

高职高专计算机专业系列教材

C++ 程序设计

李龙澍 主编

清华大学出版社
北京

内 容 简 介

本书系统讲述了 C++ 语言的基本概念和编程方法。首先介绍 C++ 的数据类型和基本表达式、基本语句和函数,接着阐述面向对象的基本概念和类、对象的设计方法,然后讲述 C++ 程序设计的技巧和输入输出结构,最后通过实际例子阐明 C++ 程序设计方法和技巧。

本书针对高职高专学生的特点,力求将复杂的概念用简洁的语言描述出来,旨在帮助读者提高分析问题和解决问题的能力,让读者学完本书后能够用 C++ 语言编写实际应用程序。本书内容丰富、结构合理、语言流畅,融趣味性于科学性于一体,同时配有大量习题和实训题目,与本书配套的《C++ 程序设计实训》可以加深对本书的理解。

本书适合作为高职高专计算机及相关专业“程序设计语言”课程的教材,也可供各类计算机应用人员学习使用。

版权所有,翻印必究。

本书封面贴有清华大学出版社激光防伪标签,无标签者不得销售。

图书在版编目(CIP)数据

C++ 程序设计/李龙澍主编.—北京:清华大学出版社,2003

(高职高专计算机专业系列教材)

ISBN 7-302-06290-0

.C... .李... .C 语言 - 程序设计 - 高等学校:技术学校 - 教材
.TP312

中国版本图书馆 CIP 数据核字(2003)第 007565 号

出 版 者:清华大学出版社(北京清华大学学研大厦,邮编 100084)

<http://www.tup.com.cn>

责任编辑:张 民

印 刷 者:北京市人民文学印刷厂

发 行 者:新华书店总店北京发行所

开 本:787×1092 1/16 印张:17.25 字数:395 千字

版 次:2003 年 3 月第 1 版 2003 年 3 月第 1 次印刷

书 号:ISBN 7-302-06290-0 TP·4755

印 数:0001~6000

定 价:23.00 元

序

1999年10月,教育部高教司主持召开了全国高职高专教材工作会议,会议要求尽快组织规划和编写一批高质量的、具有高职高专特色的基础和专业教材。根据会议精神,在清华大学出版社的支持下,于2000年1月在上海召开了由来自全国各地的部分高职、高专、成人教育及本科院校的代表参加的“高职高专计算机专业培养目标和课程设置体系研讨会”。与会的专家和教师一致认为,在当前教材建设严重滞后同高职教育迅速发展的矛盾十分突出的情况下,编写一套适应高等职业教育培养技术应用性人才要求的、真正具有高职特色的、体系完整的计算机专业系列教材十分必要而且迫切。会议成立了高职高专计算机专业系列教材编审委员会,明确了高职计算机专业的培养目标,即掌握计算机专业有关的基本理论、基本知识和基本技能,尤其要求具有对应用系统的操作使用、维护维修、管理和初步开发的能力。

根据上述目标,编委会拟定了本套教材的编写原则。在教材内容安排上,以培养计算机应用能力为主线,构造该专业的课程设置体系和教学内容体系;从计算机应用需求出发进行理论教学,强调理论教学与实验实训紧密结合,尤其突出实践体系与技术应用能力的实训环节的教学;教材编写力求内容新颖、结构合理、概念清楚、实用性强、通俗易懂、前后相关课程有较好的衔接。与本科教材相比,本套教材在培养学生的应用技能上更有特色。

根据目前各高职高专院校计算机专业的课程设置情况,编委会确定了首批出版的十几本教材。这些教材的作者多是在高职高专院校或本科院校的职业技术学院任教的、具有多年教学经验的教师,每本书均由计算机专业的资深教授或专家主审把关。我们还将在此基础上,陆续征集出版第二、三批教材,力争在3到5年内完成一套完整的高职高专计算机专业教材。

应当说明的是,凡是高等职业教育、高等专科教育和成人高等院校的计算机及其相关专业均可使用本套教材。各学校可以根据实际需要,在教学中适当增删一些内容、实训项目和练习题,从而更有针对性地帮助学生掌握计算机专业知识,并形成相关的应用能力。

由于各地区各学校在教学水平、培养目标理解等方面有所不同,加上这套教材编写时间仓促,难免会出现这样或那样的错误,敬请各学校在使用过

程中及时将修改意见或好的建议返回给教材编审委员会,以便我们及时修订、改版,使该系列教材日趋完善。

我们恳切地希望高职高专院校任课的专业教师和专家对后续教材的编写提出建设性的意见,并真诚地希望各位教师参与我们的工作。

高职高专计算机专业
系列教材编审委员会
2000年5月

前 言

随着信息科学的发展,计算机的应用范围越来越广,每一个计算机工作者都有必要学会并熟练掌握一门计算机语言。C++ 是当前最流行的一种面向对象程序设计语言,它是在 C 语言的基础上扩充发展起来的,C++ 程序设计把数据和关于数据的操作封装在一起,这种解决问题的方法更符合人们的思维习惯,使用 C++ 编制程序更方便,软件维护更容易。

本书是作者在多年从事 C++ 语言程序设计教学的基础上编写的,根据多年的教学经验,针对学生学习中遇到的问题,反复修正教学内容,总结启发式教学思路,力争使学生学得会,用得上。

本书思路清晰,重点放在使读者提高分析问题和解决问题的能力上;通俗易懂,将复杂的概念用读者容易理解的简洁语言描述出来;由浅入深,从最简单的概念开始,让读者逐步掌握 C++ 语言的完整体系;重在实用,让读者学完本书后会用 C++ 语言编写实际应用程序。

全书共分 12 章,第 1 章 C++ 入门,介绍 C++ 的简单概念,讲述应用 Visual C++ 6.0 编制小程序。第 2 章 数据类型和表达式,介绍 C++ 的基本数据类型和基本表达式。第 3 章 语句和函数,讲述 C++ 的基本语句和最小程序实体函数。第 4 章 面向对象的基本概念与类,类是 C++ 的编程基础。第 5 章 对象,讲述对象的设计和使用,对象是 C++ 最基本的程序实体。第 6 章 指针和引用,讲述指针的定义和使用,讨论引用的定义和使用。第 7 章 继承,讲述类之间的继承机制。第 8 章 静态成员和友元,介绍静态成员、友元、运算符重载等 C++ 的一些高级编程技巧。第 9 章 多态和虚函数,讨论类之间派生中的动态继承问题。第 10 章 输入/输出流,介绍 C++ 的基本输入输出机制。第 11 章 模板和异常处理,讨论 C++ 的模板使用技巧和异常处理方法。第 12 章 综合应用实例,给出实际应用例子。本书的全部例题在 Visual C++ 6.0 环境下运行通过。读者还可以学习配套的《C++ 程序设计实训》,加深对本书的理解。

本书是在李龙澍教授的主持下完成的。初稿的第 1 章、第 4 章和第 5 章由李龙澍执笔,第 2 章和第 3 章由唐彬执笔,第 6 章和第 7 章由卢冰原执笔,第 8 章、第 9 章 第 11 章由杨增光执笔,第 10 章和第 12 章由凌成执

笔。全书由李龙澍统一修改后定稿。张霞、杨涛、叶红、庞开放为本书的编写也做了大量的工作。中国科学技术大学刘振安教授、安徽大学程慧霞教授为本书提出了许多建设性的宝贵意见,作者在此表示衷心的感谢。

由于作者水平有限,难免出现一些疏 和错误,殷切希望读者提出批评建议和修改意见。

作 者
2003 年 1 月于安徽大学

目 录

第 1 章 C++ 入门	1
1.1 什么是 C++	1
1.1.1 什么是 C++ 程序设计语言	1
1.1.2 C++ 程序开发过程	2
1.2 一个 C++ 程序	3
1.3 C++ 程序的结构	5
1.3.1 主程序	5
1.3.2 函数	6
1.3.3 输入/输出	7
1.3.4 头文件	7
1.3.5 注释	8
1.4 例题分析和小结	8
1.4.1 例题	8
1.4.2 例题分析	10
1.4.3 小结	12
实训 1 编制一个简单的 C++ 程序	12
习题	12
第 2 章 数据类型和表达式	14
2.1 词法符号	14
2.1.1 标识符	14
2.1.2 关键字	15
2.1.3 常量	15
2.2 基本数据类型	17
2.2.1 简单数据类型	17
2.2.2 变量	19
2.3 结构数据类型	20
2.3.1 数组	20

2 3 2	结构体类型	25
2 3 3	共用体类型	26
2 4	表达式.....	27
2 4 1	算术表达式	28
2 4 2	关系表达式	29
2 4 3	逻辑表达式	30
2 4 4	运算顺序	30
2 5	例题分析和小结.....	32
2 5 1	例题	33
2 5 2	例题分析	36
2 5 3	小结	36
实训 2	标识符和表达式	37
习题	37
第 3 章	语句和函数	41
3 1	赋值语句.....	41
3 2	选择语句.....	43
3 2 1	条件语句	43
3 2 2	开关语句	44
3 3	循环语句.....	46
3 3 1	while 循环语句	46
3 3 2	for 循环语句.....	47
3 3 3	break 语句和 continue 语句	48
3 3 4	多重循环	48
3 4	函数.....	49
3 4 1	函数的定义	49
3 4 2	函数的调用	50
3 4 3	函数的传值参数	53
3 4 4	函数的引用参数	55
3 4 5	函数的默认值	56
3 5	函数的重载.....	56
3 5 1	函数参数类型重载	57
3 5 2	函数参数个数重载	58
3 6	系统函数的调用.....	59
3 7	例题分析和小结.....	60
3 7 1	例题	60

3.7.2	例题分析	63
3.7.3	小结	63
实训 3	职工信息处理和趣味取球	63
习题	64
第 4 章	面向对象的基本概念与类	67
4.1	面向对象程序设计的基本概念.....	67
4.1.1	对象	67
4.1.2	抽象	69
4.1.3	封装	70
4.1.4	继承	72
4.1.5	多态	73
4.2	类.....	74
4.2.1	类的定义	74
4.2.2	类的数据成员	75
4.2.3	类的成员函数	77
4.2.4	类成员函数的存取权限	79
4.3	成员函数重载.....	81
4.4	例题分析和小结.....	83
4.4.1	例题	83
4.4.2	例题分析	85
4.4.3	小结	85
实训 4	建造集合类	86
习题	86
第 5 章	对象	89
5.1	对象的建立和撤消.....	89
5.1.1	对象的定义	89
5.1.2	构造函数	92
5.1.3	析构函数	97
5.2	对象的赋值	100
5.2.1	复制构造函数.....	100
5.2.2	重载赋值运算符.....	104
5.2.3	修改对象的数据成员.....	106
5.3	例题分析和小结	107
5.3.1	例题.....	107

5 3 2	例题分析.....	109
5 3 3	小结.....	109
实训 5	数组数据处理对象	110
习题	110
第 6 章	指针和引用	115
6.1	指针	115
6.1.1	指针变量的定义.....	115
6.1.2	指针的赋值.....	117
6.1.3	对象指针.....	120
6.1.4	this 指针	122
6.2	引用	123
6.2.1	引用的定义和使用.....	123
6.2.2	引用返回值.....	125
6.3	例题分析和小结	126
6.3.1	例题.....	126
6.3.2	例题分析.....	127
6.3.3	小结.....	127
实训 6	编制一个排序数组类	128
习题	128
第 7 章	继承	132
7.1	基类和派生类	132
7.1.1	派生类的定义.....	132
7.1.2	继承方式.....	134
7.2	单继承	135
7.2.1	继承成员的访问权限.....	135
7.2.2	构造函数和析构函数.....	139
7.2.3	单继承的应用.....	141
7.3	多继承	144
7.3.1	多继承的概念.....	144
7.3.2	多继承的构造函数.....	145
7.3.3	多继承的应用.....	145
7.4	虚基类	146
7.4.1	虚基类的定义.....	146

7 4 2	虚基类的构造函数.....	148
7 4 3	虚基类的应用.....	149
7 5	例题分析和小结	151
7 5 1	例题.....	151
7 5 2	例题分析.....	153
7 5 3	小结.....	153
实训 7	人员类的继承	154
习题	154
第 8 章	静态成员和友元	158
8.1	静态成员	158
8.1.1	静态成员的定义.....	158
8.1.2	静态成员的使用.....	159
8.2	友元	161
8.2.1	友元的定义.....	161
8.2.2	友元的使用.....	163
8.3	运算符重载	165
8.3.1	运算符重载规则.....	166
8.3.2	重载为成员函数.....	166
8.3.3	重载为友元函数.....	168
8.4	例题分析和小结	170
8.4.1	例题.....	170
8.4.2	例题分析.....	174
8.4.3	小结.....	174
实训 8	个人所得税计算和运算符重载	175
习题	176
第 9 章	多态和虚函数	178
9.1	虚函数	178
9.1.1	虚函数的定义.....	178
9.1.2	纯虚函数.....	182
9.2	抽象类	183
9.3	多态	186
9.3.1	多态的概念.....	186
9.3.2	多态的应用.....	186
9.4	例题分析和小结	188

9 4 1	例题.....	188
9 4 2	例题分析.....	190
9 4 3	小结.....	190
实训 9	应用多态设计学生类	191
习题	191
第 10 章	输入/输出流	193
10 1	输入/输出流的概念	193
10 2	输出流.....	195
10 2 1	基本输出操作.....	195
10 2 2	输出格式控制.....	197
10 3	输入流.....	207
10 3 1	基本输入操作.....	207
10 3 2	输入控制格式.....	209
10 4	文件.....	210
10 4 1	文件的打开和关闭.....	210
10 4 2	文件的读写.....	212
10 4 3	文件的随机读写.....	216
10 5	例题分析与小结.....	219
10 5 1	例题.....	219
10 5 2	例题分析.....	221
10 5 3	小结.....	221
实训 10	输入流和输出流	221
习题	222
第 11 章	模板和异常处理	226
11 1	模板.....	226
11 1 1	模板的定义.....	226
11 1 2	模板的使用.....	227
11 2	异常处理.....	229
11 2 1	异常处理的语法结构.....	229
11 2 2	异常处理的应用.....	230
11 3	例题分析和小结.....	231
11 3 1	例题.....	231
11 3 2	例题分析.....	233
11 3 3	小结.....	234

实训 11 建造数组模板和异常处理	234
习题.....	234
第 12 章 综合应用实例	238
12.1 商场销售管理系统.....	238
12.1.1 商场销售管理系统类的设计.....	239
12.1.2 商场销售管理系统的实现.....	240
12.2 小结.....	258
实训 12 职工工资管理系统	258
参考文献	260

第 1 章

C++ 入门

为了更好地适应现代信息社会的工作要求,每一个计算机工作者都有必要学会并熟练掌握一门计算机语言。C++ 语言是当今最流行的一种面向对象计算机程序设计语言。

1.1 什么是 C++



1.1.1 什么是 C++ 程序设计语言

人们要进行某种游戏必须遵循这种游戏的游戏规则,计算机程序设计语言是一种人与计算机交互的游戏规则,想编写程序让计算机完成某种任务的人,必须熟悉这种程序的游戏规则。计算机是按照人编写的程序进行工作的。计算机程序设计语言是计算机可以认识的语言,人们用这种语言描述要解决的问题,计算机可以阅读和执行。

计算机语言随着计算机科学的发展而发展,它的每一步发展都使计算机语言与人类的自然语言更加接近。

在 20 世纪 80 年代以前,人们编写程序是面向过程的,就是把现实生活中的问题转化成一个一个的过程,再把每个过程编写成程序。

到了 20 世纪 80 年代,出现了面向对象的思想,就是为一个事物(或实体)编写一段程序,不要再把它转换成过程,少了个中间环节,也就减少了出错的可能性。

C++ 是当今最流行的一种面向对象的程序设计语言,它是在 20 世纪 80 年代初由贝尔实验室开发的。当时 C 语言已经非常流行,随着问题复杂度的提高和面向对象方法的提出,C 语言越来越显得力不从心,C++ 是由 C 语言扩展而成的,它继承了 C 语言的优点,又极大地扩充了 C 语言的功能。C++ 已经在众多应用领域中被作为首选的程序设计语言,它尤其适用于开发中等和大型的计算机应用项目。从开发时间、费用到形成的软件的可重用性、可扩充性、可维护性以及可靠性等方面都显示出 C++ 的优越性。



1.1.2 C++ 程序开发过程

当编写 C++ 语言程序时,必须遵循 C++ 语言的游戏规则,这个游戏规则包括 C++ 语言的语法规则和研制程序的操作规范。

世界上有很多种 C++ 语言,比较流行的有 Visual C++ (简称 VC++) 和 Borland C++,它们有一致的语法规则,但有不同的操作规范。为了便于读者的学习,本书选用 VC++ 作为编写 C++ 语言程序的环境。

VC++ 是开发 C++ 程序的集成开发环境,包括编辑、编译、连接、运行几个环节。如要研制一个命名为 Hello 的程序,其操作流程如图 1-1 所示。

图 1-1 研制 Visual C++ 程序的过程

1. 编辑

编辑是将写好的 C++ 源程序输入到计算机中,生成磁盘文件的过程。程序的编辑在计算机提供的编辑器中进行。

VC++ 提供了一个包括编辑、编译、连接、执行于一体的集成开发环境。VC++ 有一个功能良好的编辑器。主要编辑功能有:


定义块 在编辑 C++ 源程序的正文工作区,将鼠标的光标移到要定义块的一端,按下鼠标左键,拖动鼠标到要定义块的另一端,松开鼠标左键,鼠标移过的部分变黑,一块就定义好了。

移动块 将鼠标的光标放到变黑的块上,按下鼠标左键,拖动鼠标到新的位置,松开鼠标左键,块就移动到了新位置。

复制块 将鼠标的光标放到变黑的块上,左手按下 Ctrl 键,右手按下鼠标左键,拖动鼠标到新的位置,松开鼠标左键,松开 Ctrl 键,块就复制到了新位置。

删除块 按 Delete 键,定义的块就被删除。

插入 将光标移动到要插入的位置,键入要插入的字符。实际上,录入源程序就是在正文工作区的尾部进行插入。

保存 选择文件菜单下的保存命令,或点击  按钮。

将编辑的 C++ 程序存盘时,会自动加上扩展名 cpp,这是 C++ 源程序的默认扩展名。

2. 编译

编辑好的源程序必须编译成机器代码,计算机才能执行。编译器可将编辑好的程序转换成二进制机器代码的形式,编译好的机器代码称为目标代码。

C++ 的编译分两大步进行:

(1) 预处理过程

编译器首先编译预处理命令,包括找到预处理文件的位置,打开预处理文件,后面的源程序要用到预处理文件定义的内容。

(2) 编译源程序

编译源程序就是将编辑完成的 C++ 源程序翻译成计算机硬件能够认识的目标代码。编译器的一个重要功能就是检查 C++ 源程序的语法错误。如在程序中将输出流 cout 误写为 couth,进行编译后,在 VC++ 用户界面下端的输出窗口会显示:“error C2065: couth : undeclared identifier”,表明 couth 是一个没有说明的标识符,用鼠标左键双击出错信息,光标就会指到出错的 couth 程序行。修改后,重新编译,这个错误就没有了。待源程序没有语法错误时,再编译就生成目标代码。目标代码的扩展名为 obj。

3. 连接

编译得到的目标代码还不能直接在计算机上运行,必须把目标代码连接成执行文件以后才能运行。如果预处理文件没有错误,连接时就把预处理文件指定的库函数复制到源程序中用到它的地方。执行文件的扩展名是 exe。

4. 运行

C++ 源程序经过编译和连接生成的可执行文件可以在操作系统环境下单独运行,也可以在 VC++ 集成环境下运行。

选择 VC++ 集成环境下的“执行当前程序”命令,程序就进入运行状态,一般在屏幕上可以看到程序的运行结果,或提示输入数据等信息。

1 2 一个 C++ 程序

要知道 C++ 程序的奥秘,最好先编写一个简单的 C++ 程序。

现在编制一个名为 Hello 的程序,该程序在屏幕上显示下面 3 句话:

Hello, World!

你真聪明,你已经会用 C++ 编写程序了!

Bye, 朋友!

编写程序就像写文章一样,一篇文章要分成若干段,程序也要分段。每段程序称作一个模块。Hello 程序可以分为两段,显示“Hello, World! 你真聪明,你已经会用 C++ 编写程序了!”为一段,显示“Bye, 朋友!”为另一段。

按下列步骤进行:

运行 Visual C++。

在 File 菜单下选择 New 命令,打开 New 对话框。

在 Projects 选项卡中指定下列选项:

Projects Win32 Console Application
Project Name Hello
Location 指定应用程序存放的位置
Create New Workspace 选中(默认)
Platforms Win32 选中(默认)

设置完成后,单击 OK 按钮,打开 AppWizard 对话框。

在 AppWizard 对话框中,单击 A Simple Application。

单击 Finish 按钮,显示 New Project Information 对话框,内容如下:

```
+ Simple Win32 Console application .  
Main Hello.cpp  
Precompiled Header Stdafx.h and Stdafx.cpp
```

单击 OK 按钮。

至此,一个新的应用程序创建完成,自动创建了主文件 Hello.cpp,项目还自动创建了预编译头文件 Stdafx.h 和系统设置程序 Stdafx.cpp。Hello.cpp 的内容为:

```
// Hello.cpp: Defines the entry point for the console application .  
#include <stdafx.h>  
int main(int argc, char * argv[])  
{  
    return 0;  
}
```

在 Hello.cpp 中,插入写好的程序。最后,完整的 Hello 程序如下:

```
// Hello.cpp: Defines the entry point for the console application .  
#include <stdafx.h>  
#include <iostream.h>  
// 函数原型  
void SayHello();  
void SayGoodbye();  
int main(int argc, char * argv[])  
{  
    SayHello(); // 调用 SayHello 函数  
                // 输出“ Hello, World !”和“ 你真聪明,你已经会用 C++ 编写程序了!”  
    SayGoodbye(); // 调用 SayHello 函数,输出“ Bye, 朋友!”  
    return 0;  
}  
// 函数 SayHello 定义  
void SayHello()  
// 输出“ Hello, World !”和“ 你真聪明,你已经会用 C++ 编写程序了!”  
{  
    cout << Hello, << World ! << endl;
```