

21世纪高等学校规划教材 | 计算机科学与技术



C++语言程序设计 (MOOC版)

阚道宏 编著

清华大学出版社

21世纪高等学校规划教材 | 计算机科学与技术

C++语言程序设计

(MOOC版)

阚道宏 编著

清华大学出版社
北京

内 容 简 介

本书是为“C++语言程序设计”慕课(MOOC)在线教育课程而编写的教材。本书按应用需求来梳理和组织C++语言的知识点,其中包括结构化程序设计方法和面向对象程序设计方法。内容编排由易到难,循序渐进;每个小节都设计了适合在线评判的多选练习题,每章则设计了适合课堂讨论的程序阅读题、改错题和编程题。

在线开设“C++语言程序设计”课程的教师可将本书作为授课教材使用,联系作者可免费获得配套教学课件和视频。参加在线课程学习的学生可将本书作为线下阅读教材使用。

版权所有,侵权必究。侵权举报电话:010-62782989 13701121933

图书在版编目(CIP)数据

C++语言程序设计:MOOC版/阚道宏编著.--北京:清华大学出版社,2016

21世纪高等学校规划教材·计算机科学与技术

ISBN 978-7-302-42104-7

I. ①C… II. ①阚… 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(2015)第260842号

责任编辑:郑寅堃 薛 阳

封面设计:傅瑞学

责任校对:李建庄

责任印制:杨 艳

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:三河市中晟雅豪印务有限公司

经 销:全国新华书店

开 本:185mm×260mm 印 张:24 彩 插:1 字 数:604千字

版 次:2016年2月第1版 印 次:2016年2月第1次印刷

印 数:1~2000

定 价:49.00元

产品编号:067348-01

出版说明

随着我国改革开放的进一步深化,高等教育也得到了快速发展,各地高校紧密结合地方经济建设发展需要,科学运用市场调节机制,加大了使用信息科学等现代科学技术提升、改造传统学科专业的投入力度,通过教育改革合理调整和配置了教育资源,优化了传统学科专业,积极为地方经济建设输送人才,为我国经济社会的快速、健康和可持续发展以及高等教育自身的改革发展做出了巨大贡献。但是,高等教育质量还需要进一步提高以适应经济社会发展的需要,不少高校的专业设置和结构不尽合理,教师队伍整体素质亟待提高,人才培养模式、教学内容和方法需要进一步转变,学生的实践能力和创新精神亟待加强。

教育部一直十分重视高等教育质量工作。2007年1月,教育部下发了《关于实施高等学校本科教学质量与教学改革工程的意见》,计划实施“高等学校本科教学质量与教学改革工程”(简称“质量工程”),通过专业结构调整、课程教材建设、实践教学改革、教学团队建设等多项内容,进一步深化高等学校教学改革,提高人才培养的能力和水平,更好地满足经济社会发展对高素质人才的需要。在贯彻和落实教育部“质量工程”的过程中,各地高校发挥师资力量强、办学经验丰富、教学资源充裕等优势,对其特色专业及特色课程(群)加以规划、整理和总结,更新教学内容、改革课程体系,建设了一大批内容新、体系新、方法新、手段新的特色课程。在此基础上,经教育部相关教学指导委员会专家的指导和建议,清华大学出版社在多个领域精选各高校的特色课程,分别规划出版系列教材,以配合“质量工程”的实施,满足各高校教学质量和教学改革的需要。

为了深入贯彻落实教育部《关于加强高等学校本科教学工作,提高教学质量的若干意见》精神,紧密配合教育部已经启动的“高等学校教学质量与教学改革工程精品课程建设工作”,在有关专家、教授的倡议和有关部门的大力支持下,我们组织并成立了“清华大学出版社教材编审委员会”(以下简称“编委会”),旨在配合教育部制定精品课程教材的出版规划,讨论并实施精品课程教材的编写与出版工作。“编委会”成员皆来自全国各类高等学校教学与科研第一线的骨干教师,其中许多教师为各校相关院、系主管教学的院长或系主任。

按照教育部的要求,“编委会”一致认为,精品课程的建设工作从开始就要坚持高标准、严要求,处于一个比较高的起点上。精品课程教材应该能够反映各高校教学改革与课程建设的需要,要有特色风格、有创新性(新体系、新内容、新手段、新思路,教材的内容体系有较高的科学创新、技术创新和理念创新的含量)、先进性(对原有的学科体系有实质性的改革和发展,顺应并符合21世纪教学发展的规律,代表并引领课程发展的趋势和方向)、示范性(教材所体现的课程体系具有较广泛的辐射性和示范性)和一定的前瞻性。教材由个人申报或各校推荐(通过所在高校的“编委会”成员推荐),经“编委会”认真评审,最后由清华大学出版

社审定出版。

目前,针对计算机类和电子信息类相关专业成立了两个“编委会”,即“清华大学出版社计算机教材编审委员会”和“清华大学出版社电子信息教材编审委员会”。推出的特色精品教材包括:

(1) 21 世纪高等学校规划教材·计算机应用——高等学校各类专业,特别是非计算机专业的计算机应用类教材。

(2) 21 世纪高等学校规划教材·计算机科学与技术——高等学校计算机相关专业的教材。

(3) 21 世纪高等学校规划教材·电子信息——高等学校电子信息相关专业的教材。

(4) 21 世纪高等学校规划教材·软件工程——高等学校软件工程相关专业的教材。

(5) 21 世纪高等学校规划教材·信息管理与信息系统。

(6) 21 世纪高等学校规划教材·财经管理与应用。

(7) 21 世纪高等学校规划教材·电子商务。

(8) 21 世纪高等学校规划教材·物联网。

清华大学出版社经过三十多年的努力,在教材尤其是计算机和电子信息类专业教材出版方面树立了权威品牌,为我国的高等教育事业做出了重要贡献。清华版教材形成了技术准确、内容严谨的独特风格,这种风格将延续并反映在特色精品教材的建设中。

清华大学出版社教材编审委员会

联系人:魏江江

E-mail:weijj@tup.tsinghua.edu.cn

1. 关于慕课(MOOC)

慕课,即大规模开放在线课程(Massive Open Online Course, MOOC),是近两年兴起的一种基于互联网的新型教学模式。2012年被称为“慕课元年”。慕课教学模式实现了两个转变:以教师为中心向以学习者为中心的转变;由被动学习向主动学习的转变。与普通网络教学视频所不同的是,慕课实现了包含从授课到习题、讨论、答疑、测验,直到最终学习评价等环节的完整教学过程。为适应网络学习环境,便于学习者随时随地学习,慕课教学视频应使用短视频。与传统课堂授课所不同的是,开设慕课课程需重新梳理和组织知识点,并分别提供适合线上使用的练习题以及线下使用的讨论或实验题。

可以将线上慕课与线下课堂这两种教学模式结合起来(可称为 SPOC, Small Private Online Course)。线上慕课就是先由学生自主完成线上学习,例如观看视频、做线上习题等。线下课堂则是由教师组织课堂讨论、实验和测验,或讲解重点疑难问题。“线上慕课,线下课堂”是对现有“课上听课,课下作业”教学模式的翻转。虽然慕课教学模式尚处于起步试验阶段,但大多数网络学习者十分喜欢慕课。目前已涌现出很多知名的慕课网站,例如国外的 Coursera、Udacity 和 Edx,国内的爱课程(中国大学 MOOC)、学堂在线和雨虹学网等。越来越多的名校教师依照个人兴趣,积极在网络上发布、开设公开课。中国农业大学也正在基于“雨虹学网”积极开展课堂教学改革方面的尝试与探索。

学习“C++语言程序设计”,需要边阅读,边思考,边消化吸收。虽然有了大量的网上资源,纸质教材仍是“慢思考”阅读的首选方式,这也是作者编写出版本书的目的。

2. 本书特色

1) 适用于慕课(MOOC)在线教育课程

本书是为“C++语言程序设计”慕课(MOOC)在线教育课程而编写的教材。本书按应用需求来梳理和组织 C++ 语言的知识点,其中包括结构化程序设计方法和面向对象程序设计方法。内容编排由易到难,循序渐进。每个小节都设计了适合在线评判的多选练习题,每章则设计了适合课堂讨论的程序阅读题、改错题和编程题。

2) 采用案例教学

每个知识点都从精心设计的任务需求开始导入,然后提出对应的实现方法,最后系统地阐述其语法细则。既保证了知识体系的完整性,又能让读者直观理解抽象的概念和原理。

3) 创新教学方法

本书在 C++ 语言教学方面有三个创新点。一是强化初学者对“程序由计算机硬件执行”这一基本概念的认知,从有形的硬件来理解相对抽象的程序,这样各种语法概念就不再那么

抽象了。二是明确提出程序设计过程中存在不同的角色,并充分利用角色来引导读者理解语法的应用语境。三是明确提出必须从代码分类管理、数据类型、归纳抽象和代码重用等多个维度才能准确理解面向对象程序设计方法。教学实践表明上述教学方法可降低学习难度,提高学生学习的主动性。

3. 内容摘要

第1章 程序设计导论。从初学者对计算机已有的认知开始,将初学者逐步引导到计算机程序的世界。本章首先介绍计算机、指令及机器语言和程序等基本概念,然后描述程序与计算机硬件、程序员以及用户之间的关系,让读者在一开始就能明确程序员的职责,实现从用户到程序员的角色转换。本章内容按知识点顺序,依次讲解计算机硬件结构、计算机程序及其开发过程、信息分类与数据类型和 C++ 语言简介等。**本章要点:**一是让读者从有形的硬件来理解相对抽象的软件;二是让读者认识到计算机中的数据是有类型的,类型决定了数据在计算机中的存储位数和存储格式;三是让读者知道,学习程序设计和学习编程语言不是一回事。和 C 语言、Java 语言相比,C++ 语言的知识体系最系统、最全面,因而本书选用 C++ 语言作为程序设计初学者的入门语言。

第2章 数值计算。本章从最简单的数值计算问题开始,以案例教学的方法让读者领会程序设计中一些最基础、最重要的基本概念,其中包括程序中的变量和常量、表达式与运算符、数据的输入和输出等。最后介绍 C++ 程序访问内存的三种方式,它们分别是变量名、引用和指针。**本章要点:**一是让读者将程序中的数据与内存联系起来,这样就很容易理解数据类型、引用和指针等初学者难以掌握的概念;二是让读者重点关注运算符的运算规则、优先级和结合性等语法细节;三是让读者初步体会到计算机语言与人类语言的不同之处,即计算机语言的语法规则非常严格,甚至到了机械的程度,稍有不慎即会出现语法错误。

第3章 算法与控制结构。本章讲解程序中的算法及三种算法基本结构,并通过选择结构和循环结构中的条件引出布尔类型。C++ 语言通过选择语句来实现算法的选择结构,通过循环语句来实现循环结构。最后以一个求反正切(arctan)函数的算法设计案例简单讲解算法的设计与评价方法。**本章要点:**一是让读者了解绝大部分复杂算法都可以由三种基本的算法结构来完成;二是让读者掌握布尔类型的作用及其相关的运算符;三是让读者根据案例认真体会如何根据算法来合理选用不同的控制语句。

第4章 数组与文字处理。本章学习如何在程序中存储和处理大量数据。数组可以存储大量具有相同类型的数据集合。4.1 节和 4.2 节具体讲解数组的定义和访问、指针与数组的关系以及内存动态分配等内容。计算机只能存储和处理数值数据,而文字处理程序所处理的对象是字符数据,为此 C++ 语言引入了字符类型。读者需深入了解字符编码和字符类型。文字处理必须使用数组,即字符型数组。本章最后用一节的篇幅简单介绍中文处理及 Unicode 编码。**本章要点:**一是让读者重点掌握数组定义及访问的语法规则;二是让读者认识到计算机内部对数组的管理和访问是通过指针(即内存地址)来实现的;三是让读者通过具体案例初步了解数组的常用处理算法。

第5章 结构化程序设计之一。本章学习如何将一个复杂的程序设计任务分解成多个简单的模块,分而治之,这就是结构化程序设计方法。C++ 语言支持结构化程序设计方法,以函数的语法形式来描述和组装模块,即函数的定义和调用。一个 C++ 程序通常由一个主

函数 main 和若干个子函数组成。函数是结构化程序设计方法的基础,它为代码重用提供了有效的手段。函数之间需要共享数据才能完成规定的数据处理任务。C++ 语言提供了集中管理和分散管理两种不同的数据管理策略。集中管理就是函数间通过共用内存来共享数据。分散管理就是函数各自分配内存管理数据,需要时传递数据,传递方式有三种(值传递、引用传递和指针传递)。**本章要点:**一是要让读者明确,函数是在被调用时才会执行,执行时会按需传递数据;二是让读者深入计算机内部,了解程序执行时其代码和变量在内存中的存储原理,这样可以更容易理解变量存储类型和作用域等抽象的概念;三是读者要准确把握函数间传递数据的三种方式;四是读者要从两个不同的角度,即定义函数的程序员角度和调用函数的程序员角度去看待与函数相关的各种语法知识,这样才能更容易地理解。

第 6 章 结构化程序设计之二。本章学习如何以多文件结构的形式来组织和管理源代码,并介绍几种常用的编译预处理指令。然后再介绍几种特殊形式的函数,其中包括带默认形参值的函数、重载函数、内联函数、带形参和返回值的主函数以及递归函数等。最后介绍 C 语言相关的系统函数和自定义数据类型。**本章要点:**一是学习掌握与多文件结构相关的语法知识,其中包括外部函数、全局变量的声明和头文件等;二是重点掌握带默认形参值的函数、重载函数和内联函数这三种常用的特殊函数形式;三是牢固树立重用代码的思想,学会通过调用别人编写的函数来提高开发效率。

第 7 章 面向对象程序设计之一。面向对象程序设计方法将程序中的数据元素和算法元素根据其内在关联关系进行分类管理,这就形成了“类”的概念。分类可以更好地管理。类相当于是一种自定义数据类型,用类所定义的变量称为“对象”。本章通过具体案例演示结构化程序设计是如何演变到面向对象程序设计的。本章内容包括类的定义、对象的定义与访问、对象的构造与析构、类中的常成员与静态成员以及类的友元等。**本章要点:**一是读者必须从代码分类管理、数据类型、归纳抽象和代码重用等多个维度才能准确理解类与对象的概念;二是读者需认真学习类与对象编程的具体语法规则;三是深入领会面向对象程序设计通过设置访问权限来实现类封装的基本原理;四是深入了解对象的构造与析构过程,程序员通过编写构造与析构函数来参与对象的构造与析构过程;五是读者要从两个不同的角度,即定义类的程序员角度和使用类定义对象的程序员角度,去看待类与对象相关的各种语法知识,这样才能更容易地理解。

第 8 章 面向对象程序设计之二。重用类代码有三种方式,分别是使用类定义对象、类的组合和类的继承。本章讲解类的组合与继承。程序员可以基于已有的零件类来定义新的整体类,这就是类的组合。程序员可以继承已有的基类来定义新的派生类,这就是类的继承与派生。利用派生类和基类之间的特殊关系可以进一步提高程序代码的可重用性,这就是面向对象程序设计中的对象多态技术。本章具体讲解与多态相关的运算符重载、虚函数和抽象类等概念。最后本章将简单讨论一下类的多继承。**本章要点:**一是让读者学会使用组合和继承的方法来定义新类,这样可以提高类代码的开发效率;二是读者应理解,类在组合或继承时可以进行二次封装;三是从提高对象处理算法代码可重用性的角度可以更容易地理解对象多态性;四是多继承会导致语法陷阱,新的面向对象程序设计语言(例如 Java 和 C#)已不再支持类的多继承,而只支持接口的多继承,读者只需要了解多继承的基本原理即可。

第 9 章 流类库与文件处理。C 语言通过输入输出函数(例如 scanf 和 printf)实现了数据的输入和输出。C++ 语言则是通过输入输出流类为程序员提供了输入或输出的功能。

这些输入输出流类都是从类 `ios` 派生出来的, 组成了一个以 `ios` 为基类的类族, 这个类族被称为 C++ 语言的流类库。本章将介绍流类库中三组不同功能的输入输出流类, 它们分别是: 通用输入输出流类, 提供通用的输入输出(简称标准 I/O)功能; 文件输入输出流类, 提供文件输入输出(简称文件 I/O)功能; 字符串输入输出流类, 提供字符串输入输出(简称字符串 I/O)功能。**本章要点:** 一是读者应理解之前所用的 `cin`、`cout` 指令实际上分别是通用输入输出流类的对象; 二是通过本章学习, 读者可以从侧面了解全球顶尖的 C++ 程序员是如何来设计和编写类的, 这样可以帮助读者进一步深入体会前面所学习的各种面向对象程序设计知识; 三是重点学习如何进行文件读写操作, 大部分程序都需要使用文件来保存数据。

第 10 章 C++ 标准库。C++ 语言全盘继承了 C 语言的标准 C 库, 另外又增加了一些新的库。新库中包含一些新增的系统函数, 但更多的是为面向对象程序设计方法所提供的系统类库, 这些新库被统称为 C++ 标准库。为了更好地凝练源代码, C++ 语言引入了模板技术, 其中包括函数模板和类模板。模板技术是一种代码重用技术, C++ 标准库在编写时就采用了模板技术, 因此标准库能以较少的代码量提供很强大的功能。本章重点介绍模板技术、C++ 语言的异常处理机制以及 C++ 标准库所提供的数据集合并存储及处理功能。**本章要点:** 一是让读者了解如何使用模板技术来提高函数和类代码的可重用性; 二是重点学习 C++ 语言的异常处理机制; 三是初步掌握如何使用 C++ 标准库中的向量类、列表类、集合类和映射类来存储和处理数据集合并。

4. 使用建议

凡希望开设“C++ 语言程序设计”在线教育课程的教师, 可将本书作为授课教材使用。联系作者可免费获得全套课件和参考教学视频。参加在线课程学习的学生可将本书作为线下阅读教材使用。

如果将本书作为课堂教学用书, 则建议讲课学时和实验学时各为 32 学时, 合计 64 学时。每学时 50 分钟。作者本人按如下方式安排讲课学时: 第 1、3、4、9、10 章各 2 学时, 第 2、5、6、7 章各 4 学时, 第 8 章 6 学时。

联系作者: kandaohong@cau.edu.cn

5. 致谢

作者编写本书的想法源于中国农业大学“雨虹学网: 面向主动学习的教学云平台建设”项目。在参与相关系统开发和教学实践的过程中, 作者积累了一些慕课(MOOC)在线课程教学的经验。

本书编写过程中, 得到了中国农业大学信息与电气工程学院高万林院长热情鼓励 and 大力支持。本书部分素材来自于雨虹学网的教学实践活动, 这得益于张晓东教授、孙瑞志教授等领导的关心和指导。另外本书编写过程中还得到了郑立华、吕春利、冀荣华、刘云玲、陈瑛、周绪宏、胡慧、段晶洁、李鑫、李静等同事和同学的热心帮助。在此一并致以衷心的感谢!

最后, 感谢家人对我的理解和支持。

作 者

2015 年 12 月于北京

目 录

第 1 章 程序设计导论	1
1.1 计算机硬件结构	1
1.2 计算机程序	4
1.3 计算机程序开发	8
1.3.1 程序设计	8
1.3.2 程序实现	9
1.3.3 程序测试	11
1.3.4 程序发布	11
1.4 信息分类与数据类型	12
1.4.1 二进制数制	12
1.4.2 数据类型	15
1.4.3 信息分类及数字化	17
1.5 C++语言简介	20
1.6 本章习题	21
第 2 章 数值计算	22
2.1 程序中的变量	23
2.1.1 变量的定义	23
2.1.2 变量的访问	25
2.2 程序中的常量	26
2.3 算术运算	29
2.3.1 C++语言中的加减乘除	29
2.3.2 其他算术运算符	31
2.4 位运算	33
2.5 赋值运算	36
2.6 数据的输入与输出	38
2.7 引用与指针	41
2.7.1 引用	41
2.7.2 指针	42
2.8 本章习题	48
第 3 章 算法与控制结构	50
3.1 算法	51

3.2	布尔类型	52
3.2.1	关系运算符	53
3.2.2	逻辑运算符	53
3.3	选择语句	55
3.3.1	if-else 语句	55
3.3.2	switch-case 语句	59
3.4	循环语句	63
3.4.1	while 语句	64
3.4.2	do-while 语句	65
3.4.3	for 语句	66
3.4.4	break 语句和 continue 语句	67
3.5	算法设计与评价	71
3.5.1	计算复杂度	71
3.5.2	内存占用量	73
3.5.3	算法设计举例	73
3.6	本章习题	75
第 4 章	数组与文字处理	76
4.1	数组	77
4.1.1	数组变量的定义与访问	77
4.1.2	常用的数组处理算法	80
4.2	指针与数组	84
4.2.1	指针运算	84
4.2.2	动态内存分配	88
4.2.3	指针数组	90
4.3	字符类型	91
4.3.1	字符型常量	92
4.3.2	字符型运算	93
4.4	字符数组与文字处理	94
4.4.1	字符串常量	94
4.4.2	字符数组	95
4.4.3	常用文字处理算法	96
4.5	中文处理	99
4.5.1	字符编码标准	100
4.5.2	基于 ANSI 编码的中文处理程序	101
4.5.3	基于 Unicode 编码的中文处理程序	103
4.6	本章习题	105

第 5 章 结构化程序设计之一	107
5.1 结构化程序设计方法	107
5.1.1 设计举例	107
5.1.2 基于模块的团队分工协作开发	109
5.1.3 模块的四大要素	111
5.2 函数的定义和调用	112
5.2.1 函数的定义	113
5.2.2 函数的调用	114
5.2.3 函数的执行	116
5.2.4 函数的声明	119
5.3 数据的管理策略	121
5.3.1 数据分散管理,按需传递	121
5.3.2 数据集中管理,全局共享	122
5.3.3 变量的作用域	124
5.4 程序代码和变量的存储原理	129
5.4.1 程序副本与变量	130
5.4.2 动态分配的内存	134
5.4.3 函数指针	135
5.5 函数间参数传递的三种方式	137
5.5.1 值传递	137
5.5.2 引用传递	138
5.5.3 指针传递	139
5.6 在函数间传递数组	141
5.6.1 在函数间传递一维数组	141
5.6.2 在函数间传递一维数组的首地址	142
5.6.3 在函数间传递二维数组	142
5.7 本章习题	143
第 6 章 结构化程序设计之二	145
6.1 C++源程序的多文件结构	145
6.1.1 多文件结构的源代码组织	145
6.1.2 静态函数与静态变量	148
6.1.3 头文件	151
6.2 编译预处理指令	153
6.2.1 文件包含指令	153
6.2.2 宏定义指令	154
6.2.3 条件编译指令	156
6.3 几种特殊形式的函数	159

6.3.1	带默认形参值的函数	159
6.3.2	重载函数	161
6.3.3	内联函数	162
6.3.4	主函数 main 的形参和返回值	163
6.3.5	递归函数	165
6.4	系统函数	170
6.4.1	C 语言的系统函数	171
6.4.2	命名空间	175
6.4.3	C++ 语言的系统函数	177
6.5	自定义数据类型	179
6.5.1	类型定义 typedef	179
6.5.2	枚举类型	181
6.5.3	联合体类型	182
6.5.4	结构体类型	185
6.6	结构化程序设计回顾	188
6.7	本章习题	189
第 7 章	面向对象程序设计之一	192
7.1	面向对象程序设计方法	192
7.1.1	结构化程序设计	192
7.1.2	面向对象程序设计中的分类	194
7.1.3	面向对象程序设计中的封装	197
7.1.4	面向对象程序的设计过程	200
7.2	类的定义	203
7.3	对象的定义与访问	205
7.3.1	对象的定义与访问	205
7.3.2	对象指针	206
7.3.3	类与对象的编译原理	207
7.4	对象的构造与析构	210
7.4.1	构造函数	211
7.4.2	析构函数	214
7.4.3	拷贝构造函数中的深拷贝与浅拷贝	215
7.5	对象的应用	219
7.5.1	对象数组	219
7.5.2	对象的动态分配	220
7.5.3	对象作为函数的形参	221
7.6	类中的常成员与静态成员	225
7.6.1	常成员	226
7.6.2	静态成员	228

7.7 类的友元	232
7.7.1 友元函数	233
7.7.2 友元类	234
7.8 本章习题	236
第8章 面向对象程序设计之二	239
8.1 代码重用	239
8.1.1 结构化程序设计中的代码重用	239
8.1.2 面向对象程序设计中的代码重用	240
8.2 类的组合	242
8.2.1 组合类的定义	243
8.2.2 组合类对象的定义与访问	244
8.2.3 组合类对象的构造与析构	246
8.2.4 类的聚合	248
8.2.5 前向引用声明	250
8.3 类的继承与派生	252
8.3.1 派生类的定义	253
8.3.2 派生类对象的定义与访问	256
8.3.3 保护权限与保护继承	257
8.3.4 派生类对象的构造与析构	260
8.3.5 继承与派生的应用	263
8.4 多态性	267
8.4.1 运算符的多态与重载	268
8.4.2 对象的替换与多态	272
8.4.3 虚函数	278
8.4.4 抽象类	281
8.5 关于多继承的讨论	285
8.5.1 多个基类之间的成员重名	286
8.5.2 重复继承	287
8.5.3 虚基类	288
8.6 本章习题	289
第9章 流类库与文件 I/O	292
9.1 流类库	293
9.2 标准 I/O	296
9.2.1 通用输入流类 istream 及其对象 cin	296
9.2.2 通用输出流类 ostream 及其对象 cout	301
9.3 文件 I/O	307
9.3.1 文件及其操作	307

9.3.2	文件输出流类 ofstream 及文件输出	310
9.3.3	文件输入流类 ifstream 及文件输入	314
9.3.4	文件输入/输出流类 fstream	316
9.4	string 类及字符串 I/O	320
9.4.1	字符串类 string	321
9.4.2	字符串 I/O	322
9.5	基于 Unicode 编码的流类库	323
9.6	本章习题	325
第 10 章	C++ 标准库	326
10.1	函数模板	326
10.1.1	函数模板的定义与使用	327
10.1.2	函数模板的编译原理	328
10.1.3	函数模板的声明	330
10.2	类模板	331
10.2.1	类模板的定义与使用	331
10.2.2	类模板的编译原理	333
10.2.3	类模板的继承与派生	334
10.3	C++ 标准库	336
10.4	C++ 语言的异常处理机制	338
10.4.1	程序中的三类错误	338
10.4.2	程序异常处理机制	341
10.4.3	try-catch 异常处理机制	343
10.4.4	C++ 标准库中的异常类 exception	347
10.5	数据集合及其处理算法	349
10.5.1	数据集合的存储和处理	350
10.5.2	C++ 标准库中数据集合的存储和处理	353
10.5.3	向量类 vector	356
10.5.4	列表类 list	360
10.5.5	集合类 set	360
10.5.6	映射类 map	361
10.6	结语	363
10.7	本章习题	363
附录	Microsoft Visual C++ 6.0 集成开发环境	364
参考文献		370

第1章

程序设计导论

本章内容

- 1.1 计算机硬件结构
- 1.2 计算机程序
- 1.3 计算机程序开发
- 1.4 信息分类与数据类型
- 1.5 C++语言简介

计算机是一种能够按照指令完成数值计算的机器。指令由人(如程序员)下达,由计算机中的电子电路(称为硬件)识别和执行。计算机硬件能够识别和执行的指令集合称为**机器语言**。机器语言应尽量简单,以提高执行速度,同时降低硬件生产成本。

程序员可以将多条指令编排成一个指令序列(称为**程序**),一次性提交给计算机,由计算机自动按顺序连续执行(见图 1-1)。程序描述了某种数据处理的过程和步骤。程序可以重复执行,每执行一次程序,计算机就完成一次程序所规定的数据处理过程。

一台可以工作的计算机由硬件和软件两部分组成,因此被称作是一个**计算机系统**。硬件包括中央处理器、存储器和输入输出设备等。软件包括操作系统、数据库管理系统、软件开发工具以及各种不同功能的应用程序。用户可以使用计算机系统协助自己完成某种特定工作,其操作过程通常是:首先启动计算机,执行某个程序,然后按照程序提示选择功能或输入原始数据,最后查看程序的处理结果(见图 1-2)。程序由程序员编写。用户应购买或通过合法渠道获得程序,并预先安装到自己的计算机系统中。

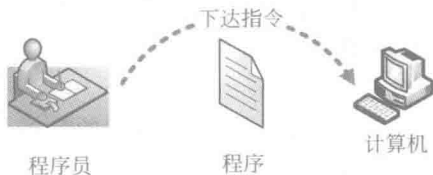


图 1-1 程序员、计算机与程序

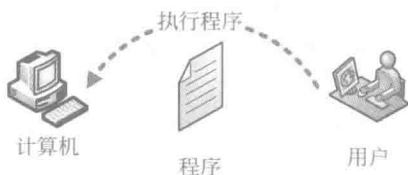


图 1-2 用户、计算机与程序

1.1 计算机硬件结构

冯·诺依曼体系结构(见图 1-3)是所有现代计算机硬件结构的基础。其核心思想是**存储程序计算机**,即包含一组指令序列的程序预先存储在存储器中,执行时由中央处理器从中

读取指令,并按顺序自动连续执行。存储程序计算机思想的最大贡献是将计算机变成了一种在程序控制下自动工作的机器,是从手动到自动的跨越。为了便于硬件实现,冯·诺依曼体系结构明确提出采用二进制数制对数据进行存储和运算。同时,用二进制数值编码来表示不同的机器语言指令。编码后,一段机器语言的程序实际上就是一个二进制数值编码序列,可以直接被计算机硬件识别、执行。

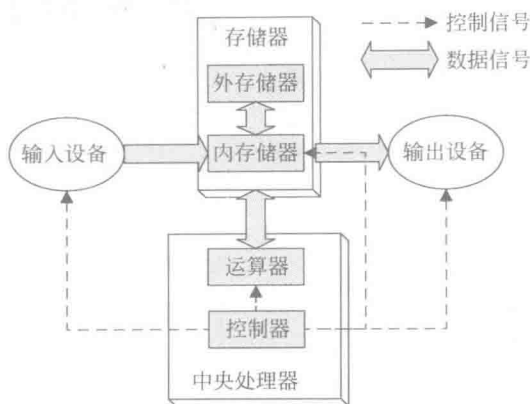


图 1-3 冯·诺依曼体系结构

冯·诺依曼体系结构包含五个基本组成部分：运算器、控制器、存储器、输入设备和输出设备。运算器和控制器合起来被称为中央处理器(Central Processing Unit,CPU)。存储器可分为内存储器(简称内存)和外存储器(简称外存,例如硬盘)两种。输入设备包括键盘、鼠标、扫描仪和麦克风等,而输出设备则包括显示器、打印机和音箱等。输入/输出设备统称为计算机的外围设备,是人与计算机进行交互的接口。

1. 存储器

程序预先存储在存储器中。执行程序时,CPU从存储器中逐条读出指令,识别并执行该指令,然后再到存储器读取下一条指令。存储器为CPU连续执行指令,实现自动工作奠定了基础。

程序是用来处理数据的。程序中的数据包括原始数据、中间结果和最终结果等。数据要存放在存储器中才能被CPU读取和处理,CPU处理后的结果也只能保存回存储器中。

存储器可分为内存和外存两种。内存的读写速度快,但造价高,而且内存中的数据在断电时(例如关机)会丢失。外存(例如硬盘)读写速度慢,但造价低,而且断电时数据不丢失,可以长期保存。程序平时是以文件形式存放在外存上,被执行时才被读入内存,并在内存中执行。程序所处理的数据也需要放在内存中才能被处理,例如硬盘上的一份Word文档,打开文档时被读入内存处理,处理完后保存文档,重新将文档写回硬盘以便长期保存。

对存储器的操作有写入(Write)和读出(Read)两种,即将数据写入存储器,或从存储器读出数据。对存储器的读写操作统称为对存储器的访问。

一个二进制位(bit)可以存储一个二进制数,即0或1。8个二进制位组成一个字节(Byte)。其他常用的存储单位还有KB(1024B)、MB(1024KB)和GB(1024MB)等。