

高等学校教材·计算机教学丛书

C++ 程序设计方法

张桂香 廉佐政 主 编

王海珍 刘娜娜 张光妲 李耀成 副主编

滕永富 主 审



北京航空航天大学出版社
BEIHANG UNIVERSITY PRESS

高等学校教材·计算机教学丛书

C++程序设计方法

张桂香 廉佐政 主 编

王海珍 刘娜娜 张光姐 李耀成 副主编

滕永富 主 审

北京航空航天大学出版社

BEIHANG UNIVERSITY PRESS

内 容 简 介

本书是根据编者多年 C++ 程序设计教学经验,精心设计的一本集理论学习、习题自测与实验练习和 C++ 开发实例于一体的程序设计方法教科书。理论内容简练清晰,阐述了 C++ 的重点、难点和易混淆点,习题配有答案,所有核心实验都给出分析与提示以及源程序。共分 13 章,包括 C++ 概述、C++ 语言基础、数据类型和表达式、控制结构、数组、函数、类与对象基础、类与对象应用、继承与派生、多态性、I/O 流类库、模板以及 C++ 开发实例和实验部分等。特别适合将 C++ 程序设计作为程序设计语言课的高等院校本科学士生作为教材,经过教师删节也可作为高职、高专的教材,也可以作为计算机培训的辅导教材以及 C++ 学习的自学教材。

图书在版编目(CIP)数据

C++ 程序设计方法 / 张桂香等主编. -- 北京: 北京航空航天大学出版社, 2012. 1

ISBN 978 - 7 - 5124 - 0668 - 1

I. ①C… II. ①张… III. ①C 语言—程序设计 IV.

①TP312

中国版本图书馆 CIP 数据核字(2011)第 249920 号

版权所有,侵权必究

C++ 程序设计方法

张桂香 廉佐政 主编

王海珍 刘娜娜 张光旦 李耀成 副主编

滕永富 主审

责任编辑 许传安

*

北京航空航天大学出版社出版发行

北京市海淀区学院路 37 号(邮编 100191) <http://www.buaapress.com.cn>

发行部电话:(010)82317024 传真:(010)82328026

读者信箱: bhpess@263.net 邮购电话:(010)82316936

北京时代华都印刷有限公司印装 各地书店经销

*

开本: 787×1092 1/16 印张: 20 字数: 512 千字

2012 年 1 月第 1 版 2012 年 1 月第 1 次印刷 印数: 4 000 册

ISBN 978 - 7 - 5124 - 0668 - 1 定价: 34.00 元

若本书有倒页、脱页、缺页等印装质量问题,请与本社发行部联系调换。联系电话:(010)82317024



总 前 言

随着科学技术、文化、教育、经济和社会的发展,计算机教学进入了我国历史上最火热的年代,欣欣向荣。就计算机专业而言,全国开办计算机本科专业的院校在 2004 年之初有 505 所,到 2006 年已经发展到 771 所。另外,在全国高校中的非计算机专业,包括理工农医以及文科(文史哲法教、经管、文艺)等专业,按各自专业的培养目标都融入了计算机课程的教学。过去出版界出版了一大批计算机教学方面的各类教材,满足了一定时期的需求,但是还不能完全适应计算机教学深化改革的要求。

面对《国家科学技术中长期发展纲要(2006 年—2020 年)》制订的信息技术发展目标,计算机教学也要随之进行改革,以便提高培养质量。教学要改革,教材建设必须跟上。面对各层次、各类型的学校和各类型的专业都要开设计算机课程,就应有多样化的教材,以适应各专业教学的需要。北京航空航天大学出版社是以出版高等教育教材为主的,愿对计算机教学的教材建设做出贡献。

为计算机类教材的出版,北京航空航天大学出版社成立了“高等学校教材·计算机教学丛书”编审委员会。出版计算机教材,得到了北京航空航天大学计算机学院的大力支持。该院有三位教育部高等学校计算机科学与技术教学指导委员会(下称教指委)的成员参加编审委员会的工作。其他成员是北京航空航天大学、北京交通大学等 6 所院校和中科院计算技术研究所对计算机教育有研究的教指委成员、专家、学者和出版社的领导。

我们组织编写、出版计算机课程教材,以大多数高校实际状况为基点,使其在现有基础上能提高一步,追求符合大多数高校本科教学适用为目标。按照教指委制订的计算机科学与技术本科专业规范和计算机基础课教学基本要求的精神,我们组织身居教学第一线,具有教学实践经验的教师进行编写。在出书品种和内容上,面对两个方面的教学:一是计算机专业本科教学,包括计算机导论、计算机专业技术基础课、计算机专业类等;二是非计算机专业的计算机基础课程的本科教学,包括理工农医类、文史哲法教类、经管类、艺术类等的计算机课程。

教材的编写注重以下几点。

1. 基础性。具有基础知识和基本理论,以使学生在专业发展上具有潜力,便于适应社会的需求。
2. 先进性。融入计算机科学与技术发展的新成果;瞄准计算机科学与技术发展的新方向,内容应具有前瞻性。这样,以使学生扩展视野,以便与科技、社会发展的脉络同步。
3. 实用性。一是适应教学的需求;二是理论与实践相结合,以使学生掌握实用技术。

编写、出版的教材能否适应教学改革的需求,只有师生在教与学的实践中做出评价,我们期望得到师生的批评和指正。



“高等学校教材·计算机教学丛书” 编审委员会成员

主 任 马殿富

副主任 麦中凡

陈炳和

委 员 (以音序排列)

陈炳和 邓文新 金茂忠

刘建宾 刘明亮 罗四维

卢湘鸿 马殿富 麦中凡

张德生 谢建勋 熊 璋

张 莉



前 言

教育部计算机教学指导委员会对理工类专业计算机程序设计基础课程的设置规定,主要是 C++ 程序设计,或 Visual Basic 程序设计。

C++ 是目前流行的计算机程序设计语言之一。它的功能强大,不仅支持面向过程的程序设计,还全面支持面向对象的程序设计,因此,学习 C++ 要比学习 C 语言和 Pascal 等语言困难得多。编写本书的目的主要是,为将 C++ 作为第一门程序设计语言的本专科学生提供一本适用教材,以便学生能够快速掌握 C++ 的基本理论、程序设计的思想和实践环节。

编者根据多年的计算机程序设计教学经验,按照学生学习的认知规律,精心设计了本教材,内容集理论学习、习题自测与实验操作以及 C++ 开发实例于一体。理论内容简练清晰,阐述了 C++ 的重点、难点和易混淆点,习题配有答案,所有核心实验题目均给出了分析与提示以及程序源代码。全书共 13 章,第 1 章 C++ 概述,介绍 C++ 的发展、上机环境和实例等,使读者对程序设计的全貌有所了解。第 2 章到第 11 章为 C++ 的基本理论,介绍了语言基础、程序、控制结构、数组、函数、类与对象基础、类与对象应用、继承与派生、多态性、I/O 流类库、模板等。第 12 章给出了一个 C++ 开发实例,是全书内容的一个综合,也可以作为学生课程设计的参考。第 13 章是实验部分,根据各章节内容给出实验的题目和要求。本书所有例题程序和实验题目均在 Visual C++ 6.0 上调试通过。

本书由张桂香、廉佐政、王海珍、刘娜娜、张光坦、李耀成编著。张桂香负责编写第 1、3、4 章和实验一、实验五到实验九。廉佐政负责编写第 6、7、12 章和实验十三到实验十六、实验二十四。王海珍负责编写第 8、9 章和实验十七到实验二十二。刘娜娜负责编写第 10、11 章和实验二十三。张光坦负责编写第 5 章和实验十到实验十二。李耀成负责编写第 2 章和实验二到实验四。张桂香、廉佐政统稿。滕永富教授审阅全书,并提出了宝贵意见。

本书在编写过程中得到了邓文新教授和出版社的大力支持和帮助,在此表示衷心的感谢。同时对编写过程中参考文献资料的作者一并表示谢意。

由于作者水平有限,书中难免有疏漏之处,敬请读者批评指正。

编著者
2011 年 7 月



目 录

第 1 章 C++ 语言概述	1	2.6 关于类型定义	24
1.1 C++ 发展简史	1	2.7 运算符	25
1.1.1 程序设计方法	1	2.7.1 算术运算符	26
1.1.2 C++ 发展简史	2	2.7.2 关系运算符	27
1.2 C++ 程序实例	3	2.7.3 逻辑运算符	28
1.2.1 从一个简单的程序看 C++ 程序的组成	3	2.7.4 位运算符	28
1.2.2 C++ 字符集和关键字	4	2.7.5 赋值运算符	29
1.2.3 书写规则和程序设计风格	5	2.7.6 其他运算符	29
1.3 C++ 上机环境	5	2.8 表达式	30
1.3.1 C++ 程序的开发过程	5	2.8.1 表达式的种类	30
1.3.2 C++ 集成开发环境有哪些	6	2.8.2 表达式的值和类型	31
1.3.3 Visual C++ 和 C++ 的关系	6	2.8.3 表达式中的类型转换	33
1.3.4 Visual C++ 6.0 介绍	6	2.9 习题二	35
1.3.5 在 Visual C++ 6.0 中开发 C++ 程序的过程	8	第 3 章 控制结构	38
1.4 习题一	11	3.1 编译预处理	38
第 2 章 数据类型与表达式	12	3.1.1 编译预处理的作用	38
2.1 基本数据类型	12	3.1.2 编译预处理语句	38
2.1.1 关于整型的要点	12	3.2 顺序结构	40
2.1.2 关于浮点型的要点	13	3.2.1 C++ 输入输出	40
2.1.3 关于字符型的要点	14	3.2.2 顺序结构程序	41
2.1.4 关于布尔型的要点	14	3.3 选择结构	42
2.1.5 关于空类型的要点	14	3.3.1 if 语句	42
2.2 常量与变量	14	3.3.2 switch 语句	45
2.2.1 关于常量的要点	14	3.4 循环结构	46
2.2.2 关于变量的要点	17	3.4.1 循环结构的组成	46
2.3 指针类型	19	3.4.2 while 语句	47
2.3.1 地址、指针与指针变量之间的联系与区别	19	3.4.3 do while 语句	48
2.3.2 什么是直接访问和间接访问	20	3.4.4 for 语句	49
2.3.3 什么是引用	20	3.4.5 三种循环结构的比较	50
2.4 结构体与共用体	21	3.4.6 循环嵌套	51
2.5 枚举类型	23	3.5 其他控制语句	52
		3.5.1 break 语句	53
		3.5.2 continue 语句	53
		3.5.3 goto 语句	54



3.6 习题三	55	5.4 函数重载	85
第4章 数 组	58	5.4.1 函数重载的概念	85
4.1 一维数组	58	5.4.2 函数重载应满足的条件	85
4.1.1 一维数组的声明	58	5.4.3 函数重载的确定方法	86
4.1.2 一维数组的初始化	59	5.4.4 函数重载时应注意的问题	86
4.1.3 一维数组应用举例	60	5.5 作用域	86
4.2 二维数组	61	5.5.1 作用域的分类	86
4.2.1 二维数组的声明	61	5.5.2 变量的分类	87
4.2.2 二维数组的初始化	62	5.5.3 函数的分类	88
4.2.3 二维数组应用举例	62	5.6 系统函数	88
4.3 字符数组	63	5.7 应用举例	88
4.3.1 字符数组的定义	63	5.8 习题五	92
4.3.2 字符数组的引用与赋值	63	第6章 类与对象基础	96
4.3.3 字符串处理函数	64	6.1 面向对象程序设计基础	96
4.3.4 字符数组举例	66	6.1.1 什么是面向对象程序设计	96
4.4 指针和数组	67	6.1.2 面向对象程序设计的要素	96
4.4.1 指针和一维数组	67	6.2 定义类与对象	97
4.4.2 指针和二维数组	68	6.2.1 如何定义类	97
4.4.3 字符指针与字符串	70	6.2.2 如何定义对象	99
4.5 应用举例	71	6.3 对象的初始化	100
4.5.1 排序算法	71	6.4 成员函数	104
4.5.2 查找算法	73	6.4.1 成员函数的访问	104
4.6 习题四	74	6.4.2 析构函数	106
第5章 函 数	79	6.5 静态成员	107
5.1 函数的定义和声明	79	6.5.1 静态数据成员	107
5.1.1 函数定义和声明的区别及注意事项	79	6.5.2 静态成员函数	108
5.1.2 函数值及其类型	80	6.6 友 元	110
5.2 函数的调用	80	6.7 类的作用域与对象的生存期	112
5.2.1 函数调用的几种方式	80	6.7.1 类的作用域	112
5.2.2 在调用时形参和实参应注意的问题	81	6.7.2 对象的生存期	113
5.2.3 设置函数默认值的注意事项	83	6.8 习题六	113
5.2.4 函数的嵌套调用规则	84	第7章 类与对象的应用	116
5.2.5 函数的递归调用	84	7.1 类与指针	116
5.3 内联函数	85	7.1.1 使用指向对象的指针	116
5.3.1 内联函数引入的原因	85	7.1.2 使用指向类成员的指针	117
5.3.2 内联函数定义方法	85	7.1.3 使用 this 指针	119
5.3.3 使用内联函数注意事项	85	7.2 类与数组	120
		7.2.1 对象数组与普通数组的异同	120



7.2.2 对象指针数组与指针数组的关系	123	9.4.1 动态联编的实现条件	153
7.2.3 指向对象数组的指针与指向数组的 指针的比较	124	9.4.2 虚函数的使用原则	153
7.3 类中 const 关键词的使用	126	9.5 习题九	154
7.3.1 使用 const 修饰对象	126	第 10 章 C++ 的 I/O 流类库	157
7.3.2 使用 const 修饰类中的成员	126	10.1 标准输入和输出	157
7.4 子对象与堆对象的使用	127	10.1.1 输入输出流的控制符	158
7.4.1 子对象的初始化与使用	127	10.1.2 用于控制输入、输出格式的流成员 函数	160
7.4.2 堆空间与堆对象	129	10.1.3 write 和 read 函数	161
7.5 习题七	131	10.1.4 cin 与 cout	162
第 8 章 继承与派生	134	10.1.5 流成员函数 get() 和 put()	163
8.1 为什么使用继承	134	10.2 字符串流	164
8.2 继承的工作方式	134	10.2.1 ostream 类的构造函数	165
8.2.1 基类与派生类的概念及其关系	134	10.2.2 istream 类的构造函数	166
8.2.2 从基类中派生新类	135	10.3 磁盘文件的 I/O 操作	167
8.2.3 继承下的访问控制	136	10.3.1 磁盘文件的打开和关闭	168
8.3 派生类对象的初始化和撤销	138	10.3.2 流错误的处理	169
8.3.1 单继承下的构造函数和析构函数	138	10.3.3 文本文件的读和写	170
8.3.2 多继承下的构造函数和析构函数	140	10.3.4 二进制文件的读和写	173
8.4 虚基类的使用	141	10.4 习题十	179
8.4.1 定义虚基类	141	第 11 章 模 板	182
8.4.2 虚基类的初始化	142	11.1 函数模板和模板函数的区别	182
8.5 继承的使用原则	143	11.1.1 函数模板定义	182
8.5.1 类的组合	143	11.1.2 模板参数与调用参数	184
8.5.2 什么情况下使用组合	143	11.2 类模板与模板类	187
8.5.3 什么情况下使用继承	144	11.2.1 类模板的定义	187
8.5.4 类型兼容原则	144	11.2.2 模板类	190
8.6 习题八	145	11.3 习题十一	195
第 9 章 多态性	148	第 12 章 C++ 开发实例	197
9.1 理解多态性	148	12.1 需求分析	197
9.2 编译时多态性的函数重载	148	12.2 系统总体设计	197
9.3 编译时多态性的运算符重载	150	12.3 系统主要模块的设计与实现	198
9.3.1 运算符重载的形式	150	12.4 系统的软硬件环境	203
9.3.2 运算符重载的使用原则	152	12.5 系统的使用说明	203
9.4 运行时多态性的虚函数	152	12.6 程序框架代码	206
9.4.1 动态联编的实现条件	153	第 13 章 实验操作	215
9.4.2 虚函数的使用原则	153	实验一 Visual C++ 6.0 集成开发环境	215
9.5 习题九	154	实验二 数据类型、常量、变量	217



实验三 运算符与表达式(一)	221	实验十八 对象数组与指针	272
实验四 运算符与表达式(二)	223	实验十九 类的继承和派生	275
实验五 选择结构	227	实验二十 类的综合应用	278
实验六 循环结构	231	实验二十一 运算符重载	281
实验七 循环嵌套	235	实验二十二 静态联编和动态联编	285
实验八 一维数组	238	实验二十三 输入/输出流与文件的访问	289
实验九 二维数组与字符数组	240	实验二十四 综合实验	294
实验十 函数的定义及参数传递	243	习题参考答案	296
实验十一 函数递归及作用域	246	附录 A C++关键字列表	302
实验十二 内联函数及函数重载	253	附录 B C++常见错误提示	305
实验十三 类和对象定义	255	附录 C C++字符串操作函数列表	307
实验十四 构造函数与析构函数	260	参考文献	308
实验十五 友元函数与静态成员	264		
实验十六 指向类成员的指针	268		
实验十七 指针数组与数组指针	270		



第 1 章

C++语言概述

本章学习目标

1. 了解结构化程序设计和面向对象程序设计方法；
2. 了解 C++ 程序的特点，与其他程序设计语言的关系和区别；
3. 学会建立一个 C++ 程序的过程；
4. 熟练使用 C++ 上机环境。

1.1 C++发展简史

1.1.1 程序设计方法

1. 算法与程序有怎样的区别和联系

算法是求解问题的方法，通常由有限个步骤组成，对于给定的具体问题可以通过执行这些步骤得到具体的答案。算法具有有穷性、确定性、可行性、输入和输出的特性。算法可以用自然语言描述、流程图描述，也可以用伪代码或程序语言描述。

程序是对计算任务的处理对象和处理规则的描述。程序规定了计算机执行的动作和动作的顺序。程序包含对数据描述和对操作的描述。

2. 程序设计就是编码吗

有人认为程序设计是将算法用某种计算机语言表达出来。其实用具体的语言来描述算法是编码，而程序设计过程主要是完成求解问题的数据结构和算法的设计。

3. 程序设计语言有哪些

程序设计语言是用于书写计算机程序的语言。它不同于自然语言，是人工语言。程序设计语言有很多，按照其发展过程可分为机器语言、汇编语言和高级语言。

机器语言是面向机器的，是特定计算机系统所固有的语言，也就是说不同的计算机类型都有自己的表示成数码形式的机器指令集，每个指令完成一个特定的基本操作。编写机器语言程序需要对机器结构有较多的了解，而且用机器语言编写的程序复杂、可读性很差，修改和维护都很不方便。

汇编语言是机器语言的符号化，用有助记忆的符号来表示机器指令中的操作码和运算数。用汇编语言书写的程序相对机器语言来说，易于书写和记忆；但汇编语言也是面向机器的，不具有可移植性，且不能直接执行，需要把用汇编语言编写的源程序转换成机器语言程序，这个过程叫汇编，完成这一过程的程序叫汇编程序。

高级语言类似于人的自然语言，是与具体机器类型无关的语言。程序设计者不需要了解机器的内部结构，只要按照计算机语言的语法编写程序即可。所以高级语言易学、易用、易维护。但它也不能直接在计算机上执行，需要编译或解释之后才能执行。比较流行的高级语言程序有 C、C++、Java 和 Visual Basic 等等。



4. 程序设计方法有哪些

常用的程序设计方法有:结构化程序设计方法与面向对象程序设计方法。

结构化程序设计方法是建立在结构化定理(任何程序逻辑都可以用顺序、选择和循环三种基本结构来表示)的基础上,设计中采用“自顶向下,逐步求精”和“模块化”的思想,程序中所使用的数据和处理数据的方法是相互分离的。其优点是适合于小规模软件的开发;缺点是软件代码可重用性差,稳定性低,软件很难维护等。为了克服结构化程序设计的缺点,出现了面向对象的程序设计方法。

面向对象程序设计方法的基本思想是认为客观现实世界是由对象组成的,求解问题就是分析这个问题是由哪些对象组成的,这些对象之间如何联系和作用。面向对象程序设计方法中使用对象和类去分析问题,具有封装、继承和多态的主要特征。面向对象程序设计方法适合于大中型软件的开发,代码可重用性高、健壮性好、易于修改和维护。

1.1.2 C++发展简史

1. C与C++的关系

C和C++都诞生在Bell实验室。C语言是一种面向过程的语言,具有数据类型和运算符丰富、方便灵活、可移植性好、执行效率高等特点。C++源于C语言,与C语言兼容,是对C的扩展,既支持传统的结构化程序设计,保持了C语言的简洁和高效,又增加了类功能,支持面向对象的程序设计。

2. C++语言的发展过程

C++源于C语言。C语言在1972年诞生,20世纪80年代初把面向对象的思想引入C,最初称为“带类的C”,1993年正式定名为C++。

发展经历了三个阶段:

第一阶段从20世纪80年代到1995年,C++语言基本上是传统类型上的面向对象的语言。

第二阶段从1995年到2000年,由于标准模板库和Boost等程序库的出现,泛型程序设计在C++中的比重加大。

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++的又得到了迅速的发展,成为当今最复杂的主流程序设计语言之一。

3. C++语言有哪些特点

C++除了有C的特点外,还有支持面向对象的新特点,主要体现在如下几个方面:

(1) 使用类进行数据的封装。类和对象是面向对象的基本概念。类是对数据抽象和相关操作的封装,通过这样的封装可以将内部信息隐藏起来;对象是类的实例化,对象可以被声明为某个给定类的变量,对象之间可以通过调用类的方法来联系。

(2) 类中可以使用构造函数和析构函数。构造函数是在每次创建这个类的新对象时自动执行,作用就是对对象进行初始化。析构函数在这个类的对象被撤销时自动执行,作用是保证类的对象正常清除。

(3) 设置访问限制符来实现信息的隐蔽性。类的成员有三种访问权限:公有、私有和保护。公有成员可以被所有函数访问,私有成员由自己类的函数访问,保护成员可以由该类和其派生类的成员函数访问。

(4) 对象和消息。对象是面向对象程序设计的基本单元,对对象的操作通过向对象发送



消息实现,对象根据消息的内容调用相应的方法。

(5) 友元。在类定义中可以声明非成员函数或其他类的该类的友元函数或友元类,则这些友元就可以访问类的私有部分或保护部分。

(6) 运算符重载和函数重载,C++允许为已有的函数和运算符重新赋予新的含义,使它们可以用于用户所希望操作的对象。运算符重载和函数重载能够以更自然的表现方式来对对象的操作,从而提高程序的可读性。

(7) 继承和派生类,通过继承,可以使新类拥有父类的属性和行为,也可以根据需要对其进行修改和扩充。在类的继承中,被继承的类称为基类,新定义的类称为派生类。利用继承可以方便地利用已有的经过测试和调试的高质量的代码,提高开发的效率和质量。

(8) 虚函数的使用。C++中的虚函数可以支持动态联编,从而也支持多态。多态性允许在设计中使用高级抽象。它使得高层代码只需要书写一次,就可以通过提供不同的底层服务来满足复用的要求,可以大大提高代码的复用性。



1.2 C++程序实例

1.2.1 从一个简单的程序看 C++程序的组成

下面是一个简单的 C++ 程序。

/* 程序名称: c1_1.cpp

程序功能: 输入一个姓名, 然后在屏幕上输出 "Hello, ×××"

作者: 张桂香

时间: 2010年11月24日*/

程序注释

#include <iostream.h>

int main()

```
{
    char name[20];
    cout<<"Please input your name:";
    cin>>name;
    cout<<"Hello,"<<name<<"!";
    return 0;
}
```

//预处理语句

//主函数

//数据声明语句

//输出提示信息

//输入姓名

//输出信息

行注释

函数体

从上面的例子看,一个 C++ 源程序一般是由预处理部分、函数、数据声明、处理语句、输入输出和注释等组成。

1. C++ 程序中的注释方法和作用

第 1 行到第 4 行是程序的注释部分,以“/*”开始,以“*/”结束。所有注释的内容放在其间,中间可以换行,所以也叫多行注释,这是传统 C 的注释方式。C++ 的注释一般采用“//”开头的方式,如第 9 行到第 11 行后面的以“//”开头的,后面的文字也是注释信息,一直到行尾结束。无论哪种注释方式都是对程序起解释说明作用,只是为了增强程序的可读性,不是必须的,对于程序的执行不起任何作用。但是给程序添加必要的注释是一种良好的程序设计风格。

2. 预处理的作用

第 5 行是预处理语句,以符号“#”开头,编译预处理的作用是在编译之前将文件 iostream.h



增加到本程序中,以便使用其中的输入输出流相关的类等。

3. 函数的组成

第 6 行是 `main()` 函数。每个 C++ 程序都是由一个或多个函数组成,其中有且只有一个主函数,即 `main()` 函数,它是整个程序执行的入口。`main` 前面的 `int` 表示函数的返回值是一个整数。`main` 是函数名,函数名后面必须是一对圆括号。括号里面可以根据需要有 0 到多个参数。第 7 行的左花括号“{”和第 13 行的右花括号“}”是 `main()` 函数的函数体的开始和结尾。第 8 行到 12 行就是函数体,用来实现函数的功能;函数体也可以是空的,什么也不做。第 12 行是函数的返回语句 `return`,一般放在函数的最后一行,返回程序的执行结果;因为前面定义 `main()` 函数是 `int` 类型,所以此处用 `return` 后面值为 0,表示程序正常结束。

4. C++ 的语句

语句是 C++ 程序中最小的可执行单元。一条语句由一个分号结束。语句既可以很简单,也可以很复杂。如第 8 到 12 行都是 C++ 语句,第 8 行是变量声明语句,而第 5 行预处理不是 C++ 语句,所以结束没有分号。

5. 输入输出的方法

第 9 行是一个输出语句,`cout` 代表一个标准输出流设备。它是在 `iostream.h` 中预定义的对象。如果前面没有预处理,此处就会出错。“<<”是输出操作运算符,它表示将运算符右边的数据送到输出设备。这里的标准输出设备就是显示器。“<<”右边的就是要输出的字符串常量,用双引号括起来。

第 10 行是一个输入语句,`cin` 代表一个标准输入流设备。它也是在 `iostream.h` 中预定义的对象。“>>”是输入操作运算符,它表示将标准输入设备的内容输入到运算符右边的 `name` 数组中。这里的标准输出设备就是键盘。

第 11 行也是一个输出语句,由三段组成,先输出“Hello”,然后输出 `name` 变量中的内容,最后输出“!”。

1.2.2 C++ 字符集和关键字

1. C++ 字符集

字符是可以区分的最小符号,C++ 字符包括:

- (1) 大小写英文字母: `A~Z, a~z`
- (2) 数字字符: `0~9`
- (3) 特殊字符: 空格 ! ? # % ^ & * _ + - ~ < > / \ " ; , () { } []

2. C++ 标识符

标识符是用来表示常量、变量、函数、数组或者类型等实体名字的有效字符序列。C++ 标识符命名规则:

- (1) 标识符必须由字母或者下划线开头,后面可以是字母、数字或者下划线。
- (2) 标识符中英文大小写是区分的。
- (3) 标识符只有前 32 个字符有效。
- (4) 用户定义的标识符不能使用系统关键字。

3. C++ 关键字

关键字又称保留字或者保留关键字,是 C++ 语言预定义的具有特殊意义的标识符,如表



示数据类型、变量的属性等,用户在标识符命名时不能使用。表 1.1 列出 C++ 中的关键字。

表 1.1 C++ 的关键字

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

1.2.3 书写规则和程序设计风格

1. 良好的程序设计风格的重要性

程序设计风格是指一个人编制程序时所表现出来的特点、习惯和思路等。在程序设计中要使程序结构合理、清晰,形成良好的编程习惯,对程序的要求不仅是可以在机器上执行,给出正确的结果,而且要便于程序的调试和维护,这就要求编写的程序不仅自己看得懂,而且也要让别人能看懂。良好的程序设计风格是高水平程序设计员的必备素质。

2. 养成良好的程序设计风格的基本规范

C++ 语言程序书写非常灵活,同一程序可以有多种方式写程序,但是当今主导的程序设计风格是“清晰第一,效率第二”的观点。

(1) C++ 严格区分英文字母的大小写。

(2) 一行一般写一条语句。短语句可以一行写多个,每条语句以分号结束。长语句可以一条写多行。分行原则是不能将一个单词分开。用双引号引用的一个字符串也最好不分开。如果一定要分开,有的编译系统要求在行尾加续行符(“\”)。

(3) 标识符应按意取名。

(4) 程序应适当加注释。注释方便以后对程序的理解和维护。

(5) 缩进形式显示程序结构,使用一致的缩行和加括号风格。人们常用的格式形式是:逻辑上属于同一个层次的互相对齐;逻辑上属于内部层次的推到下一个对齐位置。

(6) 恰当地使用空格、空行以改善清晰度,空行起着分隔程序段落的作用。空行得体(不过多也过少)将使程序的布局更加清晰。



1.3 C++ 上机环境

1.3.1 C++ 程序的开发过程

开发 C++ 程序的步骤通常包括编辑、预处理、编译、连接、运行和调试。

(1) 编辑:是程序开发过程的第一步,主要包括程序文本的输入和修改。编辑通过编辑器来完成,C++ 程序的源代码通常是存储在扩展名为 .cpp 的文件中,如 cl_1.cpp。

(2) 编译预处理:在编译前将要包含其他文件增加到程序中。

(3) 编译:编译器完成将一个扩展名为 .cpp 的源程序(如 cl_1.cpp)文件转换成一个以 .obj 为扩展名的目标文件(如 cl_1.obj),即机器指令代码文件。



(4) 连接:连接是将目标文件与程序的库文件链接起来,形成一个可以在操作系统中直接运行的可执行程序,扩展名为.exe(如 cl_1.exe)。

(5) 运行:控制执行该程序,一般就可以看到程序的运行结果。

(6) 调试:如果源文件存在语法错误或链接错误,编译器和链接器会分别给出相关的错误信息,这时需要重新编辑源文件,改正其中的错误,再一次进行编译和链接,直到错误信息不再出现为止。如果程序运行结果与预期不同,也需要重新修改源程序。

具体过程参见图 1.1。

1.3.2 C++集成开发环境有哪些

集成开发环境就是将源程序的编写、编译、连接、调试和运行以及应用程序的文件管理等集成在一起,给开发程序带来很大的方便。C++的集成开发环境有很多,比较有名的有 Microsoft Visual C++、Borland 公司的 C++ Builder、IBM 公司的 Visual Age for C++ 等。本书使用的是 Visual C++ 6.0,它是 Windows 操作系统下最流行的 C++开发环境之一。

1.3.3 Visual C++和 C++的关系

学习高级语言包括基本语言理论和开发工具的使用。Visual C++是集成开发环境,是集程序编辑器、编译器、调试工具和其他建立应用程序的工具于一体的用于开发应用程序的软件系统。在 Visual C++下可以运行 C 语言程序,也可以运行 C++程序。C++是高级程序设计语言,掌握 C++语言基础是学习好程序设计的前提,而初学语言时,一定要熟悉开发环境,才能顺利开发程序。

1.3.4 Visual C++ 6.0 介绍

微软公司 Visual C++ 6.0 有三个版本:标准版、专业版和企业版,不同的版本适合于不同类型的应用开发。

1. Visual C++ 6.0 的安装

Visual C++ 6.0 可以独立安装,也可以与 Visual Studio 同时安装。

2. Visual C++ 6.0 的启动

打开 Visual C++ 6.0 | “开始” | “所有程序” | Microsoft Visual Studio 6.0 | Microsoft Visual C++ 6.0, 出现 Microsoft Visual C++窗口,如图 1.2 所示。

3. Visual C++ 6.0 工作界面

主窗口中包括标题栏、菜单栏、工具栏、工作区窗口,代码编辑窗口、输出窗口和状态栏组成。刚打开 C++而没有启动任何文件时,窗口中只有标题栏、菜单栏、工具栏和状态栏,随着打开文件时,其他窗口才会出现。

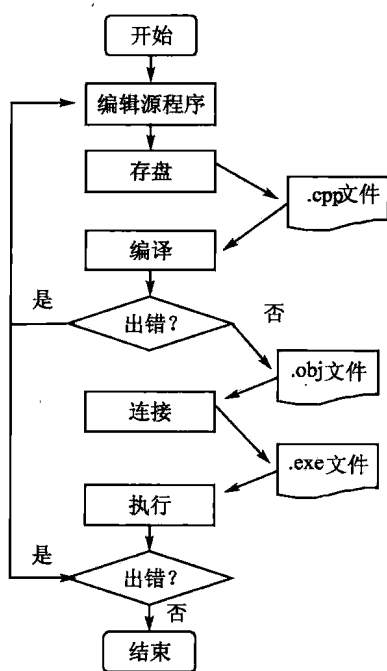


图 1.1 C++程序的开发过程

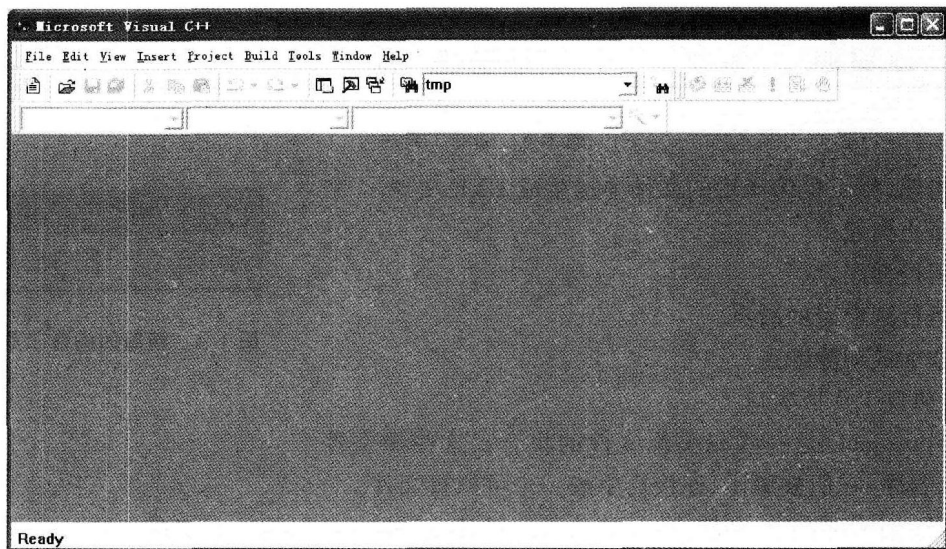


图 1.2 Microsoft Visual C++ 窗口

(1) 菜单栏

菜单栏中的菜单主要有以下几项。

File(文件):用于进行和文件有关的操作,如创建(New)、打开(Open)、保存(Save)和关闭(Close)文件等。

Edit(编辑):用于进行和编辑有关的操作,如复制(Copy)、剪切(Cut)、粘贴(Paste)、查找(Find)、替换(Replace)、删除>Delete)和恢复(Undo)等。

View(视图):用于激活所需要的各种窗口,如工作区窗口(Workplace)、输出窗口(Output)、调试窗口(Debug Windows)和属性窗口(Properties)等。

Insert(插入):用于插入新建类(New Class)、资源(Resource)等。

Project(工程):用于和工程有关的操作,如设置活动工程(Set Active Project)、向工程添加文件(Add to Project)、插入工程到工作区(Inset Project into Workspace)等。

Build(编译):用于和程序的编译有关的操作,如编译(Compile)、构建(Build)、调试(Debug)和运行(Execute)等。

Tools(工具):用于选定或定制集成开发环境中的一些工具,如浏览用户程序中定义的符号(Source Brower)、激活常用工具、更改选项和变量的设置等。

Window(窗口):用于分割窗口、隐藏和显示窗口等。

Help(帮助):帮助用户学习和使用 C++。

(2) 工具栏

在 Visual C++ 中工具栏很多,常用的有标准工具栏、编译工具栏和精简的编译工具栏、编辑工具栏、调试工具栏等。如图 1.3 所示标准工具栏和其他 Windows 应用程序类似,这里不一一叙述。图 1.4 所示是精简的编译工具栏。

下面介绍编译工具栏中各个按钮的功能。

 编译源程序,生成 OBJ 文件。