？
10. 什么是程序的语法错误？什么是程序的逻辑错误？
11. 为什么不直接用自然语言和计算机进行交互，而要设计专门的程序设计语言？
12. 试列举出高级语言的若干优点（相比于机器语言）。
13. 为什么不同的计算机可以运行同一个C++程序，而不同的计算机不能运行同一个汇编程序？
14. 机器语言为什么要用难以理解、难以记忆的二进制位串来表示指令，而不用人们容易理解的符号来表示？

15．为什么电视机只能播放电视台发布的电视，DVD播放机只能播放DVD碟片，而计算机却既能当电视机用，又能当DVD机用，甚至还可以当游戏机用？

16．说明下面概念的异同点：

（1）硬件和软件

（2）算法与程序

（3）编译与解释

（4）高级语言和机器语言

第2章

通过例子学习

写一个程序就像是写一篇实验指导书，让计算机按你的实验指导书一步步往下做，最终就能完成任务。实验指导书有实验指导书的格式，程序也有程序的格式。本章将从一些简单的程序出发，介绍程序的基本框架及组成程序的最基本的元素。

2.1 第一个程序：输出 Hello world.

代码清单2-1给出了一个完整的程序的结构。一个C++程序主要由3个部分组成：程序注释、预编译命令和主程序。尽管这个程序结构非常简单，但它却是所有程序结构的典范，可以将它作为C++语言程序组织的范例。

代码清单2-1 输出Hello world.的C++程序

```
//文件名：hello.cpp                                    } 程序注释
//在屏幕上显示 "Hello world."

#include <iostream>                                    } 预编译命令

int main()
{
    std::cout << "Hello world." << std::endl;          } 主程序
    return 0;
}
```

2.1.1 注释

hello.cpp程序的第一部分为一段注释。在C++语言中，注释是从//开始到本行结束。在C++程序中也可以用C语言风格的注释，即在/*与*/之间所有的文字都是注释，可以是连续的几行。

注释是写给人看的，而不是写给计算机的。它们向其他程序员传递该程序的有关信息。在C++语言编译器将程序转换为可由机器执行的形式时，注释被完全忽略。

一般来说，每个程序都以一个专门的、从整体描述程序操作过程的注释开头，称为程序注释。它包括源程序文件的名称和一些与程序操作和实现有关的信息。程序注释还可以描述程序中特别

试读结束：需要全本请在线购买： www.ertongbook.com