

21世纪高等学校计算机规划教材

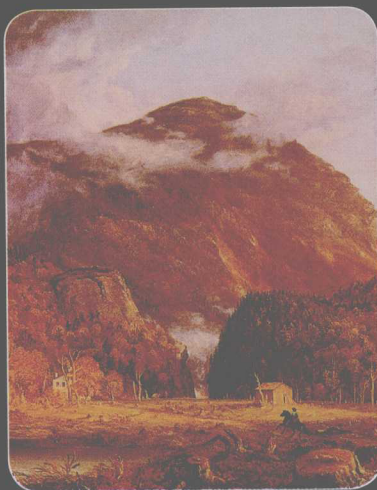
21st Century University Planned Textbooks of Computer Science

C++教程

C++ Programming

郑莉 李宁 编著

- 问题驱动案例演示
- 举一反三启发思考
- 老师好教学生易学



名家系列

 人民邮电出版社
POSTS & TELECOM PRESS

21世纪高等学校计算机规划教材

21st Century University Planned Textbooks of Computer Science

C++教程

C++ Programming

郑莉 李宁 编著



名家系列

人民邮电出版社

北京

图书在版编目 (C I P) 数据

C++教程 / 郑莉, 李宁编著. — 北京: 人民邮电出版社, 2010.5
21世纪高等学校计算机规划教材
ISBN 978-7-115-22664-8

I. ①C… II. ①郑… ②李… 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054622号

内 容 提 要

本书的内容涵盖了 C++语言的基本语法、面向对象的概念和程序设计方法、数据结构基础、模板和泛型程序设计简介。对于每一个知识点,都是先给出一个简要的综述,然后通过例题来讲解。全书一共有 174 个例题,每个例题包括:题目、主要知识点、源程序、运行结果、思考与提示几个部分。

本书是面向没有程序设计基础的读者编写的入门教材,特点是问题驱动、案例教学,适用于大学的 C++ 程序设计课程,也可用于自学。

21 世纪高等学校计算机规划教材

C++教程

-
- ◆ 编 著 郑 莉 李 宁
责任编辑 武恩玉
 - ◆ 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 电子函件 315@ptpress.com.cn
网址 <http://www.ptpress.com.cn>
北京隆昌伟业印刷有限公司印刷
 - ◆ 开本: 787×1092 1/16
印张: 22.75
字数: 598 千字
印数: 1—3 000 册
- 2010 年 5 月第 1 版
2010 年 5 月北京第 1 次印刷

ISBN 978-7-115-22664-8

定价: 36.00 元

读者服务热线: (010)67170985 印装质量热线: (010)67129223
反盗版热线: (010)67171154

计算机科学与技术日新月异的发展,对我国高校计算机人才的培养提出了更高的要求。许多高校主动研究和调整学科内部结构、人才培养目标,提高学科水平和教学质量,精炼教学内容,拓宽专业基础,优化课程结构,改进教学方法,逐步形成了“基础课程精深,专业课程宽新”的良性格局。作为大学计算机教材建设的生力军,人民邮电出版社始终坚持服务高校教学、致力教育资源建设的出版理念,在总结前期教材建设成功经验的同时,深入调研和分析课程体系,并充分结合我国高校计算机教育现状和改革成果,推出“推介名师好书,共享教育资源”的教材建设项目,出版了“21 世纪高等学校计算机规划教材”名家系列。

本套教材的突出特点如下:

(1) 作者权威 本套教材的作者均为国内计算机学科中的学术泰斗或高校教学一线的教学名师,他们有着深厚的科研功底和丰富的教学经验。可以说,这套教材汇聚了众师之精华,充分显示了这套教材的格调和品位。无论是刚入杏坛的年轻教师,还是象牙塔内的莘莘学子,细细品读其中的章节文字,定会受益匪浅。

(2) 定位准确 本套教材是为普通高等院校的学生量身定做的精品教材。具体体现在:一是本套教材的作者长期从事一线科研和教学工作,对高校教学有着深刻而独到的见解;二是本套教材在选题策划阶段便多次召开调研会,对普通高校的教学需求和教材建设情况进行充分摸底,从而保证教材在内容组织和结构安排上更加贴近实际教学;三是组织有关作者到较为典型的普通高等院校讲授课程教学方法,深入了解教师的教学需求,充分把握学生的理解能力,以教材内容引导授课教师严格按照科学方法实施教学。

(3) 教材内容与时俱进 本套教材在充分吸收国内外最新计算机教学理念和教育体系的同时,更加注重基础理论、基本知识和基本技能的培养,集思想性、科学性、启发性、先进性和适应性于一身。

(4) 一纲多本,合理配套 根据不同的教学法,同一门课程可以有多本不同的教材,教材内容各具特色,实现教材系列资源配套。

总之,本套教材中的每一本精品教材都切实体现了各位教学名师的教学水平,充分折射出名师的教学思想,淋漓尽致地表达着名师的教学风格。我们相信,这套教材的出版发行一定能够启发年轻教师们真正领悟教学精髓,教会学生科学地掌握计算机专业的基本理论和知识,并通过实践深化对理论的理解,学以致用。

我们相信,这套教材的策划和出版,无论在形式上还是在内容上都能够显著提高我国高校计算机专业教材的整体水平,为培养符合时代发展要求的具有较强国际竞争力的高素质创新型计算机人才,为我国普通高等教育的计算机教材建设工作做出新的贡献。欢迎各位老师和读者给我们的工作提出宝贵意见。

C++语言是目前广泛使用的一种程序设计语言。它支持面向过程、面向对象和泛型程序设计，主要应用于面向对象的程序设计。C++语言课程是在大学里普遍开设的程序设计基础类课程。学习 C++的途径大致有两种，一种是首先学习一种面向过程的语言（通常是 C 语言），熟悉程序设计的基础知识，然后再学习 C++；另一种是，直接以 C++作为程序设计的入门课程。这两种途径各有优点。

本书是面向没有程序设计基础的读者编写的入门教材，适用于大学的 C++程序设计课程，也可用于自学。

从 1999 年开始，我在清华大学讲授 C++课程，以前也出版过 C++教材，还算是受欢迎的书。这些年来与各个学校的同行有过很多交流、探讨。除了我自己教的学生，还收到过大量读者的来信。师生们在教学过程中，对教材有着多种不同的需求。其中有一些教师，会抱怨讲大课的时候学生坐不下、听不进；有一些学生会抱怨老师讲得没意思、听不懂。虽然我自己授课的时候没遇到这样的情况，但是这样的情况确实是存在的。我理解不同学校的学生，学习习惯和学习兴趣可能有所差别，所以教材的编写思路和讲述形式也需要多样化，以适应不同的需求。

本书的内容涵盖了 C++语言的基本语法、面向对象的概念和程序设计方法、数据结构基础、模板和泛型程序设计简介。对于每一个知识点，都是先给出一个简要的综述，然后通过例题来讲解。全书一共有 174 个例题，每个例题包括：题目、主要知识点、源程序、运行结果、思考与提示几个部分。书中的习题主要是选择题和填空题。这些习题中包含了很多“国家计算机等级考试二级（C++）笔试”的真题，对于检验学生的学习情况和准备相关的考试都有裨益。学生编程能力的训练主要是通过实验来完成的，本书另外配有实验指导书。

我写这本书的初衷，就是想给师生们提供一本容易教、容易学的教材：没有长篇大论的理论，以问题驱动，将概念和语法都融在例题之中；还有很多知识点是以启发式的方法，出现在例题的“思考与练习”中。这样，老师讲的时候就可以很生动，先提出问题，再介绍解决这个问题需要的概念和语法，然后讲解源程序、演示运行结果，最后对运行时的现象进行分析，巩固概念和语法，并对学生的进一步思考和练习给出提示；也可以针对“思考与练习”中的问题组织课内外的讨论。

与本书配套的资源如下：

- 电子版的教师资料：包括 PPT 格式的讲稿、全部例题源代码、“思考与练习”参考答案、实验参考程序。
- 习题解答与实验指导：包括全部习题的答案、每章一个实验（实验任务和步骤提示）。

● 例题源代码：放在人民邮电出版社的教学服务与资源网（<http://www.ptpedu.com.cn>）上，所有读者都可以下载。

本书第 1 章由郑莉编写，第 2 章、第 3 章、第 5 章、第 11 章、第 12 章由郑莉、郭志鹏共同编写，第 4 章、第 10 章、第 13 章、第 15 章由郑莉、邹志文共同编写，第 6 章~第 9 章、第 14 章由李宁编写。郑辑涛调试了全书的例题，并对全书提出了很多修改建议，高原编写了全书的习题。

感谢读者选用本书，欢迎使用本书的教师和学生提出宝贵意见，作者将不胜感激。作者联系方式：zhengli@tsinghua.edu.cn。

作者 郑莉

2010 年 3 月于清华园

目 录

第 1 章 绪论	1	第 3 章 流程控制	30
1.1 C++语言的特点	1	3.1 分支结构	30
1.2 面向对象程序设计	1	3.1.1 用 if 语句实现选择结构	30
1.3 C++语言的基本符号	2	3.1.2 switch 语句	34
1.4 C++语言的词汇	3	3.2 循环语句	36
1.4.1 关键字	3	3.2.1 while 语句	36
1.4.2 标识符	3	3.2.2 do while 语句	37
1.4.3 常量	3	3.2.3 for 语句	38
1.4.4 运算符	3	3.3 循环与选择结构的嵌套	40
1.5 最简单的 C++程序	4	3.4 break & continue 语句	41
小结	5	3.5 goto 语句	43
习题	5	小结	44
第 2 章 简单数据及运算	7	习题	44
2.1 程序中的数据	7	第 4 章 复杂数据及运算	47
2.2 基本数据类型	11	4.1 数组	47
2.2.1 整数类型	11	4.1.1 一维数组的声明与使用	48
2.2.2 浮点数类型	15	4.1.2 多维数组的声明与使用	50
2.2.3 字符类型	16	4.2 指针	52
2.2.4 字符串类型	17	4.2.1 数据在内存中的地址	53
2.2.5 布尔类型	18	4.2.2 指针及指针运算	56
2.3 简单运算	19	4.3 字符串	61
2.3.1 算术运算	19	4.3.1 以数组存放 C 风格的字符串	61
2.3.2 赋值运算	21	4.3.2 string 类型字符串	63
2.3.3 逗号运算	23	4.4 枚举类型	67
2.3.4 关系运算	23	4.5 运算优先级与类型转换	68
2.3.5 逻辑运算	24	4.5.1 运算优先级	68
2.3.6 sizeof 运算	25	4.5.2 类型转换	70
2.3.7 位运算	26	小结	71
2.4 语句	27	习题	71
2.4.1 声明语句	28	第 5 章 抽象与分类	74
2.4.2 表达式语句	28	5.1 面向过程的功能抽象——函数	74
2.4.3 复合语句	28	5.1.1 函数的定义与调用	74
小结	28	5.1.2 函数参数的单向传递	77
习题	28		

5.1.3 引用类型的参数	78	6.3.5 重载赋值运算符	138
5.1.4 指针类型的参数	81	6.3.6 重载类型转换符	142
5.1.5 数组类型的参数	81	6.3.7 重载下标访问运算符	144
5.1.6 缺省参数值	84	小结	145
5.1.7 省略参数	85	习题	145
5.1.8 嵌套与递归调用	85		
5.1.9 内联函数	87	第 7 章 多文件工程和编译预处理命令	150
5.1.10 函数指针	88		
5.1.11 函数的重载	89	7.1 多文件工程	150
5.2 类与对象的概念	91	7.2 外部变量与外部函数	152
5.2.1 分类与抽象的思想	91	7.3 标准 C++库和命名空间	154
5.2.2 对象的属性与行为	91	7.3.1 标准 C++库	154
5.3 程序中的类的设计	91	7.3.2 什么是命名空间	155
5.3.1 对象属性的抽象	92	7.3.3 命名空间的定义	155
5.3.2 对象功能与行为的抽象	93	7.3.4 命名空间的使用	157
5.4 类的对象	94	7.3.5 命名空间的嵌套	161
5.4.1 对象的定义与初始化	94	7.4 编译预处理	162
5.4.2 对象的析构	99	7.4.1 在程序中包含头文件	163
5.4.3 对象数组	100	7.4.2 程序中的置换	163
5.4.4 通过指针访问对象	103	7.4.3 条件预编译	165
5.4.5 以对象作为函数的参数	106	小结	168
5.5 特殊的类	106	习题	169
5.5.1 结构体	107		
5.5.2 联合体	109	第 8 章 类的重用	171
5.6 友元	111		
5.6.1 友元函数	111	8.1 类的组合	171
5.6.2 友元类	112	8.1.1 类的组合的含义	171
小结	113	8.1.2 类的组合实现	172
习题	113	8.1.3 前向引用声明	175
		8.2 类的继承	176
第 6 章 自定义类型的运算	121	8.2.1 什么是继承	176
6.1 用函数实现对象的运算	121	8.2.2 派生类的定义	177
6.2 运算符重载函数	124	8.2.3 继承成员的访问控制	179
6.2.1 运算符重载为成员函数	125	8.2.4 派生类的构造函数和析构函数	182
6.2.2 运算符重载为非成员函数	127	8.2.5 向上转型	185
6.3 典型运算符的重载	129	8.3 多继承与虚基类	187
6.3.1 重载乘法运算符	129	8.3.1 多继承中同名隐藏和二义性问题	187
6.3.2 重载求相反数运算符	131	8.3.2 重复继承与虚基类	188
6.3.3 重载相等比较运算符	133	小结	190
6.3.4 重载 ++ 运算符	135	习题	191

第 9 章 数据的共享与保护195	11.3.3 格式控制操纵符.....238
9.1 变量的生存期与作用域.....195	11.4 I/O 流的状态.....248
9.1.1 对象的生存期.....195	11.5 文件流.....249
9.1.2 作用域及可见性.....196	11.5.1 文件的概念.....250
9.2 同类对象间的数据共享——静态成员.....199	11.5.2 文件流的建立.....250
9.2.1 静态数据成员.....199	11.5.3 文件流的关闭.....251
9.2.2 静态函数成员.....200	11.5.4 两种特殊的文件流.....251
9.3 共享数据的保护.....202	11.6 字符串流.....253
9.3.1 常引用.....202	11.6.1 字符串流的建立.....253
9.3.2 常成员.....203	11.6.2 字符串流的缓冲区操作.....254
9.3.3 常对象.....206	11.7 针对自定义类型重载>>和<<.....255
小结.....208	小结.....257
习题.....208	习题.....257
第 10 章 多态性213	第 12 章 容错及异常处理基础260
10.1 多态性概述.....213	12.1 异常处理的基本思想.....260
10.2 虚函数.....214	12.2 抛出异常.....261
10.2.1 函数的覆盖.....214	12.3 捕获异常.....263
10.2.2 虚函数成员.....215	12.4 异常对象.....266
10.2.3 虚析构造函数.....219	12.5 标准程序库异常处理.....267
10.3 抽象类.....221	12.6 异常规范.....269
小结.....223	小结.....270
习题.....223	习题.....270
第 11 章 输入/输出流230	第 13 章 数据结构简介273
11.1 C++输入/输出流的概念.....230	13.1 线性群体.....273
11.1.1 什么是输入/输出流.....230	13.1.1 线性群体的概念.....273
11.1.2 流类库的头文件.....231	13.1.2 直接存取群体——数组.....274
11.1.3 预定义流对象.....231	13.1.3 顺序存取群体——链表.....277
11.1.4 I/O 流库的类体系.....231	13.1.4 栈.....286
11.1.5 输出流举例.....232	13.1.5 队列.....288
11.1.6 输入流举例.....233	13.2 群体数据的组织.....290
11.2 无格式的输入/输出.....234	13.2.1 顺序查找.....290
11.2.1 无格式输入.....235	13.2.2 折半查找.....291
11.2.2 无格式输出.....236	13.2.3 插入排序.....293
11.3 有格式输入/输出.....236	13.2.4 选择排序.....294
11.3.1 默认的输入/输出格式.....236	13.2.5 交换排序.....295
11.3.2 格式标志的设置.....237	小结.....297
	习题.....297

第 14 章 模板简介	301	15.2.1 顺序容器.....	322
14.1 函数模板.....	301	15.2.2 关联容器.....	333
14.1.1 函数模板的定义和实例化.....	301	15.2.3 容器适配器.....	335
14.1.2 模板实参的使用.....	304	15.3 迭代器.....	338
14.1.3 模板的常规参数.....	306	15.3.1 迭代器的分类.....	339
14.2 类模板.....	307	15.3.2 迭代器适配器.....	341
14.3 类模板与继承.....	310	15.3.3 迭代器相关的辅助函数.....	341
小结.....	312	15.4 标准 C++库中的算法简介.....	342
习题.....	312	15.4.1 非可变序列算法.....	343
第 15 章 泛型程序设计与 C++ STL		15.4.2 可变序列算法.....	345
简介	319	15.4.3 排序及相关算法.....	348
15.1 泛型程序设计的概念和术语.....	319	15.4.4 数值算法.....	350
15.1.1 泛型程序设计.....	319	15.5 函数对象.....	351
15.1.2 STL 的相关概念和术语.....	320	15.5.1 函数对象.....	351
15.2 C++ STL 中的容器.....	322	15.5.2 函数适配器.....	351
		小结.....	352
		习题.....	352

第 1 章

绪 论

C++语言是目前使用最为广泛的程序设计语言。它与 C 语言完全兼容，既保留了传统的结构化程序设计方法，又对面向对象程序设计方法提供了完整的支持。本章简要介绍面向对象思想和 C++语言的基本特征。

学习目标

- 掌握 C++语言的基本符号和词汇；
- 掌握 C++程序的两种基本框架；
- 初步了解面向对象的思想。

1.1 C++语言的特点

在众多的高级程序设计语言中，C++能够取得成功的原因在于它有着许多与众不同的特点。

1. C++是一种面向对象的程序设计语言

C++支持几乎所有的面向对象程序设计特征。

2. C++是更好的 C 语言

C++中包含 C 语言的全部语法特征。因此，每一个用 C 语言编写的程序都是一个 C++程序。

C++的设计宗旨就是在不改变 C 语言语法规则的基础上扩充新的特性。

实际上，能够很好地兼容 C 语言正是 C++取得成功的原因之一，这是因为：

- C++继承了 C 语言简明、高效、灵活等众多优点。
- 以前使用 C 语言编写的大批软件可以不加任何修改，直接在 C++开发环境下维护。
- C 语言程序员只需要学习 C++扩充的新特性，就可以很快地使用 C++编写程序。

由于具有上述特点，C++已经开始取代 C 语言，被广泛地应用于各种领域的程序设计工作中。实践表明，对于中型和大型程序的开发工作，使用 C++的效果要比 C 语言好得多。C++正在从软件的可靠性、可重用性、可扩充性、可维护性等方面体现出它的优越性。

1.2 面向对象程序设计

面向对象程序设计（Object-Oriented Programming，OOP）是 20 世纪 80 年代发展起来的一种

程序设计方法，它通过模拟现实世界中的事物和关系，利用抽象、分类、归纳等方法来构造软件系统。

在面向对象程序设计出现之前，人们一直采用结构化程序设计（Structured Programming, SP）来解决实际问题。结构化程序设计是面向过程的，其主要思想是将功能分解并逐步求精。Pascal 语言和 C 语言都很好地体现了结构化程序设计的思想。

C++是一种面向对象的程序设计语言，它充分支持面向对象思想中的 3 个主要特征。

1. 封装性

封装性是指将数据和处理数据的方法封装在一起成为一个类。类的实体就是对象，对象一旦定义，就可看成是一个完全封装的实体，可以作为一个整体单元来使用。类的内部数据表示被隐藏起来，类的用户不需要知道类内数据的表示方法，只需执行类对外提供的算法，就可以完成某项功能。

2. 继承性

继承性是指一种事物保留了另一种事物的全部特征，并且具有自身的独有特征。C++采用继承来支持重用，程序可以在现有类型的基础上扩展功能来定义新类型。新类型是从现有类型中派生出来的，因此被称为派生类。

3. 多态性

多态性是指当多种事物继承自一种事物时，同一种操作在它们之间表现出不同的行为。C++中使用虚函数和抽象类的概念来支持多态性。

C++主要包括面向过程和面向对象两部分内容。其中面向过程部分可以看成是功能增强的 C 语言；而面向对象部分是 C 语言中所没有的，它是 C++支持面向对象程序设计的主体。要学习面向对象程序设计，首先必须具有面向过程语言的基础。所以，学习 C++时，应该先学习其面向过程部分，再学习面向对象部分。由于，C++中几乎包含了 C 语言的全部内容，所以在学习 C++之前，不需要先学习 C 语言。

1.3 C++语言的基本符号

组成语言的最小元素是基本符号，C++语言中的基本符号可以分为 3 类。

1. 字母

包括大写英文字母

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

和小写英文字母

a b c d e f g h i j k l m n o p q r s t u v w x y z

2. 数字

0 1 2 3 4 5 6 7 8 9

3. 特殊符号

+ - * / = , . _ : ; ? \ " ' ~

| ! # % & () [] { } ^ < > 空格

这 3 类符号共计 92 个，它们组成了 C++语言的基本符号集合。

1.4 C++语言的词汇

这里我们要介绍 C++语言的常用词汇，主要包括 C++的关键字、标识符、文字常量、运算符。

1.4.1 关键字

关键字也称为保留字，它是由 C++语言本身预先定义好的一类单词。下面列出的是 C++的关键字：

auto	bool	break	case	catch	char	class
const	const_cast	continue	default	delete	do	double
dynamic_cast	else	enum	explicit	extern	false	float
for	friend	goto	if	inline	int	long
mutable	namespace	new	operator	private	protected	public
register	reinterpret_cast	return	short	signed	sizeof	static
static_cast	struct	switch	template	this	throw	true
try	typedef	typeid	typename	union	unsigned	using
virtual	void	volatile	while			

每个关键字在 C++语言中都具有特殊的含义，并实现着一定的功能。所以，不能将上述关键字再当作其他类型的单词使用。

1.4.2 标识符

标识符是程序员自定义的名称，它命名程序正文中的一些实体，其构成要符合一定的规则：

(1) 标识符是一个以字母或下划线开头的，由字母、数字、下划线组成的字符串。例如，Hello、var2、myBook、func_1、_TEST_H 都是合法的标识符；而 012、3var、A*B、\$sys、pay.due 都是非法的标识符。

(2) 标识符不能与任何一个关键字同名。例如，if、case、for 等都不能作为标识符使用。

(3) 标识符中的字母区分大小写。例如，Abc 和 abc 是两个不同的标识符。

(4) 标识符不宜过长。C++语言一般设定一个标识符的有效长度为 32 个字符。如果程序中的标识符超过了这个长度，超出的部分将被忽略不计。

1.4.3 常量

常量是指在程序运行的整个过程中其值始终不可改变的量。常量一般有两种表示形式，即符号常量和文字常量。例如，在“pi 等于 3.1416”这句话中，pi 是一个符号常量，pi 的值 3.1416 称为文字常量。C++语言中符号常量的名字就是一个标识符，而文字常量却是直接写出来的值。文字常量分为整型、浮点型、字符型和字符串型 4 类，具体的将在第 2 章中介绍。

1.4.4 运算符

运算符（操作符）是用于实现各种运算的符号。C++语言中运算符的种类非常丰富：

- 单字符组成的运算符，例如，+、-、*、/等；

- 双字符组成的运算符，例如，++、<=、&&、*=、->等；
 - 3个字符组成的运算符，例如，<<=、>>=等；
 - 关键字运算符，例如，new、delete、sizeof。
- 各种运算符的具体使用方法将在第2章中介绍。

1.5 最简单的 C++ 程序

由于 C++ 既支持结构化程序设计，又支持面向对象程序设计，所以它同时具有结构化程序设计和面向对象程序设计两种基本框架。为了便于理解，我们先来看一个最简单的 C++ 程序。

例 1-1 一个最简单的 C++ 程序。

主要知识点

主函数是所有 C++ 程序开始执行的入口。无论主函数处于程序中的什么位置，其中的代码总是最先被执行。按照 C++ 语言的规定，每个程序都必须有且仅有一个主函数，主函数的名称必须为 main。

主函数 main 应该返回一个 int 类型的值。int 代表整型值，它是 C++ 中的一个基本数据类型，第2章将详细介绍。

在 C++ 中所有的函数名称后面都紧跟着一对圆括号，其中可以没有内容（即没有参数），也可以包含函数的参数。

主函数 main 的声明之后是用大括号“{}”括起来的函数主体部分。

cout 是 C++ 中的标准输出流对象，它通常代表计算机的屏幕。cout 在标准头文件 iostream 中被声明，因此要使用它就必须先包含此头文件。<< 是输出操作符，功能是将它右边的内容输出到它左边的指定设备上。这里，要输出的内容是用双引号括起来的字符串，例如“This is the first C++ program.\n”，其中的字符“\n”是换行符。语句末尾的分号“;”表示这条语句的结束。在 C++ 中所有的语句都要以分号结束。

要使主函数 main 结束并将整数 0 返回给运行此程序的操作系统，需要写如下语句：

```
return 0;
```

返回整数 0 表示程序在执行过程中没有发生任何错误而正常结束。

实际上，按照 C++ 标准本行是可以省略的。如果在主函数 main 中不写“return 0;”语句，程序正常结束时也会自动向操作系统返回 0。因此，最短的 C++ 程序仅由 12 个字符组成，即“int main(){}”，但这个程序不实现任何功能。

源代码：

```
// 1_1.cpp
#include <iostream >
using namespace std;
int main()
{
    cout << "This is the first C++ program.\n";
    return 0;
}
```

运行结果：

```
This is the first C++ program.
```

思考与练习：

上述程序虽然只有几行，但它却包含了每一个 C++ 程序都要具备的几个基本组成部分。由于这是本书中第一个完整的 C++ 程序，我们将对它进行逐行解释：

```
// 1_1.cpp
```

这是一个注释行。注释是程序编写者对程序进行的注释和说明，有助于提高程序的可读性。注释对于程序的功能实现不起任何作用。

```
#include <iostream >
```

这是一条预处理命令。作用是将头文件 `iostream` 的内容加入到程序中。`iostream` 中定义了一些输入输出流对象，例如这里用到的 `cout`。

```
using namespace std;
```

这条语句的作用是，说明程序中将要使用 `std` 这个名空间中的名字，例如对象名 `cout`。名空间是避免名字冲突的一种机制，将在后续章节中详细介绍。

```
int main ()
```

本行是主函数的声明。这里 `main` 是主函数名，函数体用一对大括号括住。`main()` 函数之前的 `int` 表示 `main` 函数的返回值类型。

```
cout << "This is the first C++ program.\n";
```

本行完成了此程序的主要功能，即向屏幕上输出一行字符串。这里，`cout` 是一个输出流对象，包含许多有用的输出功能。“<<”是输出操作符，其作用是将紧随其后的双引号中的字符串输出到标准输出设备（显示器）上。

```
return 0;
```

本行的功能是使主函数 `main` 结束并将整数 0 返回给运行此程序的操作系统。

请读者修改此程序，尝试向屏幕输出更多的信息。

小 结

本章简单简介 C++ 语言的基本知识，旨在使读者对 C++ 程序有一个初步了解，为后续章节的学习奠定知识基础。本章的内容虽然简单，但是对于没有基础的初学者来说，仍然会有许多概念难以理解。遇到这样的情况，读者可以暂时搁置不理解的内容，继续学习后续章节，随着学习的深入和实践的增多，一些问题自然会迎刃而解的。

习 题

一、选择题

1. 2005 年上半年全国计算机等级考试二级笔试试卷 C++ 语言程序设计。

在 C++ 中，实现封装性需借助于 ()。

- A. 枚举 B. 类 C. 数组 D. 函数

2. 2005 年下半年全国计算机等级考试二级笔试试卷 C++ 语言程序设计。

下列字符串中，正确的 C++ 标识符是 ()。

- A. foo-1 B. 2b C. new D. _256

3. 2006 年 4 月全国计算机等级考试二级笔试试卷 C++ 语言程序设计。

下列符号中可以用作 C++ 标识符的是 ()。

- A. _radius B. foo~bar C. else D. 3room

4. 以下不是 C++ 关键字的是 ()。

- A. new B. char C. try D. get

二、填空题

1. C++ 是面向_____的程序设计语言。

2. C++ 语言多态性依赖于_____和_____的概念。

第 2 章

简单数据及运算

计算机程序的最基本功能，就是对计算问题以及计算方法进行准确描述，使计算机执行程序时能够进行正确、高效的运算。在计算过程中通常会用到一些数据，复杂计算中还会用到大量数据，这些数据在程序中如何表示呢？本章将介绍数据存储的基本知识和 C++ 中的数据类型。

学习目标

- 掌握常量与变量的概念及其定义和初始化方法；
- 掌握 C++ 的基本数据类型；
- 掌握基本类型数据的运算实现方法。

2.1 程序中的数据

参与运算的数据，以及运行的代码都需要首先装入内存。代码是在启动程序时被操作系统装入内存的，小规模的程序一般是将全部代码一次性装入内存，而大型程序往往是将当前需要执行的代码装入内存。程序中的数据需要在内存中被分配存储空间，一些数据是在程序运行开始时就被分配内存空间，另一些数据是在程序运行过程中根据需要动态获得内存空间。关于动态获得内存空间的问题，将在第 3 章中介绍，本章只介绍静态内存分配。

例 2-1 在程序中读入、表示并输出显示整数。

主要知识点：常量、变量、简单数据输入/输出

1. 常量与变量

程序的最基本功能是对数据进行处理，而需要处理的数据如何交给程序，程序的处理结果如何输出呢？输出信息的最简单方法我们在例 1-1 中看到了。程序得到数据的最基本途径主要有以下两种：

在源程序中直接写明要处理的数据。这些数据一旦写出，其值在整个程序运行期间就是不可改变的，这样的数据称为文字常量；

另一些数据是在运行过程中从计算机的外部设备（如键盘、硬盘）读取的，这些数据的值在程序运行过程中是允许被改变的，这样的数据称为变量。

2. 从键盘输入数据

为了能够在运行过程中读取数据，就要用到 C++ 语言提供预先编写好的功能模块，istream 类。它包含了许多输入功能。使用 istream 类的对象 cin 的 >> 操作，可以简单地实现从标准输入设备（通常是键盘）读入数据，这一点在第 9 章中将会详细介绍。