

21 世纪高等学校计算机(基础)实用规划教材

C/C++ 高效率教程

——用自然语言思想递进学习C/C++

李春庚 编著

清华大学出版社



21世纪高等学校计算机**基础**实用规划教材

C/C++高效率教程

——用自然语言思想递进学习C/C++

李春庚 编著

清华大学出版社

北京

内 容 简 介

本书是《基于自然语言思想的递进 C/C++ 教程》上下册的合订版,虽然结构上前半部分是 C 语言,后半部分是 C++,但逻辑思想上以自然语言“选词造句,连句成段,多段成章,词不断丰富,句式不断变化”的递进学习思想贯穿全书。本书将 C 语言中的类型和量值(常量和变量)理解为“名词”,各种运算符理解为“动词”,构成分支结构的 if...else、构成循环结构的 while 等理解为“连词”,由名词、动词或连词的有机组合就形成了 C 语言的语句;而将数组和结构体等类型和变量理解为集合名词,是基本名词的横向发展,将指针、位、函数和文件理解为名词的纵向发展,简洁深入地阐述了 C 语言的概念和语法。本书在 C 语言的“名词”——结构体基础上,进一步扩展到 C++ 语言的“标志性名词”——类和对象;将 C 语言中的指针和数组应用于 C++ 的类对象,建立起 C 和 C++ 的连贯性;根据自然语言中同一个名称或动词在不同的语境下有不同的意思,要根据上下文来理解的规律,讲解了 C++ 语言中运算符的重载;根据 C 语言中名词的空间域和时间域,深入阐释了 C++ 中类的封装、继承和派生。而模板、流、命名空间、标准模板库更是名词、动词及连词的时间空间作用域融合发展的结果。

全书围绕“学生信息管理系统”一个例题程序展开,随着不断深入学习,不断完善系统的功能,使之接近实际开发。每章的习题都是例题的变化或功能的增强,以加强学生对知识的理解,锻炼程序开发能力,不但可以学到 C/C++ 的语法,更可以开发实际应用程序。

本书可作为普通高校 C/C++ 语言课程的教材,也可作为培训教材和自学教材。

本书封面贴有清华大学出版社防伪标签,无标签者不得销售。

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/C++ 高效率教程:用自然语言思想递进学习 C/C++/李春庚编著.--北京:清华大学出版社,2015
21 世纪高等学校计算机基础实用规划教材
ISBN 978-7-302-38390-1

I. ①C… II. ①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(2014)第 250890 号

责任编辑:付弘宇 王冰飞

封面设计:常雪影

责任校对:李建庄

责任印制:宋 林

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 刷 者:北京富博印刷有限公司

装 订 者:北京市密云县京文制本装订厂

经 销:全国新华书店

开 本:185mm×260mm

印 张:27.75

字 数:696 千字

版 次:2015 年 2 月第 1 版

印 次:2015 年 2 月第 1 次印刷

印 数:1~2000

定 价:49.50 元

产品编号:061780-01

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、教学内容和教学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。

本系列教材立足于计算机公共课程领域,以公共基础课为主、专业基础课为辅,横向满足高校多层次教学的需要。在规划过程中体现了如下一些基本原则和特点。

(1) 面向多层次、多学科专业,强调计算机在各专业中的应用。教材内容坚持基本理论适度,反映各层次对基本理论和原理的需求,同时加强实践和应用环节。

(2) 反映教学需要,促进教学发展。教材要适应多样化的教学需要,正确把握教学内容和课程体系的改革方向,在选择教材内容和编写体系时注意体现素质教育、创新能力与实践能力的培养,为学生的知识、能力、素质协调发展创造条件。

(3) 实施精品战略,突出重点,保证质量。规划教材把重点放在公共基础课和专业基础课的教材建设上;特别注意选择并安排一部分原来基础比较好的优秀教材或讲义修订再版,逐步形成精品教材;提倡并鼓励编写体现教学质量和教学改革成果的教材。

(4) 主张一纲多本,合理配套。基础课和专业基础课教材配套,同一门课程可以有针对不同层次、面向不同专业的多本具有各自内容特点的教材。处理好教材统一性与多样化,基本教材与辅助教材、教学参考书,文字教材与软件教材的关系,实现教材系列资源配套。

(5) 依靠专家,择优选用。在制定教材规划时依靠各课程专家在调查研究本课程教材建设现状的基础上提出规划选题。在落实主编人选时,要引入竞争机制,通过申报、评审确定主题。书稿完成后要认真实行审稿程序,确保出书质量。

繁荣教材出版事业,提高教材质量的关键是教师。建立一支高水平教材编写梯队才能保证教材的编写质量和建设力度,希望有志于教材建设的教师能够加入到我们的编写队伍中来。

21 世纪高等学校计算机基础实用规划教材

联系人:魏江江 weijj@tup.tsinghua.edu.cn

前言

1. 本书编写背景

本书是应广大读者要求,将《基于自然语言思想的递进 C/C++教程(上册 C)》和《基于自然语言思想的递进 C/C++教程(下册 C++)》合并而成,修正了使用中发现的错误和不足,精炼了讲解语言和思路,是两本书的合订版。

C语言是应用面广、效率高的结构化高级计算机语言。C++语言是在C语言基础上发展起来的、面向对象的高级计算机语言,它不但继承了C语言的所有优点,兼容C语言的所有语法,更增强了安全性高、适应性强和编程效率高的特点。通常将C语言和C++语言合并,泛称C++语言。C++语言也成为全世界多数大学理工类专业计算机编程训练的首选教学语言。

怎样才能通过教和学,高效率地掌握C++语言呢?当前有以下两种教学模式。

(1) 先学习C语言,然后学习C++语言。这种教学模式知识内容掌握得比较深入,但耗费的时间比较长,效率不高。另外,容易带来“结构化”和“面向对象”两种编程思想的对立。因为在C语言的教材中,通常只讲结构化的编程,而C++教材通常强调面向对象的优点及其与结构化思想的不同,忽略了“结构化”和“面向对象”的联系,及“面向对象”是如何从“结构化”发展而来的。

(2) 模糊C语言和C++语言的界限,直接学习C++语言。这种教学模式效率比较高,耗费的时间较短,但教学效果往往不够理想。因为面向对象编程程序语句排列的先后顺序和程序执行的顺序不一致,具有“超文本的跳跃性”,对于没有结构化程序设计基础的初学者理解难度较大。

为了克服以上两种教学模式的不足,我们寻着计算机编程语言产生发展的脉络,提出了“基于自然语言思想的递进 C/C++教学”思路。因为计算机编程语言从机器语言到汇编语言再到高级语言的发展过程,就是越来越逼近人类自然语言的过程;从C语言到C++语言的发展过程,就如同人类自然语言不断丰富的发展过程,它是一个递进的补充、完善过程。

人类自然语言是由词构成短语,多个词或短语构成句子,多个句子形成段落,多个段落组成文章。基于人类自然语言的思想将C语言中的类型、常量和变量看做名词,将各种运算符看做动词,将if...else、switch...case、while等看做连词。表达式就是短语,程序语句就是句子,程序模块就是段落,整个工程的程序就是文章。而数组、结构体、函数和指针是名词的发展;语句嵌套、函数的递归等是句式的发展;沿着这样的理解思路,C++中的类自然是C中结构体类型的发展;类的对象是结构体变量的发展;当以类和对象为着眼点编程时,是面向对象的编程,但类的构建过程,即类的内部实现却是结构化的。这样就清晰了从C发展到C++的过程,找到了贯穿C和C++的一致性知识框架体系,使C语言和C++语言无缝

衔接。使学生在在学习过程中不用区分 C 和 C++，更多地学习 C 和 C++ 的递进性、一致性，从而在课程结束后可以深入地理解 C 和 C++ 的不同，领略 C++ 对 C 的补充和增强，深入理解 C 语言和 C++ 语言的知识内容，真正做到融会贯通，培养编程实践能力。

本书讲稿及上下册已经在大连海事大学的教学中使用多年，教学效果显著。

2. 本书特色

(1) 以自然语言的思想讲授计算机语言，形象易懂。

(2) 内容紧凑，中心突出，练习题通常是例题的发展和变化，少而精，非常适合于课堂教学和初学者自学。

(3) 用一个“学生信息管理系统”例题贯穿本书始末，将学习的各种语法递进加入例题，使之不断完善、丰富，一步步接近实际开发。

(4) 引入四则运算数学方法，讲解二维数组指针难点，使语言的学习成为简单的推导，深入而准确。

(5) 提出表达式、函数是复合的变量，常量是变量的“瞬间快照”等观点，蕴涵了“泛化”的思想，潜移默化训练学生的思维方式，为 C++ 的泛型程序设计打下思想基础。

(6) 发展 C/C++ 语言中的名词，得到类、对象、模板和命名空间；发展 C/C++ 语言中的动词，得到运算符的重载；各种词性融合发展便得到标准模板库。

(7) C++ 部分的例题是以面向对象的编程方法对 C 结构化编程例题的重新编写和进一步补充、发展，充分展示了结构化编程和面向对象编程的区别和联系。

感谢大连海事大学信息科学技术学院计算机系“C/C++ 语言课程组”的史一民老师、刘宁老师、黄健老师、刘亚清老师、于纯妍老师、宋梅萍老师和郭浩老师给予的大力支持，感谢遥感信息研究所的安居白教授给予的大量帮助。

感谢读者选择本书，欢迎您对本书内容提出批评和修改意见，我们将不胜感激。作者联系地址如下：

电子邮件地址：licgsh@163.com；licgsh@sina.com；licgsh@dlmu.edu.cn

通信地址：辽宁省大连市大连海事大学信息科学技术学院计算机系李春庚

邮政编码：116026

作者

2015 年 1 月

目 录

第 1 章 概述	1
1.1 自然语言与计算机语言	1
1.2 计算机语言、C 语言及 C++ 语言	2
1.3 简单的 C/C++ 程序及其运行	2
1.3.1 C/C++ 程序开发运行环境	2
1.3.2 格式化输出输入函数 printf 和 scanf	6
习题 1	9
第 2 章 基本的 C/C++ 语言	10
2.1 C/C++ 语言中的名词——类型、量值	10
2.1.1 整型和整型量值	10
2.1.2 浮点型和浮点量值	12
2.1.3 字符型和字符量值	13
2.1.4 字符串常量	14
2.2 C/C++ 语言中的动词——运算符、短语——表达式和特殊动词性关键字	15
2.2.1 赋值运算符和赋值表达式	15
2.2.2 算术运算符和算术表达式	16
2.2.3 逻辑值、逻辑运算符和逻辑表达式	17
2.2.4 关系运算符和关系表达式	18
2.2.5 其他运算符和表达式	19
2.2.6 不同类型量值的不同运算的混合	22
2.3 C/C++ 语言中的连词及句子——分支、循环和顺序结构	25
2.3.1 顺序结构	25
2.3.2 if...else——分支结构	25
2.3.3 switch...case——分支结构	27
2.3.4 ? :——分支结构	29
2.3.5 for——循环结构	29
2.3.6 while——循环结构	30
2.3.7 do...while——循环结构	31
2.3.8 goto...label...if——循环结构	32

2.3.9	break——终止循环或开关	34
2.3.10	continue——终止本次循环	34
2.4	C/C++ 语言中的句子小结	35
2.5	用 C/C++ 语言编写程序——段落	35
2.6	用 C/C++ 语言编写复杂段落——语句嵌套	36
2.6.1	语句嵌套的含义	36
2.6.2	嵌套语句的示例	37
2.6.3	语句嵌套的小结	40
2.7	本章总结	41
	习题 2	42

第 3 章 C/C++ 语言的横向名词性扩充 43

3.1	同类变量的集合——数组	43
3.1.1	一维数组	43
3.1.2	二维数组	47
3.1.3	n 维数组	50
3.1.4	字符数组	51
3.2	自定义类型——typedef	56
3.2.1	简单类型的自定义	56
3.2.2	同类集合类型的自定义	56
3.3	异类集合类型和异类集合变量	57
3.3.1	结构体类型和结构体变量	57
3.3.2	共用体类型和共用体变量	60
3.3.3	枚举类型和枚举变量	61
3.4	异类集合类型的同类集合类型——结构体数组	63
3.5	各种横向扩充类型的融合使用	65
	习题 3	68

第 4 章 C/C++ 语言的纵向名词性扩充 1: 指针 69

4.1	简单指针类型和指针量值	69
4.1.1	变量的三层含义	69
4.1.2	变量的指针和指针类型的变量	69
4.1.3	与指针相关的动词补充——& 和 * 运算符	70
4.1.4	指针和指针变量示例	71
4.2	同类集合变量——数组的指针和指针变量	73
4.2.1	一维数组的指针和操作一维数组的指针变量	73
4.2.2	二维数组的各级指针	77
4.2.3	操作二维数组的各种指针变量	81
4.2.4	m 维数组的指针和指向 m 维数组的指针变量	84

4.3 指向指针变量的指针量值——二级指针	85
4.3.1 二级地址和二级指针变量	85
4.3.2 各种复合的二级指针及多级指针	85
习题 4	92
第 5 章 C/C++ 语言的纵向名词性扩充 2: 函数	94
5.1 函数的构成	94
5.1.1 函数的定义	94
5.1.2 函数的声明和调用	95
5.1.3 函数的参数和返回值	95
5.2 不同参数和不同返回值形式的各种函数	96
5.2.1 无返回值函数	96
5.2.2 参数或返回值是异类集合量值的函数	97
5.2.3 参数是简单指针量值的函数	98
5.2.4 参数或返回值是异类集合类型的指针量值的函数	100
5.2.5 参数是和一维数组名对应的指针量值的函数	103
5.2.6 参数是和二维数组名对应的指针量值的函数	104
5.2.7 返回值是指针量值的函数	105
5.3 函数的多级调用	108
5.3.1 函数的嵌套调用	108
5.3.2 函数的递归调用	109
5.4 函数的指针和指向函数的指针变量	111
5.4.1 函数的指针	111
5.4.2 指向函数的指针变量	111
5.5 main 函数的参数	114
5.6 动态内存管理函数	116
习题 5	117
第 6 章 名词的空间域、时间域和物理存储位置	119
6.1 C/C++ 程序的逻辑结构	119
6.2 名词性的类型和量值在各逻辑结构上的空间域	120
6.2.1 模块域	120
6.2.2 嵌套模块的层域	121
6.2.3 文件域	122
6.2.4 工程域	124
6.2.5 工作区域	126
6.3 名词性的类型和量值在各逻辑结构上的时间域	126
6.3.1 模块内的 static	126
6.3.2 文件内的 static	128

6.3.3 关于 static 的总结	128
6.4 变量的物理存储位置	129
习题 6	129
第 7 章 编译预处理	130
7.1 简单替换宏——符号常量	130
7.2 复杂替换宏——宏函数	132
7.3 批量的嵌入——文件包含	134
7.4 条件编译	134
7.4.1 #ifndef... #else... #endif	134
7.4.2 #ifdef... #else... #endif	140
7.4.3 #if... #else... #endif	140
7.4.4 assert 故障诊断	140
7.4.5 字符串替换——#	141
7.4.6 数字连接——##	142
习题 7	142
第 8 章 C/C++ 语言的纵向名词性扩充 3: 宏观扩充——文件	143
8.1 输入输出流和文件的指针	143
8.1.1 输入输出流	143
8.1.2 文件的指针	143
8.2 文件的打开和关闭	144
8.2.1 文件的物理形式	144
8.2.2 文件的打开	144
8.2.3 文件的关闭	145
8.3 文件的顺序读写	146
8.3.1 向文件中顺序输出(写)字符	146
8.3.2 从文件中顺序输入(读)字符	147
8.3.3 文件的顺序多个数据读写	148
8.3.4 文件的格式化输入和输出	152
8.3.5 文件的字符串读写	152
8.4 文件内的定位和随机读写	154
8.4.1 检测文件位置指针	154
8.4.2 置文件位置指针于文件头	154
8.4.3 置文件位置指针于指定位置	154
8.5 文件的出错控制	157
8.5.1 检查是否到达文件尾 feof()	157
8.5.2 检查当前文件操作是否出错 ferror()	157
8.5.3 清除文件错误标识 clearerr()	158

习题 8	158
第 9 章 C/C++ 语言的纵向名词性扩充 4: 微观扩充——位和位域	159
9.1 位运算	159
9.1.1 位运算符和位运算	159
9.1.2 位运算举例	161
9.2 位域	162
习题 9	164
第 10 章 C++ 给 C 语言打的补丁	165
10.1 补丁 1——C++ 的基本输入输出	165
10.2 补丁 2——代词引用 &	166
10.3 补丁 3——形容词 const	168
10.3.1 符号普通常量	168
10.3.2 符号引用常量	169
10.3.3 符号常量指针	171
10.3.4 符号指针常量	173
10.4 补丁 4——参数带有默认值的函数	174
10.5 补丁 5——函数的重载	176
10.6 补丁 6——函数的内联	178
10.7 补丁 7——内存的动态分配和释放	179
习题 10	182
第 11 章 C++ 的标志性新增名词: 类和对象	183
11.1 类和对象的解析	183
11.1.1 类的 C 语言根源	183
11.1.2 类的 C++ 语言创建	184
11.1.3 对象的定义	186
11.1.4 类的名词作用域审视	189
11.1.5 类数据成员的说明	189
11.1.6 类成员函数的说明	191
11.2 类对象的指针、引用及数组的应用	193
11.3 类的静态数据成员和静态成员函数	197
11.4 类对象的初始构造和析构	202
11.4.1 析构函数	203
11.4.2 构造函数	203
11.4.3 复制构造函数	207
11.4.4 类内包含指针变量及内存动态分配——深构造对象	210
11.5 对类作用域的破坏——友元	214

11.5.1	类的友元函数	215
11.5.2	类的友元类	218
11.5.3	对象的 this 指针	222
11.6	与类和对象相关的 const	225
11.6.1	类内的常数据成员	225
11.6.2	类内的常成员函数	227
11.6.3	常对象	230
11.7	类内的其他类对象——子对象	239
	习题 11	244
第 12 章	C++ 的动词扩充：类作用域内的运算符和类型重载	245
12.1	运算符重载	245
12.1.1	将运算符重载为类的成员函数	246
12.1.2	将运算符重载为类的友元函数	247
12.2	类型重载	253
	习题 12	255
第 13 章	C++ 标志性名词(类)的发展 1——类作用域的嵌套(继承和派生)	256
13.1	类作用域的单路线嵌套(单继承和派生)	257
13.1.1	类作用域的单路线结构化嵌套	257
13.1.2	类的单路线交叉嵌套(继承派生中的交叉问题)	273
13.1.3	类作用域单路线嵌套中的二义性问题	275
13.1.4	基类子对象的提取	279
13.1.5	类的单路线继承派生中的构造和析构函数	283
13.2	类作用域的多路线嵌套(多继承和派生)	287
13.2.1	类作用域多路线嵌套的单路线嵌套解析	287
13.2.2	类作用域多路线嵌套带来的二义性问题	288
13.2.3	多父类具有共同最基类时多义性问题的解决	289
13.2.4	多父类具有共同最基类时子类对象的创建	291
13.3	同一类家族众对象的综合管理问题——虚函数	299
13.3.1	用指向基类的指针变量来统一管理类家族的众对象的问题	299
13.3.2	用基类指针调用派生类中定义的与基类中同名的函数	306
13.3.3	用基类指针调用派生类中新定义的函数	314
13.3.4	基类中虚空函数的发展——纯虚函数和虚基类	321
13.3.5	用基类指针释放派生类中动态分配的内存空间——虚析构函数	328
	习题 13	346
第 14 章	C++ 标志性名词(类)的发展 2——模板	347
14.1	函数模板	348

14.1.1	函数模板的定义和应用	348
14.1.2	函数模板的函数特性	351
14.2	类模板	353
14.2.1	类模板的定义和应用	353
14.2.2	类模板和类的关系	356
14.2.3	类模板类型的对象的各种形式作为函数的参数及类模板的友元函数模板	357
14.2.4	类模板中的静态成员	360
14.2.5	类模板的继承和派生	363
习题 14		367
第 15 章 基于模板的文件操作——流		368
15.1	计算机操作文件的内在逻辑	368
15.2	流的概念和流类库	368
15.3	标准流对象及其操作	370
15.3.1	标准输入流对象及其操作方法	371
15.3.2	标准输出流对象及其操作方法	373
15.3.3	标准流操作示例	374
15.4	基于流类的磁盘文件操作	389
15.4.1	文本文件的操作	391
15.4.2	二进制文件的操作	395
习题 15		397
第 16 章 C++表示有效范围的名词发展 3——命名空间		398
16.1	命名空间的作用	398
16.2	命名空间的定义和使用	399
16.2.1	命名空间的定义	399
16.2.2	命名空间的使用	399
习题 16		415
第 17 章 C++对预想不到的运行错误控制——异常处理		416
17.1	异常处理的作用	416
17.2	异常处理的语句规则	417
17.3	异常处理示例	418
习题 17		422
第 18 章 C++的名词、动词融合发展 4——标准模板库		423
18.1	标准模板库的引出	423
18.2	标准模板库中的容器(类)	423

18.3	标准模板库中的迭代器	425
18.4	标准模板库中的容器适配器	426
18.5	标准模板库中的泛型算法	427
习题 18		427
附录 A	ASCII 码表	428
参考文献		429

第1章

概 述

1.1 自然语言与计算机语言

自然语言有着悠久的历史,它是在人类漫长的历史中自然发展、约定俗成的。它的构成单位由小到大可以粗略地分为字、词、短语、句子、段落和篇章等。在英文中没有字的概念,最小的单位是词(本书统称为字词)。人们学习语言的过程是先学习简单的字词,通常是名词、动词和连词,如车、雨、乘、如果……那么……等,这些字词已经在漫长的历史中被固定,无法改变,如果你非要改变,那么只能是属于你自己的语言,无法和别人交流沟通。对简单的字词有了一定的掌握后,两个以上的字词就构成短语,如出租车、下雨等;短语是有一定自由度的,可以适当变化。在短语的基础上,配合连词可以说出“如果下雨,那么就乘出租车”的句子,句子的灵活性就更大了。当然,如果一个人掌握了很多的句子,就可以将句子加以组合,形成段落和篇章。随着社会的发展和社会实践的需要,人们可以不断创造出新的字词,使语言不断丰富。这是每个人学习自然语言的过程。但是,必须明确,熟练掌握一种自然语言并不是说就可以写出优美的文章。因为文章中不但要有正确的字词、短语和句子,更要有深邃的思想,而思想不属于语言本身,只是以语言作为表达的载体。思想属于某个专门的领域,需要人们对该领域知识的深刻理解和掌握。这就是说,语言只是人们工作于社会中的基本技能,在此基础上,需要对某一领域的掌握和理解,然后更要将语言恰当地应用于该领域中,用语言表达该领域的思想,解决该领域的问题。另外,语言是非常依赖实践的,只是听别人说而自己不说,只是阅读别人写的文章而自己并不写,是永远不可能掌握语言的。

计算机语言的构成及学习过程和人类的自然语言是非常相似的,也包含“名词”,在计算机语言中通常叫类型和量值,量值包括常量和变量;“动词”,在计算机语言中叫运算符;“连词”,在计算机语言中叫控制流程;构成的“短语”在计算机语言中叫表达式;用表达式构成的“句子”在计算机语言中叫语句;相当于自然语言的“段落”叫程序段或模块;和“篇章”相对应的就是程序了。计算机语言发展到今天,已经有了和“著作”相对应的单位,那就是工程。学习计算机编程语言的过程与学习自然语言的过程是一致的。先要学会类型的定义,量值的声明,表达式的创建,程序语句的书写,以及模块和程序的编制。完全类似的是,并不是掌握了计算机语言就成为了编程高手。计算机的程序中同样必须蕴涵着思想,用计算机的术语叫算法,它不属于计算机语言本身,它属于要解决问题的专业领域。只有熟练掌握了计算机语言,又真正理解了要解决的专业问题,才能编写出可以解决问题的、有一定规模的计算机程序。当然,对于很多类似的问题有通用的算法,这是算法课要学习的内容。程序设计课是介绍一种计算机语言的基本格式和语法规则,以及如何用该种计算机语言与

计算机进行交流。该课程通常不涉及复杂的算法,只是讲授语言的规则和思路。如果和汉语文课相类比的话,程序设计课相当于小学到初中的语文,强调的是字词的正确,语句的通顺,意思简单,但完整、正确,重点不放在主题的确立和文章的布局谋篇。

1.2 计算机语言、C 语言及 C++ 语言

计算机语言有好多种,各有特点,适合不同的应用需要。但 C 和 C++ 语言蕴涵了所有计算机语言思想和方法。所以,全世界的大学里在做计算机语言的基础训练时都是用 C 或 C++ 语言。C 语言是结构化程序设计思想的代表;C++ 语言是面向对象程序设计思想的代表。面向对象程序设计思想是建立在结构化程序设计思想基础上的。C 语言是 C++ 语言的子集,所有 C 语言的关键字、表达式、语句、函数等在 C++ 语言环境中都能够被正确识别。从 C 语言开始,像学习自然语言那样,通过递进地学习,可以很自然地过渡到 C++ 语言。熟练掌握 C 和 C++ 语言后,理论上可以很容易掌握其他各种计算机语言。本书通常提“C/C++ 语言”,而不强调 C 语言,因为 C 语言是 C++ 语言的一部分。

1.3 简单的 C/C++ 程序及其运行

1.3.1 C/C++ 程序开发运行环境

怎样才能使用计算机编写一段 C/C++ 语言的程序并使之运行起来,输出结果呢?

(1) 要安装一套用来开发 C/C++ 程序的软件。用来开发 C/C++ 程序的软件很多,通常叫做程序开发环境。如 Microsoft Visual C++、Turbo C/C++、Bland C/C++ 等。本书以 Microsoft Visual C++ 6.0 为例来讲解。它是当前使用广泛、功能强大、操作方便的微机 C/C++ 开发环境。

(2) 在“我的电脑”或“资源管理器”中创建工作文件夹,以后编写的程序和结果都存储在此处。这是管理计算机文件的好习惯,可以有效避免找不到自己需要的文件。

(3) 从“开始”菜单运行 Microsoft Visual C++,执行“文件”→“新建”命令,打开“新建”对话框。如图 1-1 所示,在“工程”选项卡中选定工程类型为 Win32 Console Application,在“位置”文本框中指定希望的工作文件夹,在“工程”文本框中输入一个和实际问题相关的工程名,选择“创建新工作区”单选按钮,选中 Win32 平台,单击“确定”按钮。如图 1-2 所示,在打开的对话框中选择 An empty project 单选按钮,单击“完成”按钮。如图 1-3 所示,在打开的“新建工程信息”对话框中单击“确定”按钮,完成新工程的创建。

(4) 再执行“文件”→“新建”命令,打开“新建”对话框,从“文件”选项卡中选定文件类型为 C++ Source File,在“文件”文本框中输入一个和问题相关的文件名,如图 1-4 所示,单击“确定”按钮。

注意: 文件名不一定要和工程名相同,但仅有一个源文件的小工程,文件名通常和工程名相同。

(5) 如图 1-5 所示,在右侧编辑区输入程序代码。