

图书基本信息

书名：<<基于自然语言思想的递进C/C++教程 (上册C) >>

13位ISBN编号：9787302203919

10位ISBN编号：7302203911

出版时间：2009-9

出版时间：清华大学出版社

作者：李春庚

页数：195

字数：302000

版权说明：本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问：<http://www.tushu007.com>

前言

1. 本书编写背景 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，更增加了安全性高、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。

该教、学模式下，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和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。

因为面向对象编程中，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，多个句子就形成段落，多个段落就是文章。

基于人类自然语言的思想，我们将C/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。

这样就明晰了从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，不去区分C和C++，更多地学习C和C++的递进性、一致性，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。

本书讲稿已经在大连海事大学的教学中使用，得到良好的教学效果。

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) 篇幅短小，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) 仅用几个例题，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) 本书提出表达式、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。

内容概要

本书以自然语言“选词造句，连句成段，多段成章”的渐进学习思想贯穿全书，简捷深入地阐述了C语言的概念和语法。

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”。

名词、动词和连词的有机组合就形成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。

全书例题围绕几个问题展开，要么用不同方法解决，要么不断完善和深化；同时用一个综合实例，融合C语言的各个知识点，让C语言的学习更接近于实际开发。

每章的习题都是例题的变形和深化，以加强学生对知识的理解。

本书适合作为普通高校C语言课程的教材，也可用作培训教材及自学教材使用。

书籍目录

| | | | |
|--------|--|---------------------------------------|--|
| 第1章 概述 | 1.1 自然语言和计算机语言 | 1.2 C语言及C++语言 | 1.3 简单的C/C++程序及其运行方法 (环境的使用) |
| | 1.3.1 C/C++程序开发运行环境 | 1.3.2 格式化输出输入函数printf()和scanf() | 习题第2章 基本的C/C++语言 |
| | 2.1 C/C++语言中的名词——类型、量值 (常量和变量) | | |
| | 2.1.1 整型和整型量值 | 2.1.2 浮点型和浮点量值 (常量和变量) | 2.1.3 字符型和字符量子值 (常量和变量) |
| | 2.1.4 字符串常量 | 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 | |
| | 2.2.1 赋值运算符和赋值表达式 | 2.2.2 算术运算符和算术表达式 | 2.2.3 逻辑运算符和逻辑表达式 |
| | 2.2.4 关系运算符和关系表达式 | 2.2.5 其他运算符和表达式 | 2.2.6 不同类型量值的不同运算的混合 |
| | 2.3 C/C++语言中的连词及句子——分支、循环和顺序结构 | | |
| | 2.3.1 顺序结构 | 2.3.2 if...else——分支结构 | 2.3.3 switch...case——分支结构 |
| | 2.3.4 ?...:——分支结构 | 2.3.5 for——循环结构 | 2.3.6 while——循环结构 |
| | 2.3.7 do...while——循环结构 | 2.3.8 goto...label...if——循环结构 | 2.3.9 break——终止循环或开关 |
| | 2.3.10 continue——终止本次循环 | 2.4 C/C++语言中的句子小结 | 2.5 用C/C++语言编写程序——段落 |
| | 2.6 用C/C++语言编写复杂段落——语句嵌套 | 2.6.1 复杂段落——语句嵌套的含义 | 2.6.2 嵌套语句的示例 |
| | 2.6.3 复杂段落——语句嵌套的小结 | 2.7 总结 | 习题第3章 C/C++语言的横向名词性扩充 |
| | 3.1 C/C++语言的横向名词性扩充 (1) : 同类变量的集合——数组 | 3.1.1 一维数组 | 3.1.2 二维数组 |
| | 3.1.3 n维数组 | 3.1.4 字符数组 | 3.2 C/C++语言的横向名词性扩充 (2) : 自定义类型——typedef |
| | 3.2.1 简单类型的自定义 | 3.2.2 同类集合类型的自定义——变量数组类型的定义 | 3.3 C/C++语言的横向名词性扩充 (3) : 异类集合类型和异类集合变量——结构体类型 (值)、共用体类型 (值)、枚举型 (值) |
| | 3.3.1 结构体类型和结构体变量 | 3.3.2 共用体类型和共用体变量 | 3.3.3 枚举类型和枚举变量 |
| | 3.4 C/C++语言的横向名词性扩充 (4) : 异类集合类型的同类集合类型——结构体数组 | 3.5 各种横向扩充类型的融合使用 | 习题第4章 C/C++语言的纵向名词性扩充 (1) : 指针 |
| | 4.1 简单指针类型和指针量值 (常量和变量) | 4.1.1 变量的三层含义 | 4.1.2 变量的指针和指针类型的变量 |
| | 4.1.3 与指针相关的动词补充——&和*运算符 | 4.1.4 指针和指针变量示例 | 4.2 同类集合变量——数组的指针和对应的指针变量 |
| | 4.2.1 一维数组的指针和操作 | 4.2.2 二维数组的各级指针 | 4.2.3 操作二维数组的各种指针变量——本质是指向简单变量和一维数组的指针变量 |
| | 4.2.4 m维数组的指针和指向m维数组的指针变量 | 4.3 指向指针变量的指针量值——二级指针 | 习题第5章 C/C++语言的纵向名词性扩充 (2) : 函数 |
| | 第6章 名词的空间域、时间域和物理存储位置 | 第7章 编译预处理 | 第8章 C/C++语言的纵向名词性扩充 (3) : 宏观扩充——文件 |
| | 第9章 C/C++语言的纵向名词性扩充 (4) : 微观扩充——位和位域 | 附录 | 习题答案参考文献 |

章节摘录

自然语言有着悠久的历史，它的一切都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（如无需要，可统称为字词）。

人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，如果非要改变，那么就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；短语有一定自由度，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，人们是可以不断创造出新的字词，使语言不断丰富。

这是我们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。

就好像可以熟练地使用汉语，而又有多少人可以用汉语写出优美的文章呢！

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，需要对该领域知识的深刻理解和掌握。

这就是说，语言只是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，自己不写，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，它属于要解决专业的专业领域。

只有熟练掌握了计算机语言，又真正理解了所要解决的专业问题，才能编写出可以解决问题的、有一定规模的计算机程序。

编辑推荐

丛书特点：教学目标明确，注重理论与实践的结合；教学方法灵活，培养学生自主学习的能力；教学内容先进，强调计算机在各专业中的应用；教学模式完善，提供配套的教学资源解决方案。

版权说明

本站所提供下载的PDF图书仅提供预览和简介，请支持正版图书。

更多资源请访问:<http://www.tushu007.com>