



普通高等教育“十一五”国家级规划教材

C++程序设计教程

高福成 编著 边莫英 主审



天津大学出版社
TIANJIN UNIVERSITY PRESS

普通高等教育“十一五”国家级规划教材

C++ 程序设计教程

高福成 主编

边奠英 主审

高福成 潘旭华 王春嫻 宁爱军 编

天津大学出版社
地址：天津市卫津路92号
电话：(022) 23340000
网址：www.tjup.com
邮编：300072
天津泰宇印务有限公司
全国各埠均有代售
185mm × 260mm
本册
定价
天津
地区
发行
天津
地区
发行
天津
地区
发行



天津大学出版社

TIANJIN UNIVERSITY PRESS

内 容 提 要

本书力图将 C++ 语言中最精彩和最实用的内容展现在读者面前。全书共 10 章,介绍了 C++ 基本知识、数据运算、控制结构、构造数据类型、函数、指针与引用、类与对象、继承与派生、多态性与虚函数、模板等内容。全书以短小的实例解释 C++ 中的重点和难点问题,尽量推介语言发展中的新元素,摒弃那些陈旧的知识。本书以 Visual Studio.net 2005 作为开发平台,使读者能及时掌握最新的语言标准和编程环境,为进入 VC++ MFC、VC++ CLR 等可视化编程打好基础。

本书可作为高等理工学校 C++ 语言程序设计课程教材、全国计算机等级考试培训教材和考前复习参考书,也可供相关工程技术人员参考。

图书在版编目(CIP)数据

C++ 程序设计教程/高福成主编. —天津:天津大学出版社,2008.9

普通高等教育“十一五”国家级规划教材

ISBN 978-7-5618-2687-4

I . C . . . II . 高 . . . III . C 语言 - 程序设计 - 高等学校 - 教材 IV . TP312

中国版本图书馆 CIP 数据核字(2008)第 076884 号

出版发行	天津大学出版社
出 版 人	杨欢
地 址	天津市卫津路 92 号天津大学内(邮编:300072)
电 话	发行部:022-27403647 邮购部:022-27402742
网 址	www.tjup.com
短信网址	发送“天大”至 916088
印 刷	天津泰宇印务有限公司
经 销	全国各地新华书店
开 本	185mm × 260mm
印 张	23
字 数	575 千
版 次	2008 年 9 月第 1 版
印 次	2008 年 9 月第 1 次
印 数	1 - 3 000
定 价	40.00 元

凡购本书,如有缺页、倒页、脱页等质量问题,烦请向我社发行部门联系调换

版权所有 侵权必究

序 言

把计算机基础教育纳入大学公共基础课程的教学重点,是一件具有深远意义的事情。

计算机基础教育,既是大学生文化基础教育,又是人才发展教育,更是技术基础教育。

设置计算机基础课程,要明确促进学生个性健康发展的目标,重视认知与情感的统一、知识与能力的统一、主体精神与社会责任的统一,强调学生的素质发展和强调探索创新能力、实践能力、学习能力以及信息素养。

搞好计算机基础教学,首先要搞好教材建设。要编写出课程内容具有先进性、科学性和教育性,符合本课程内在逻辑体系和学生认知规律,表达形式符合国家有关规范标准的教材,不是一件易事。为此,我们邀请了长期从事高校计算机基础教学、富于教材编写经验的名师主持编写工作。

这套普通高等教育“十一五”国家级规划教材的出版,旨在推动高校计算机基础教育活动,提高大学生计算机基础知识水平,培养学生抽象思维、逻辑推理能力以及运用计算机分析问题解决问题的能力 and 强烈的信息意识。我们殷切地希望广大学生、教师和专家提出宝贵意见,以便再版时修改补充。

在这套教材编写出版过程中,得到了各方人士的大力支持和帮助,特别是天津大学出版社自始至终给予了积极的配合,在此,我们表示衷心的感谢。

天津市普通高等学校计算机基础课程教学指导委员会

2008年5月

前言

C++ 语言是当前程序设计领域广泛使用的面向对象的程序设计语言,它在运行时间、代码紧凑性和数据紧凑性方面能够与 C 语言相媲美,成为介绍面向对象程序设计的主要语言,在当前程序设计领域也占有半壁江山。

C++ 语言是在 C 语言的基础上增加了“类”而诞生的。C 语言适应了底层系统程序设计的需要,它以高效、灵活、功能丰富、表达力强和移植性好等特点著称于世,C++ 语言则适应了面向对象程序设计的需要。根据《C++ 编程思想》(“Thinking in C++”)一书所述,C++ 语言与 C 语言的效率相差约 $\pm 5\%$,在大多数场合 C++ 语言完全可以取代 C 语言,然而在单片机等需要谨慎利用空间、直接操作硬件的地方还是使用 C 语言。所以,C 语言和 C++ 语言并存的局面还会延续下去。

程序设计语言只是进行程序设计与编程的工具。我们要强调语言知识的学习,很难想象一个对语言功能没有深入了解的人能够真正使用它;但是,我们更强调如何用语言这个工具解决实际问题,在用中学,在学中用,在学习语言的过程中了解这种语言是如何支持编程技术的,并通过编程实践的训练,使自己成为一名优秀的编程人员和设计人员。

本书是一本给初学者介绍 C++ 语言的大学基础教材,希望奉献给那些想学习 C++ 语言的读者,尤其是那些理工科的学子们。本书有如下特点:

①用短小的程序设计例题介绍和解释 C++ 语言的重点和难点。C++ 语言有不少概念是比较难理解和掌握的,最好的方法是通过程序进行学习。为此,本书精选那些短小的程序设计例题将概念阐述清楚,使读者避免陷入长程序的困惑之中。

②使用目前较新的 Visual Studio.net 2005 集成环境,它提供了 VC++ .net、VB.net、C# 和 VJ++ 统一的开发平台,既可以开发字符界面的控制台应用程序,也可以开发 VC++ MFC 应用程序、VC++ CLR 应用程序,还可以开发 ATL Web 应用程序。

③每章都提供了学习指导和学习总结,希望以此引导读者掌握每一章的重点和难点。

④每章都配有大量的练习题。其中,选择题用于验证对基本概念的了解,填空题用于考核读者阅读程序的水平,程序改错题和编程题用于检测读者调试程序和解决实际问题的能力。书末列出了习题答案,以方便读者自学。

本书共 10 章,前 6 章全面介绍 C++ 支持结构化程序设计的基本内容,包括 C++ 语言简介、数据类型和数据运算、程序控制结构、构造数据类型和自定义数据类型、函数、指针和引用,除了引用外,这些都是 C++ 程序设计的基础;后 4 章系统介绍 C++ 面向对象的主要特征:封装性(类与对象)、继承性(基类和派生类)和多态性(重载与虚函数)以及 C++ 的高级概念及编程技术(模板)。对于 C++ 的输入输出流库的使用,则本着实用和够用的原则分散在各个章节中,随用随介绍。

全书由高福成主编,边莫英主审。参加编写的有潘旭华、王春娴、宁爱军等长期从事 C 和 C++ 教学的经验丰富的教师。

本书在编写过程中,作者参阅了国内外许多相关著作及网上的热点问题讨论,从中吸收了新的思想、新的内容和新的要求,力图将 C++ 语言中最精彩和最实用的内容展现在读者面前。由于作者水平和能力有限,书中可能有许多遗漏甚至错误之处,恳请读者提出宝贵意见。

本书的问世,始终得到天津市高等学校计算机课程教学指导委员会的关心和指导,得到编者所在学校的大力支持和帮助。天津大学出版社也为本教材立项付出了努力。在此,编者一并表示最诚挚的谢意。

编者

2008 年 2 月

目 录

第1章 C++ 语言简介	(1)
1.1 C++ 与 C 语言的关系	(1)
1.2 C++ 对面向对象程序设计的支持	(2)
1.3 C++ 程序的基本组成单元	(3)
1.4 简单的 C++ 程序设计	(8)
1.5 C++ 程序的开发过程	(10)
1.6 Visual Studio.net 集成开发环境	(11)
1.7 小结	(18)
习题1	(18)
第2章 基本数据类型和数据运算	(20)
2.1 基本数据类型	(20)
2.2 常量及其类型	(22)
2.3 变量的定义及初始化	(24)
2.4 符号常量	(27)
2.5 运算符	(28)
2.6 表达式	(35)
2.7 不同类型数据的输入与输出	(46)
2.8 小结	(50)
习题2	(51)
第3章 程序流程控制和编译预处理	(57)
3.1 结构化程序设计的三种基本结构	(57)
3.2 选择结构	(58)
3.3 循环结构	(69)
3.4 转移控制语句	(77)
3.5 编译预处理	(80)
3.6 小结	(83)
习题3	(83)
第4章 构造数据类型和自定义数据类型	(91)
4.1 数组	(91)
4.2 结构	(112)
4.3 枚举	(119)
4.4 类型定义 (typedef)	(120)
4.5 小结	(122)
习题4	(122)
第5章 函数	(129)
5.1 函数的定义和调用	(129)
5.2 函数间的数据传递	(135)
5.3 变量的存储类型对函数调用的影响	(137)
5.4 函数的存储类型对函数调用的影响	(145)
5.5 函数的递归调用	(145)
5.6 内联函数和重载函数	(148)
5.7 小结	(153)

习题5	(153)
第6章 指针和引用	(162)
6.1 指针的概念	(162)
6.2 指针的运算	(166)
6.3 指针和数组	(169)
6.4 用指针进行内存动态分配	(180)
6.5 引用	(182)
6.6 函数参数的指针传递和引用传递	(185)
6.7 指针型函数返回值和引用型函数返回值	(195)
6.8 小结	(199)
习题6	(199)
第7章 类与对象	(209)
7.1 类与对象的概念和定义	(209)
7.2 对象的初始化、构造函数和析构函数	(214)
7.3 类的包含和子对象的初始化	(221)
7.4 对象的使用	(223)
7.5 对象在函数间的传递	(226)
7.6 对象的作用域和生命期	(229)
7.7 静态成员	(230)
7.8 友元	(231)
7.9 共享数据的保护	(234)
7.10 小结	(236)
习题7	(236)
第8章 继承性与派生类	(244)
8.1 基类和派生类	(244)
8.2 单一继承	(245)
8.3 多重继承	(254)
8.4 虚基类	(263)
8.5 小结	(266)
习题8	(267)
第9章 多态性与虚函数	(274)
9.1 多态性	(274)
9.2 运算符重载	(274)
9.3 静态联编和动态联编	(283)
9.4 虚函数	(285)
9.5 纯虚函数和抽象类	(290)
9.6 小结	(292)
习题9	(293)
第10章 模板	(299)
10.1 模板的概念	(299)
10.2 函数模板	(301)
10.3 类模板	(307)
10.4 模板应用举例——单链表操作	(311)
10.5 小结	(318)
习题10	(318)
附录 习题参考答案	(324)

第1章 C++ 语言简介

C++ 是广泛应用的程序设计语言,是一种技术;VC++ 或 Visual Studio.net 则是一种程序设计环境,是一种工具。本章通过 C++ 程序实例介绍 C++ 程序的基本结构以及结构化程序设计方法和面向对象程序设计方法的概念,并以 Visual Studio.net 为开发平台介绍 C++ 程序的开发过程。

1.1 C++ 与 C 语言的关系

C 语言的祖先是 ALGOL 60(ALGOritm Language)。ALGOL 60 是 1960 年由国际计算机委员会设计的一种面向过程的结构化程序设计语言,用它编写的程序具有可读性和可移植性好的特点。但是,它不能直接对硬件进行操作,不宜用来编写系统程序。真正意义上的 C 语言是 Dennis M. Richie 于 1972 年在 B 语言的基础上修改而成的。C 语言以表达简明、使用灵活、结构化的流程控制、丰富的数据结构和操作符集合、良好的程序可移植性和高效率的目标代码为特征,不仅具有高级语言的要素,还兼有低级语言的功能,因此既可用于编写系统程序,也可用于编写不同领域的应用程序。特别是 1973 年, Ken Thompson 和 Dennis M. Richie 合作用 C 语言改写了 UNIX 操作系统(其中 C 语言代码占 90% 以上,只保留了少量汇编语言代码),从而使 UNIX 操作系统向其他类型的机器上移植变得相当简单,也使 UNIX 操作系统和 C 语言作为软件设计的良好工具开始风靡世界,被誉为软件工程师的语言。1978 年,以 UNIX 第 7 版中的 C 编译程序为基础,Brain W. Kernighan 和 Dennis M. Richie 合著了影响深远的名著“The C Programming Language”,书中介绍的 C 语言成为后来广泛使用的 C 语言版本的基础,称为 K&R C 语言。其后的十几年中,适用于不同机种和不同操作系统的 C 编译系统相继问世,从而把 C 语言的应用推向了更加广泛普及的阶段。1983 年美国国家标准局 ANSI 开始制定 C 语言标准,到 1989 年 12 月 14 日批准为 ANSI X3.159—1989,称为 ANSI C 或 C89。1995 年的 C95 标准对 C89 作了修正,1999 年国际标准化组织(ISO)颁布了编号为 ISO IEC 9899 的最新标准,称为 C99。该标准是对 C 的最新完整描述,包括了 C 语言的扩展和新的函数库。

从本质上说,C 语言是一种结构化的面向过程的程序设计语言。所谓结构化程序设计(Structure Programming, SP),是指采用自顶向下的分析方法,从系统的功能入手,将系统分解为若干功能模块,在对每个功能模块单独编程时,均使用只有一个入口和一个出口的程序控制结构(顺序结构、选择结构和循环结构)。这种按照工程的标准和严格的规范设计的程序具有较好的可读性及可维护性,能有效地组织整个系统中的各个过程(或函数)和处理大量的数据;所谓面向过程,是指用 C 语言编程时,必须按照算法的实现过程逐条语句编写,通知计算机一步一步去做。

由于用户的需要和软、硬件技术的不断发展变化,按照功能划分设计的系统模块必然是易变和不稳定的,这样开发出来的模块可重用性不高,尤其是那些大规模系统的设计变得十分困难。统计表明,对于超过 25 000 行代码的程序而言,用 C 语言已经很难管理。

因此,进入 20 世纪 80 年代后,面向对象的程序设计(Object Oriented Programming, OOP)概念日益普及。所谓面向对象,是指通过类和对象把程序所涉及的数据结构和对它施行的操作有机地组织成模块,将数据及对数据的处理进行最大限度的封装,并通过继承和重载等手段,使开发出来的软件易重用、易修改、易测试、易维护、易扩充。

正如其他传统的程序设计语言都在发展自己面向对象的新版本一样,C 语言也在发展的同时,朝着支持面向对象程序设计的方向迈出了步伐。1980 年,美国 AT&T 的贝尔研究所的 Bjarne Stroustrup 推出了 C 语言的超集 C++ 语言,也叫“带类的 C”,1983 年正式命名为 C++。

C++ 全面兼容 C,支持面向过程的程序设计,许多 C 程序可以不经任何修改为 C++ 所用;同时,C++ 也支持面向对象的程序设计。虽然 C++ 不是最早的面向对象的程序设计语言,但它很受人喜欢,使用相当广泛。1998 年国际标准化组织(ISO)颁布了 C++ 语言的国际标准 ISO/IEC 14882—1998,虚函数、运算符重载、多重继承、模板、异常、RTTI、名字空间等新特性均被加入标准。

C++ 语言的发展大概可以分为三个阶段:第一阶段从 20 世纪 80 年代到 1995 年,这一阶段 C++ 语言基本上传统类型上的面向对象语言,并且凭借着接近 C 语言的效率,在工业界使用的开发语言中占据了相当大份额;第二阶段从 1995 年到 2000 年,这一阶段由于标准模板库(STL)和后来的 Boost 等程序库的出现,泛型程序设计在 C++ 中占据了越来越多的比重性。同时由于 Java、C# 等语言的出现和硬件价格的大规模下降,C++ 受到了一定的冲击;第三阶段从 2000 年至今,由于以 Loki、MPL 等程序库为代表的产生式编程和模板元编程的出现,C++ 出现了发展历史上又一个新的高峰。这些新技术的出现以及和原有技术的融合,使 C++ 已经成为当今主流程序设计语言中最复杂的一员。

毋庸讳言,在开发系统程序(如操作系统、嵌入式系统等)和要求运行速度的小型应用程序(如接口程序、通信程序、自动控制程序等)时,C 语言仍然是不可替代的工具,但在开发大型图形界面的应用程序时,C++、C#、VB、Java 等面向对象的语言当属首选。尤其是以 C++ 为核心语言的 VC++ 和 VC++ .net 等开发工具,可以大大简化程序设计的难度,减轻程序开发人员的劳动,提高程序开发的速度。

1.2 C++ 对面向对象程序设计的支持

程序是为了解决某个实际问题而编写的计算机指令的有序集合。计算机程序主要描述以下两部分内容:

- ①描述问题的数据及其相互关系,即数据结构的内容;
- ②描述问题的处理规则和步骤,即求解问题的算法。

20 世纪 70 年代以前,程序设计方法主要采用流程图,随着结构化设计(Structure Programming, SP)思想趋于成熟,整个 20 世纪 80 年代 SP 是主要的程序设计方法。20 世纪 80 年代后,面向对象的程序设计方法和技术成为软件研究和应用开发中最活跃的一个领域。

1. OOP 方法的基本思想

OOP 方法认为世界是由各种对象组成的,任何事物都是对象。若干个简单对象可以通过某种方式组织成复杂对象,因此,整个世界就是一个最复杂的对象。每个对象都有属性和行为两大特征:属性表示对象的状态,通常用数据描述;行为则表示对象具有的功能,通常用方法

(即对数据的各种操作)描述。OOP 方法把数据及处理这些数据的过程合并为一个独立的“对象”,以“对象”为中心进行分析和设计,使这些对象形成了解决目标问题的基本构件。

从解决过程上说,OOP 方法采用自底向上方法,首先将问题空间划分为一系列对象的集合,再将对象集合进行分类抽象,一些具有相同属性行为的对象抽象为一个类,类还可以抽象分为子类、超类(超类是子类的抽象);其次采用继承来建立这些类之间的联系,形成层次结构;同时对于每个具体类的内部结构,又可采用自顶向下逐步细化的方法由粗到细精化之;调试运行时通过向类对象发送消息来完成,对象执行相应操作并返回结果使对象集的初始状态变成了终止状态。所以,面向对象的程序设计主要是不断设计新的类和创建对象的过程。正由于 OOP 方法紧抓“模型化世界”的对象,不仅符合人们的思维习惯,也解决了结构化程序设计方法所不能解决的代码重用性的问题。

OOP 方法具有两个显著的优点:一是开发时间短、效率高、可靠性高,开发的程序更健壮,特别是由于面向对象编程的可重用性,可以在应用程序中大量采用成熟的类库,从而缩短了开发时间;二是应用程序更易于维护、更新、修改和升级。

2. OOP 方法的基本特征

C++ 支持 OOP 方法,主要体现在它具有 OOP 方法所应具有封装性、继承性和多态性三个基本特征。

封装是将数据及处理数据的代码包含在一个称为类的数据类型中。对象是类的实例。对象中的某些数据和代码只能用于本对象,而不能由对象外的任何实体所存取,这样就提供了一个有效的保护级,以避免数据被意外修改或不正确使用。通常,应禁止直接访问一个对象的数据和代码,只能通过操作接口访问对象,这称为信息隐藏。事实上,封装是为信息隐藏提供的支持手段,而信息隐藏则可以将修改应用程序带来的影响减少到最低程度。封装性减少了程序成分间的相互依赖,降低了程序的复杂性,提高了程序的可靠性和数据的安全性。

继承性反映的是类与类之间的不同抽象级别。根据继承与被继承的关系,类可分为基类和派生类。基类也称为父类,派生类也称为子类。正如“继承”的字面含义一样,子类从父类那里获得所有的属性和方法,并且可以对这些获得的属性和方法加以改造,使之具有自己的特点,而这些新特点则将上层类的同名属性屏蔽掉。继承性使得相似的对象可以共享程序代码和数据,避免了信息的冗余,是实现软件重用的关键。

多态性是实现一种接口、多种方法的技术。换言之,多态是指同样的消息被不同类型的对象吸收,会导致完全不同的行为。将多态性映射到现实世界中,则表现为同一个事物随着环境的不同,可以有不同的表现形态及与其他事物通信的不同方式。多态性使程序员能在一个类等级中使用相同函数的多个版本,从而可以集中精力开发可重用的类和方法而不必过分担心名字的冲突问题。

1.3 C++ 程序的基本组成单元

本节介绍 C++ 程序的基本语法结构框架。

1.3.1 一个简单的 C++ 程序

例 1.1 输入圆的半径 r , 计算圆的面积 a 和周长 c 。

按照结构化程序设计方法,程序编制如下:

```
#include <iostream>
using namespace std;
const double PI = 3.14159; //定义符号常量
int main() //函数体开始
{
    double r,a,c; //r 为半径;a 为面积;c 为周长
    cout << "请输入半径的值:"; //程序运行后的提示信息
    cin >> r; //键盘输入半径 r 的值
    a = PI * r * r; //计算面积 a
    c = 2 * PI * r; //计算周长 c
    cout << "r=" << r << endl;
    cout << "a=" << a << ",c=" << c << endl; //输出 r,a,c
} //函数体结束
```

程序只有一个名为 `main()` 的函数。C++ 规定,每个程序都有一个且只能有一个固定名称为 `main()` 的函数。ISO 标准 C++ 规定,`main()` 前必须加类型说明符 `int`,但目前运行在 Windows 平台上的编译系统也允许加类型说明符 `void`,因此,本书使用的程序例题既用 `int` 也用 `void`。`main()` 后面的一对花括号分别表示函数的开始和结束,花括号内有若干条语句,称为函数体,是函数的执行部分。

程序的第 1 行是 `include` 命令,用来包含 C++ 编译程序提供的输入输出流类库的头文件 `iostream`,以便能使用该库中的输入输出函数,为程序提供原始数据及输出程序的运行结果。

第 2 行是 C++ 新标准所必需的。与 C 语言及 C++ 早期版本的库函数不同,新标准规定头文件 `iostream` 不再使用后缀 `.h`。由于 C++ 标准程序库中的所有标识符都被定义于一个名为 `std` 的 `namespace`(名字空间)中,使用“`using namespace std;`”可以全程识别标识符 `cout` 和 `cin`。否则,每用到 `cout` 或 `cin` 时,都要写成“`std::cout`”或“`std::cin`”的形式。

第 3 行定义了一个符号常量 `PI`,其值为 3.14159,代表圆周率。

第 6 行是定义三个双精度类型的变量,依次存放圆的半径、面积和周长。

第 7 行是输出流 `cout`,表示在屏幕上输出,其后的“`<<`”是插入运算符,它与 `cout` 联用,将其后面的字符串“请输入半径的值”照原样显示在屏幕上。

第 8 行是输入流 `cin`,表示接收用户从键盘输入的数据,并保存在变量 `r` 中。其中,“`>>`”是提取运算符,它与 `cin` 联用,将从键盘输入的数据赋给变量 `r`。

第 9 行和第 10 行是两条赋值语句,用来计算圆的面积和周长,并分别存放在变量 `a` 和 `c` 中。

第 11 和 12 行是将变量 `r`、`a`、`c` 值显示在屏幕上。各行后面的“`//`”表示注释,其后的文字是注释的内容,提供用户阅读程序时的说明。

程序运行后,用户会看到屏幕上显示提示信息“请输入半径的值:”,若从键盘输入 2.5,并回车,程序自动计算面积 `a` 和周长 `c`,并在屏幕上显示 `r`、`a`、`c` 的值为:

```
请输入半径的值: 2.5
a=20.5
c=19.6349,c=15.708
```

例 1.2 用面向对象的方法重新编制例 1.1 的程序。

```
#include <iostream>
using namespace std;
const double PI = 3.14159;

class Circle
{
public:
    double areacal() { return PI * r * r; } //计算面积
    double lengthcal() { return 2 * PI * r; } //计算周长
    void getr() { cout<<"请输入半径的值:"; cin>>r; } //输入半径
    void putr() { cout<<"r=" << r << endl; } //输出半径
private:
    double r; //半径
};

int main()
{
    Circle A; //定义对象
    double a, c;
    A.getr(); //键盘输入半径
    a = A.areacal();
    c = A.lengthcal();
    A.putr(); //输出半径 r
    cout<<"a=" << a << ", " << "c=" << c << endl; //输出面积 a, 周长 c
} //函数体结束
```

程序定义了一个名为 Circle 的类,它把数据(圆半径)和对数据的操作(计算圆的面积和周长,输入和输出圆的半径)封装在一起。main()函数中定义了一个名为 A 的 Circle 类对象,通过键盘输入圆的半径,并将计算得到的面积和周长分别保存在变量 a 和 c 中。

程序的第 1 行到第 3 行与例 1.1 相同。

第 4 行的 class 是关键字,用来定义一个名为 Circle 的类。从下一行的左花括号开始直到第 13 行的右花括号结束都是 Circle 的类体。其中,数据成员为 double 型的变量 r,属于私有成员;四个成员函数 areacal、lengthcal、putr 和 getr 分别用来计算圆的面积、周长,以及输入输出半径的值(为简单计,未显式定义构造函数和析构函数)。

第 16 行是用类 Circle 定义一个名为 A 的对象(也可以看成是 Circle 型的变量)。

第 17 行定义两个 double 型的变量,分别存放圆的面积和周长。

第 18 行用对象 A 的成员函数 getr()输入圆的半径。

第 19 行和 20 行分别用对象 A 的成员函数 areacal()和 lengthcal()计算圆的面积和周长。

第 21 行和 22 行输出半径 r、面积 a 和周长 c 的值。

程序运行时,当从键盘输入 2.5 并按回车键后,运行结果为:

```
请输入半径的值: 2.5
r=2.5
a=19.6349, c=15.708
```

1.3.2 C++ 源程序的基本组成单元

从上面两个例子可以看出: C++ 兼容 C, 既能编制面向过程的结构化程序(程序是由一个 main() 函数和 0 或多个其他函数组成的), 也能编制面向对象的程序(程序是由一个 main() 函数和若干个其他函数和类组成的)。函数和类具有基本相同的结构形式: 一个函数或类由若干个 C 语句组成, 一个 C 语句由若干个基本单词组成。

1) 函数

函数是完成某个整体功能的最小单位, 它是相对独立的模块。

函数的结构形式如下:

(1) 函数名 主函数有固定的名称 main(), 其他函数则可以根据标识符的命名方法(下面即将介绍)任意取名。主函数通常包括整个程序的轮廓, 由它再调用其他函数和定义类对象。

(2) 形式参数 在函数名的后面有一对圆括号, 其中放置一个或多个形式参数, 简称形参、虚参或哑元。一个函数也可以没有形式参数, 但圆括号不能省略。

(3) 函数体 用花括号包围起来的部分是函数体, 即函数的主体。它主要由两大部分组成: 第一部分是本函数内部用到的局部变量类型定义; 第二部分是语句序列, 完成本函数的功能。

2) 类

类是某些数据及其操作的封装体, 代表一个事物, 一个人群、一个现象或一个过程。类的结构形式如下。

(1) class 类名 用来定义一个类。

(2) 类体 即在类名后用花括号括住的部分, 通常包含该类的数据(称为数据成员)和对这些数据进行的操作(称为成员函数)。有些学者建议这样找“类”: 分析一个句子的语法, 找出名词和动词, 名词就是类, 名词的属性就是其数据成员, 动词则是其方法(即成员函数)。

3) 语句

C++ 语句是完成某种程序功能(如赋值、输入、输出等)的最小单位, 所有的 C++ 语句都以分号结尾。C++ 语句可分为表达式语句、复合语句和空语句三类。

① 任何 C++ 表达式末尾加上分号后, 就构成一条表达式语句, 如

```
i = 0;
x = x + 1;
cin << r;
```

等。

② 一组 C++ 语句用花括号括起来, 就构成复合语句, 如

```
while(i < 10)
{
    sum = sum + i;
    i++;
}
```

复合语句被视为一个整体, 通常用在条件分支或循环语句中。有时为了隐藏数据, 用复合语句

形成一个代码块,块中定义的局部变量不会对程序的其他部分发生副作用。

③只有一个分号的语句称为空语句,如

```
for(i=0;i<1 000; i++)
```

```
;
```

由一条 for 语句(循环语句)和一条空语句组成。空语句用作循环语句的循环体,表示什么也不做。事实上,这个循环的功能是延迟一小段时间。有时,空语句被用作转向点。

4)基本单词

一个 C++ 语句由若干个基本单词组成。C++ 语言共有五种基本单词,即关键字(亦称保留字)、标识符、常数、操作符和分隔符。例如语句

```
float r,a,c;
```

中, float 是关键字(代表数据类型), r、a 和 c 是标识符(表示变量);又如语句

```
a=PI*r*r,c=2*PI*r;
```

中,“=”、“*”和“,”是操作符,“2”是常数,PI 是符号常量。

①关键字是 C++ 语言中有特定意义和用途、不得作为他用的字符序列,常用的 C++ 关键字见表 1-1。

表 1-1 C++ 常用关键字

流程控制	break	case	catch	continue	default	do	else	for	goto	if	return
	switch	throw	try	while							
存储属性	auto	extern	static	register							
数据类型	bitand	bitor	bool	char	class	compl	const	double	enum	float	int
	long	short	signed	struct	union	unsigned	void	volatile			
运算变换及其他	_ cast	and _ eq	const _ cast	dynamic _ cast	delete	explicit	false	friend	inline		
	mutable	namespace	new	not _ eq	operator	private	protected	public	reinterpret _		
	cast	sizeof	static _ cast	this	template	true	typedef	typename	typed	using	
	virtual	wchar t	xor eq								

C++ 关键字必须小写。

②标识符用来定义变量名、数组名、函数名、指针名、结构名、符号常量名、类名、对象名、用户定义的数据类型名及语句标号等用途的字符序列,可由 1~32 个字符组成,第一个字符必须是字母或下画线,后面的字符可以是字母、数字或下画线。例如 AB、Ab、aB、ab、A _ b、_ ab、ab _ s2d、W _ length 等都是正确的标识符,而 A + B、A'B、A. B、2abc、α、β、d% 等是错误的标识符。

标识符不能与 C++ 关键字相同,而且区分大小写。例如 AB 和 Ab 是两个不同的标识符,ELSE 可以作为标识符,它不同于 C++ 关键字 else。

③常数包括数值常数(如 123、-23.5、1.2E4 等)、字符常数(如'a'、'B'、'c'等)、字符串常数(如“xyz”、“good morning”等)、逻辑常数(true、false)、符号常量以及枚举常数。

④操作符包括各种运算符(如 +、-、*、/等)和有特定意义的标点符号(如花括号、方括号、圆括号、逗号)等。

⑤分隔符用来分隔相邻的标识符、关键字和常数,最常用的分隔符是空格,此外还可以用制表符、换行符、换页符等作为分隔符,统称为空白字符。

1.3.3 C++ 程序的书写风格

1) 函数和类

每个函数和类可以出现在整个程序文件中的任何位置,main()函数不一定出现在程序的开始处,但不管它位于程序中的何处,程序运行时总是从 main()函数开始。

2) 程序行

每个程序行中的语句数量任意,既允许一行内写几条语句,也允许一条语句分几行书写,但每条语句都必须以分号“;”结束。有时还可以在程序的适当地方(如两个函数之间)加进一个或多个空行,使程序结构更加清晰。

3) 注释

注释不是 C++ 语句,对程序的编译和运行没有影响,使用注释的唯一目的是增加程序的可读性。在编译程序时,注释被当空格处理。在 C++ 中,注释有两种形式。一种是以“/*”开始,以“*/”结束(书写时,“/”和“*”之间不能有空格),在“/*”和“*/”之间的字符均视为注释。这种形式通常用于需要跨行的注释。另一种是以“//”开始,“//”后面的字符直到行末均视为注释。这种形式通常用于不跨行的注释。

注释可以出现在程序的任何位置,既可以出现在某语句的开头或结尾处,也可以独占一行或几行。

4) 书写格式

尽管 C++ 对程序书写几乎没有限制,但为使程序清晰易读一般这样写:通常每行写一条语句;不同结构层次的语句从不同的位置开始,即按缩进格式书写成阶梯形状,可以用 Tab 键或空格键调整各行的起始位置。

1.4 简单的 C++ 程序设计

一个程序通常具备三个功能,即输入数据、数据运算和输出结果。在 C++ 中,数据运算主要是由赋值语句完成的,数据的输入、输出则需要调用 C++ 编译系统提供的输入/输出函数。本节简要介绍如何使用赋值语句和格式输入、输出函数,以便能进行最简单的程序设计。

1.4.1 赋值语句的简单使用

赋值语句具有计算和存储两大功能,一般格式为:

`v = e;`

其中,v 是变量名,e 是表达式。例如,语句 `x = a + b` 的功能是先计算表达式 `a + b` 的值,并把该值转换成和 x 相同的数据类型后保存在变量 x 中。

1.4.2 输入输出的实现

输入指的是通过输入设备将原始数据送入内存,以便计算机对它们进行处理;输出指的是将保存在内存中的计算结果送到输出设备上,以便阅读。为了不依赖具体的计算机硬件,C++ 不提供输入输出语句。在 C++ 程序设计中,C++ 编译系统除了提供与 C 兼容的标准输入输出函数库外,还提供面向对象的 I/O 流类库。因此,用户既可以使用 `studio.h` 中定义的 I/O

函数,也可以使用 I/O 流类库的输入/输出软件包实现输入、输出和文件操作。其中,I/O 流类库中的标题文件 `iostream` 定义了 4 个流对象供用户的程序进行标准输入和标准输出之用,它们分别是:

`cin` 用来处理标准输入,即键盘输入
`cout` 用来处理标准输出,即屏幕输出
`cerr` 用来处理标准错误信息,并在屏幕输出
`clog` 用来处理打印机输出

由于 C++ 具有运算符重载能力,使用比 ANSI C 的 I/O 函数更简单,因此,本书主要使用 C++ 的标准输出和标准输入。

1. 标准输出

最简单的屏幕输出是将预定义的插入运算符“<<”作用于流对象 `cout`,将“<<”后面表达式的值显示在屏幕上。例如

```
//This is a C++ program
#include <iostream>
using namespace std;
int main()
{ cout << "Hello World!" << endl; }
```

程序运行结果为

Hello World!

程序分析如下。

①在使用标准输出流时,要用 `# include` 命令将标题文件 `iostream` 包含进来。以“#”开头的 `include` 是编译预处理命令,行末不加“;”。它告诉编译系统在编译本程序时把系统提供的头文件 `iostream` 的内容插入到第二行的位置。标题文件 `iostream` 要用尖括号或双引号括住,即 `<iostream>` 或“`iostream`”。如前所述,由于 C++ 新标准程序库中的所有标识符都被定义于一个名为 `std` 的 namespace 中,使用“`using namespace std;`”可以全程识别 `cout`、`cin` 等标识符。

②插入运算符“<<”作用于流对象 `cout`,其作用是将“<<”后面的字符串输出到屏幕上。

③可使用多个插入运算符实现多个输出。实际输出时,按从左到右的顺序依次输出各个“<<”后面的值。例如

```
cout << 2 << " + " << 3;
```

输出为:2+3

④若“<<”后面是一个表达式,则先计算表达式的值,再输出。例如:

```
cout << 2 << " + " << 3 << " = " << 2 + 3;
```

输出为:2+3=5

⑤若“<<”后面是 `endl` 或“`\n`”,则表示回车换行,`endl` 与“`\n`”可以互换使用。例如:

```
cout << 2 << " + " << 3 << " = " << 2 + 3 << endl << 6 << " - " << 3 << " = " << 6 - 3;
```

将分成两行输出,即

2+3=5

6-3=3

⑥`cout` 可以输出整型、浮点型、双精度型、字符型和字符串型数据。