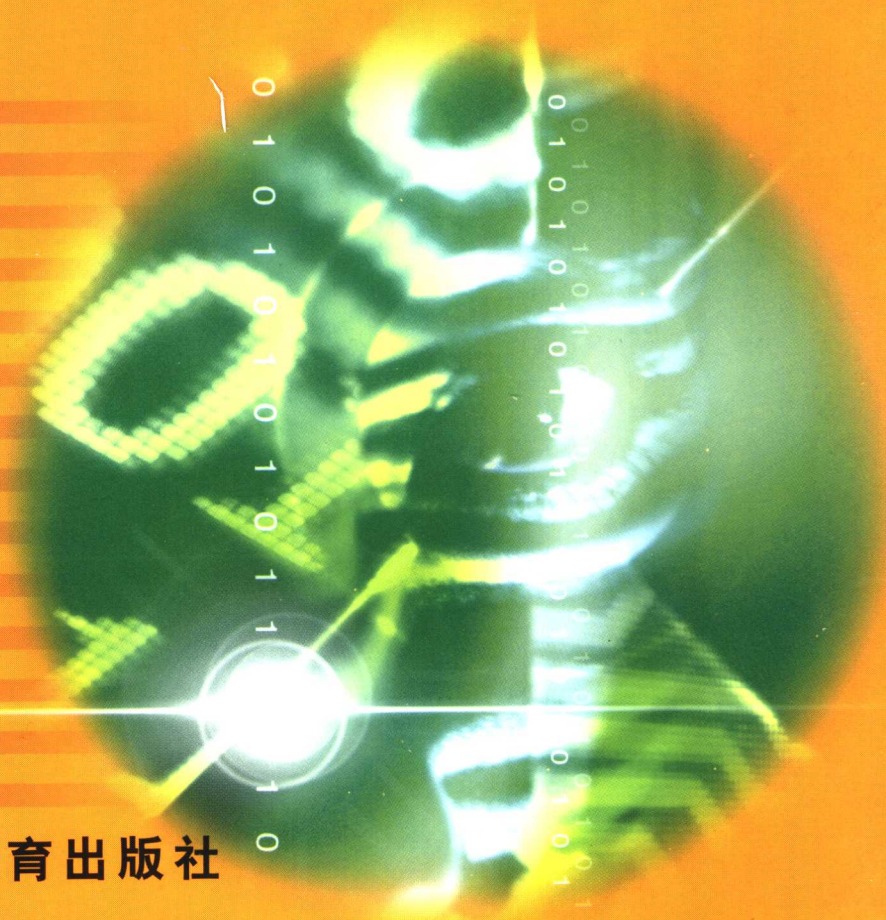
新世纪高职高专教改项目成果教材

高等职业教育技能型紧缺人才培养试用

软件技术基础

——离散数学、数据结构、C++编程实训

来可伟 编



高等教育出版社

新世纪高职高专教改项目成果教材

高等职业教育技能型紧缺人才培养试用

软件技术基础

——离散数学、数据结构、C++编程实训

来可伟　编

高 等 教 育 出 版 社

内容提要

本书是教育部新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目成果，是组织有关教育部高职高专教育专业教学改革试点院校编写的。

本教材系统地介绍了离散数学、数据结构、C++语言编程三个领域的知识。在理论方面，将离散数学作为编程技术必要的数学常识进行了深入浅出的介绍，以培养学员将实际问题抽象为数学表达式的抽象思维能力。在实验技能方面，采用基于案例（case-based）的方式，通过完整的编程全过程练习，使学员能按现代软件工业一线编程人员的要求掌握编程的基本技能，养成良好的规范化作业的习惯。

教材对各种常见数据结构采用了基于 C++语言类模板的实现方法，不但方法新颖、充分体现基础理论对编程的指导作用，而且有很高的实用价值。

本书适合于高等职业学校、高等专科学校、成人高校、示范性软件职业技术学院、本科院校举办的二级职业技术学院，也适合本科院校、继续教育学院、民办高校及技能型紧缺人才使用，还可供计算机专业人员和爱好者参考使用。

图书在版编目(CIP)数据

软件技术基础/来可伟编 .—北京:高等教育出版社，
2004.5
ISBN 7-04-014765-3

Ⅰ.软… Ⅱ.来… Ⅲ.软件-基本知识Ⅳ.TP31

中国版本图书馆 CIP 数据核字(2004)第 026944 号

策划编辑 冯 英 **责任编辑** 关 旭 **封面设计** 王凌波
版式设计 胡志萍 **责任校对** 杨雪莲 **责任印制** 杨 明

出版发行	高等教育出版社	**购书热线**	010-64054588
社 址	北京市西城区德外大街 4 号	**免费咨询**	800-810-0598
邮政编码	100011	**网 址**	http://www.hep.edu.cn
总 机	010-82028899		http://www.hep.com.cn
经 销	新华书店北京发行所		
印 刷	北京未来科学技术研究所有限责任公司印刷厂		
开 本	787×1092 1/16	**版 次**	2004 年 5 月第 1 版
印 张	16.25	**印 次**	2004 年 5 月第 1 次印刷
字 数	390 000	**定 价**	26.60 元(含光盘)

本书如有缺页、倒页、脱页等质量问题，请到所购图书销售部门联系调换。

出 版 说 明

为认真贯彻《中共中央国务院关于深化教育改革全面推进素质教育的决定》和《面向 21 世纪教育振兴行动计划》，研究高职高专教育跨世纪发展战略和改革措施，整体推进高职高专教学改革，教育部决定组织实施《新世纪高职高专教育人才培养模式和教学内容体系改革与建设项目计划》（教高[2000]3 号，以下简称《计划》）。《计划》的目标是："经过五年的努力，初步形成适应社会主义现代化建设需要的具有中国特色的高职高专教育人才培养模式和教学内容体系。"《计划》的研究项目涉及高职高专教育的地位、作用、性质、培养目标、培养模式、教学内容与课程体系、教学方法与手段、教学管理等诸多方面，重点是人才培养模式的改革和教学内容体系的改革，先导是教育思想的改革和教育观念的转变。与此同时，为了贯彻落实《教育部关于加强高职高专教育人才培养工作的意见》（教高[2000]2 号）的精神，教育部高等教育司决定从 2000 年起，在全国各省市的高等职业学校、高等专科学校、成人高等学校以及本科院校的职业技术学院（以下简称高职高专院校）中广泛开展专业教学改革试点工作，目标是：在全国高职高专院校中，遴选若干专业点，进行以提高人才培养质量为目的、人才培养模式改革与创新为主题的专业教学改革试点，经过几年的努力，力争在全国建成一批特色鲜明、在国内同类教育中具有带头作用的示范专业，推动高职高专教育的改革与发展。

教育部《计划》和专业试点等新世纪高职高专教改项目工作开展以来，各有关高职高专院校投入了大量的人力、物力和财力，在高职高专教育人才培养目标、人才培养模式以及专业设置、课程改革等方面做了大量的研究、探索和实践，取得了不少成果。为使这些教改项目成果能够得以固化并更好地推广，从而总体上提高高职高专教育人才培养的质量，我们组织了有关高职高专院校进行了多次研讨，并从中遴选出了一些较为成熟的成果，组织编写了一批"新世纪高职高专教改项目成果"教材。这些教材结合教改项目成果，反映了最新的教学改革方向，很值得广大高职高专院校借鉴。

新世纪高职高专教改项目成果教材适用于高等职业学校、高等专科学校、成人高校及本科院校举办的二级职业技术学院、继续教育学院和民办高校使用。

高等教育出版社

2002 年 11 月 30 日

前　言

长期以来，在初级和中级计算机教育，如高等职业教育的计算机专业以及非信息专业的计算机教育中，通常只教授编程语言，不教授相关的理论知识，这是一个弊病，因为程序编写仅仅是软件技术领域的一小部分，而软件技术是一项具有数学严密性的技术。科学技术的知识结构一般包括三个方面的内容：基础理论、技术基础理论和专业技术。对于软件技术，可以说离散数学是其基础理论，数据结构与算法设计是其技术基础理论。若对软件技术赖以发展的这些数学基础理论和专业理论的系统缺乏了解，所进行的编程实践就会成为盲目的实践，所编写的程序也许有漂亮的外观，但内部的数据和控制逻辑却是杂乱无章的。软件技术又是一项具有工程严密性的技术，如果没有形成规范化编程作业的观念、技能和作业方式，所编写的程序就会漏洞百出，缺乏文档，难于更新与维护。因此，近年来各方面一直在努力加强以上两方面的教学，本教材就是我们努力的结果。正如教材副标题“离散数学、数据结构、C++编程实训”所表明的，本书强调两点：一是从指导编程作业的目的出发，掌握离散数学和数据结构的一些常识；二是通过完整的编程作业训练，逐步掌握科学的、严密的程序设计方法，养成规范化作业的习惯。

本书主要面向信息专业高等职业教育，也可应用于非信息专业的其他工程专业本科计算机教育。本教材最初是以网络多媒体教材形式提供的，以便于学员自学一些较难理解的概念和编写程序的具体操作步骤。这次改编，又结合近年来的教学经验，在形式和内容上都作了多处重要修改，原有的多媒体素材经修改后作为书后配盘提供给读者。

虽然本书强调学习基本数学概念，但学习本书并不要求具备高深的数学预备知识，只要具备高中程度的数学基础即可。非信息专业的人员经过努力，都能掌握计算机程序开发的基本理论和方法。学习计算机程序编制并不需要特殊才能，只要按照科学的、严密的方法和步骤，养成规范化的编程作业习惯，成为一个好的编程人员并不困难。另外，本书还设计了一些选读内容，目的是为了扩大读者的知识面，这些章节以*号标识，学员可根据自身情况有选择地阅读这些带*号的章节。

本书由张志浩教授审阅，根据他的意见，书中做了多处重要修改，特此致谢。

本书的多媒体版本（书后配盘）得到了同济大学网络学院和同济大学高等技术学院的大力支持，特此致谢。

目　录

第 1 章　绪论 ····· 1
1.1　本课程的知识结构 ····· 1
1.2　离散数学 ····· 1
1.3　数据结构与算法设计 ····· 2
1.4　C++语言 ····· 2
1.5　学习要求和方法 ····· 3
1.6　书后配盘 ····· 4
1.7　软件技术教育 ····· 4
习题 ····· 6
第 2 章　命题逻辑 ····· 7
2.1　命题逻辑的基本概念 ····· 7
2.1.1　命题 ····· 7
2.1.2　复合命题 ····· 8
2.2　命题公式与真值表 ····· 12
2.2.1　命题公式 ····· 12
2.2.2　真值表 ····· 12
2.2.3　永真式、永假式及可满足公式 ····· 13
2.3　命题演算 ····· 13
2.3.1　命题公式的化简 ····· 13
*2.3.2　命题推理规则与方法 ····· 15
2.4　命题模型 ····· 17
2.4.1　命题建模 ····· 17
*2.4.2　命题模型推理 ····· 19
习题 ····· 20
第 3 章　谓词逻辑 ····· 22
3.1　谓词命题和谓词公式演算 ····· 22
3.1.1　谓词和个体词 ····· 22
3.1.2　量词 ····· 23
3.1.3　谓词公式演算 ····· 24
3.2　谓词模型 ····· 27
3.2.1　谓词建模 ····· 27
*3.2.2　谓词推理 ····· 28
习题 ····· 29
第 4 章　集合论 ····· 31
4.1　集合的概念 ····· 31
4.1.1　集合 ····· 31
4.1.2　集合的描述方法 ····· 32
4.1.3　集合间的关系 ····· 33
4.2　集合运算 ····· 33
4.2.1　集合的运算 ····· 33
4.2.2　集合的运算定律 ····· 34
4.3　集合模型 ····· 35
4.3.1　集合建模 ····· 36
*4.3.2　集合命题推理 ····· 37
***4.4　集合基数推理** ····· 38
习题 ····· 39
第 5 章　图论 ····· 42
5.1　图与树 ····· 42
5.1.1　图 ····· 42
5.1.2　图的性质 ····· 43
5.1.3　完全图和子图 ····· 44
5.1.4　图的同构 ····· 45
5.1.5　平面图 ····· 45
5.1.6　有权图和网络 ····· 46
5.1.7　树和根树 ····· 46
5.1.8　二叉树 ····· 48
5.2　图的运算 ····· 48
5.2.1　图的连通性 ····· 48
5.2.2　欧拉回路 ····· 49
5.2.3　哈密顿回路 ····· 51
5.2.4　生成树和最小费用生成树 ····· 52
5.2.5　狄克斯特算法 ····· 53
5.2.6　图的遍历 ····· 54
5.2.7　树的遍历 ····· 56
5.2.8　二叉树的遍历 ····· 56
5.3　图论建模 ····· 57
5.3.1　用图表示网络关系 ····· 57
5.3.2　用树表示分类的层次关系 ····· 59

5.3.3 搜索树……60
习题……62
第 6 章 C++编程作业入门……64
6.1 程序编写作业概述……64
6.1.1 编程语言……64
6.1.2 编译器与编译作业流程……65
6.2 用 VC++编译器进行编程作业……66
6.2.1 建立 VC++项目……66
6.2.2 编辑源代码文件……69
6.2.3 编译和查错……71
6.2.4 连编和运行程序……73
6.2.5 项目的关闭和再打开……74
6.2.6 向项目中添加文件和从项目中删除文件……74
6.3 C++语言词法概要……76
6.3.1 基本词汇……77
6.3.2 标点符号……77
6.3.3 关键词……78
6.3.4 标识符……78
6.3.5 常数……79
6.3.6 运算符……81
6.3.7 注释……83
6.4 C++语言句法概要……84
6.4.1 定义语句……84
6.4.2 数据类型的转换……86
6.4.3 导出数据类型……87
6.4.4 函数和函数调用机制……88
6.4.5 运算式……92
6.4.6 程序控制语句……93
6.4.7 应用数理逻辑设计程序控制语句……96
6.4.8 指针变量……99
6.4.9 字符串的运算……101
6.4.10 数据的输入输出函数……103
习题……109
第 7 章 用类编写面向对象的程序……111
7.1 C++语言中类的概念……111
7.1.1 概述……111
7.1.2 类定义……114
7.1.3 构造函数和析构函数……115
7.1.4 成员函数的定义与调用……117
7.1.5 引用数据类型和左值成员函数……119
7.1.6 成员函数的重载与运算符成员函数……121
7.1.7 案例——用类的运算符函数解线性方程组……122
7.2 面向对象程序设计方法与 C++的类……126
7.2.1 人的抽象思维方法……126
7.2.2 C++的类的聚集机制……127
7.2.3 C++的类的继承机制……128
7.2.4 按 C++的类划分程序模块……131
习题……134
第 8 章 用类模板实现线性数据结构……138
8.1 类模板……138
8.1.1 数据结构和离散数学……138
8.1.2 固定长度的 List 模板……139
8.1.3 模板的实例化……141
8.1.4 长度可自动改变的 List 模板……143
8.1.5 List 模板三……146
8.1.6 Linked List……149
8.2 矢量、矩阵和线性方程组的 C++模板……150
8.2.1 矢量和矩阵的数学概念……150
8.2.2 矢量和矩阵的模板……151
8.3 排序和检索……153
8.3.1 气泡法排序和函数模板……153
8.3.2 对分检索法……156
8.3.3 插入排序……157
8.4 队列和堆栈……158
8.4.1 Stack 模板……159
8.4.2 Queue 模板……160
习题……162
第 9 章 编程作业全过程……164
9.1 软件系统开发过程……164
9.1.1 系统分析、系统设计和系统实施……164
9.1.2 UML 方法……165
9.2 用类图建立数据模型……166
9.2.1 类和实例……167
9.2.2 属性……167
9.2.3 运算……167

9.2.4 类的图形表示……168
9.2.5 关联……168
9.2.6 关联类……169
9.2.7 关联的约束……169
9.2.8 继承……170
9.2.9 聚集……171
9.2.10 案例——学籍管理系统的数据模型……172
9.3 由数据模型设计 C++程序……172
9.3.1 类的映射规则……172
9.3.2 继承的映射规则……173
9.3.3 聚集的映射规则……173
9.3.4 关联的映射规则……175
9.3.5 关联类的映射规则……182
9.3.6 通过计算获取冗余信息……184
9.4 数据模型的一致性和完整性……185
9.4.1 数据模型的概念一致性……185
9.4.2 数据一致性和完整性的动态维护……187
9.5 用户界面的设计……189
9.5.1 用户界面的作用……189
9.5.2 UCD……190
9.5.3 设计用户菜单……192
9.5.4 验证用户输入……194
9.5.5 输出数据的可读性……195
9.6 程序的检测……195
9.6.1 程序错误的种类和原因……195
9.6.2 程序运行检测步骤……197
9.6.3 用 VC++编译器的调试功能跟踪程序运行过程……197
9.6.4 测试数据……201
9.7 编程作业的文档工作……201
习题……201
第 10 章 树和图的 C++模板……203
10.1 根树模板……203
10.1.1 根树的数据模型和 C++模板……203
10.1.2 根树的广度优先遍历函数……205
10.1.3 根树的深度优先遍历函数……206
10.1.4 求根树中所有路径……207
10.2 二叉树……208
10.2.1 二叉树的数据模型和 C++模板……208
10.2.2 二叉树的遍历算法……209
10.2.3 二叉检索树简介……210
10.3 递归……210
10.3.1 递归的数学概念……210
*10.3.2 递归算法的化解……212
10.4 图的 C++模板和程序……215
10.4.1 图的数据模型和 C++模板……215
10.4.2 无向图的最小费用生成树和克鲁斯克尔函数……216
习题……219
第 11 章 课程作业……221
11.1 课程作业一——学籍管理系统……221
11.1.1 根据系统数据模型设计 C++类定义……221
11.1.2 定义管理实例的序列……221
11.1.3 完成 UCD……222
11.1.4 设计菜单函数……222
11.1.5 设计交互式数据输入函数……224
11.1.6 划分程序模块……226
11.1.7 测试程序……227
11.1.8 编写完整的文档……227
11.1.9 其他要求……227
11.2 课程作业二——五子棋游戏……227
11.2.1 程序工作原理分析……227
11.2.2 数据建模……228
11.2.3 函数 Win()的实现……229
11.2.4 显示棋盘和棋子的函数……230
11.2.5 主函数控制逻辑……231
11.2.6 产生棋着的算法……232
11.2.7 其他要求……235
附录一 名词索引……236
附录二 离散数学部分习题参考答案和提示……243
附录三 如何阅读用形式文法描述的 C++语法规则……248
参考文献……251

第1章 绪 论

1.1 本课程的知识结构

工程技术学科的知识组成一般都包括3个方面的内容：基础理论、技术基础理论和专业技术。软件技术学科的3个知识层次分别为离散数学、数据结构与算法设计以及用某种编程语言进行编程作业。

计算机所能处理的信息称为数据。在计算机发展初期，计算机主要用于数值计算，处理的数据主要是数值数据，如整数、分数、有理数、无理数、实数、复数等。随着计算机科学和技术的发展，所处理的数据对象的概念随之拓宽，文字、图形、图像、语音、视频等人类感官所能感受的各种信息，都可成为数据对象，这就是非数值计算领域。

离散数学是现代数学的一个重要分支，与许多我们所学过的数学不同，计算机在非数值计算领域所处理的很多数据都具有离散性的特点。所以，对离散数学的学习和研究是计算机软件技术的数学理论基础。

用计算机程序解决实际问题时，必须解决两个方面的问题：

- 把要解决的问题表示为计算机所能处理的数据；
- 将解决问题的过程表示为计算机所能执行的步骤。

前者属于数据结构所研究的范畴，后者属于算法设计所研究的范畴。因此可以说，对数据结构和算法设计的学习和研究是计算机软件技术的技术基础理论。

程序是计算机解决实际问题的基本形式，因此，能用某种计算机语言熟练地编程是软件专业人员必须掌握的专业技术。

没有理论指导的实践是盲目的实践，如果对软件技术赖以发展的基础理论缺乏系统的了解，就不能成为一个合格的软件技术人员。所以，本教材强调从掌握基础理论的有关常识开始学习软件技术，但考虑到一般软件技术人员从事实际工作的需要，教材在阐述有关理论时将把重点放在一些与程序设计有密切关系的基本概念以及如何应用其指导程序设计上。同时，为兼顾知识的完整性，教材对相关概念也有必要的简介，但将其列为选读内容，使学员有充分的学习自主权。

下面简要介绍组成软件技术学科的3个领域中本教材所涉及的内容。

1.2 离散数学

离散数学是指导程序设计的主要基础理论。数学永远是工程学科生存和发展的基础，计算机程序设计更不例外。长久以来有一种误解，认为掌握了编程语言，就等于掌握了软件开发技术的全部。因此，许多有着漂亮界面的程序，其内部的数据组织却是杂乱无章的，其内部的控制逻辑也是混乱或者不严密的，其原因就是编写人员不注意应用，或不知如何应用指导程序设

计的数学理论。所以，本教材强调在学习编程前应首先了解基本的数学理论。

离散数学是现代数学的一个重要分支，它是研究如何描述和处理各种离散性数据对象的科学，涉及领域十分广泛。因此，并不仅仅只有软件技术人员才需要学习离散数学。在国外大学中，集合论、数理逻辑、图论已经是从事社会科学、教育学、商业、艺术等非理工类专业学生必须掌握的数学知识。本书用了一些源自日常生活的有趣例子来说明严密的数学思维过程，这种思维方式对于解决任何问题都是十分重要的。

本教材根据软件开发的一般需要，着重介绍命题逻辑、集合论、图论与树几个方面的内容，其中命题逻辑是关于如何将待解决的问题表示为程序中的逻辑运算的，集合论、图论与树则是关于如何将待处理的数据对象表示为计算机所能处理的形式的。

为了便于学员理解，教材对有关数学概念的阐述尽量通过实例进行深入浅出的说明，学员只要有高中数学基础，就可以掌握这些概念。结合实例阐述概念不但使学员容易理解概念，而且可以对如何应用这些概念去解决实际问题有直观的认识。

1.3 数据结构与算法设计

数据是信息的载体，在计算机科学中，数据一般指计算机程序能识别和处理的符号。数据一般又分为数值性数据和非数值性数据两大类，数据结构是数据（包括数值性和非数值性）之间内在关系的数学描述。本教材应用面向对象的概念来研究数据结构。从面向对象的观点来看，具有相同性质的数据抽象为数据对象，而那些具体的数据则是数据对象的成员或元素，数据结构描述的就是数据成员以及成员之间的关系。这种关系又分为逻辑关系和物理关系。数据结构的逻辑关系是面向问题的，是数据功能的描述，如存放学生数据的表格需要哪些栏目；数据结构的物理关系描述数据在计算机中的具体表达方式，如用整数还是用字符表示等。

虽然数据结构和算法设计研究领域的内容十分丰富，本教材只着重介绍其中应用最广的一种线性数据结构——序列（list），另外也将简要介绍根树和图。本教材对所介绍的数据结构的实现完全基于C++语言的类模板（class template）和运算符重载（operator overload）方法。用类模板编写数据结构的最大好处是使关于算法的数学描述和具体程序能完全对应起来，教材中提供的范例程序不仅使学员可以通过观察程序的实际运行过程来学习有关数学概念，还可以直接应用这些范例程序解决各种实际问题，这也是本教材的特点之一。

1.4 C++语言

计算机编程语言一般可分为常规编程语言和人工智能编程语言，常规编程语言又分为面向过程的和面向对象的。面向过程语言有FORTRAN、BASIC、C等，面向对象语言则包括C++、SmallTalk等。

面向对象编程的基本出发点是尽可能地按照人类的思维方式分析和解决问题。可以将其特点概括为3条：

- 封装性
- 聚集性

• 继承性

客观世界是复杂的，而计算机处理信息的能力是有限的。为了能在计算机能力所及的范围内尽可能如实地反映客观世界，需要对事物进行抽象。抽象是一种有目的的“取主去次”的过程，将那些对当前问题来说最重要的信息与其他无关紧要的信息分开，并依据这些主要特性对事物进行分类，这就是对象的封装性。对象的封装使人们可以把握事物的共性，以区分不同类的事物。对于复杂的事物，可用两种方法来分解其复杂性，一种是用对象的聚集性把它分解为比较简单的类；另一种是用对象的继承性把它分解为抽象程度较低的子类。

通过学习C++语言可以掌握面向对象程序设计的方法与技术。C++语言是由美国Bell实验室（AT&T实验室的前身）的B. Stroustrup在C语言的基础上开发的，是目前应用最广的面向对象编程语言之一。C++的名称总是给人以误导，总以为学习C语言是学习C++语言的基础，其实两者的设计理念完全不同。C 语言是面向过程的，C++是面向对象的。教学实践表明，许多学过面向过程编程语言（如 C、BASIC）的学员，在对面向对象方法的理解和掌握上往往比没有学过任何编程语言的学员更困难。因此建议在使用本教材前不必先学习其他任何编程语言。本教材采用的是基于案例的教学方法，是以没有任何编程基础的学员为对象的，入门不难，也有利于深造。

1.5 学习要求和方法

虽然本教材涉及传统上由不同课程讲述的内容，但并不是简单地把几本教材合并为一本，而是力求从内容编排上将数学概念与其实践应用有机结合起来，以充分体现理论对实践的指导意义。具体方法是把离散数学、数据结构学科领域中对程序设计最具指导意义的一些基本概念抽取出来，围绕理论和程序设计的关系进行阐述。同时，在程序设计的教学内容中加强了关于如何应用这些概念的论述。近年来的教学实践证明，增加关于离散数学和数据结构常识的教学内容以后，不仅没有增加学习难度，反而使随后的程序设计教学变得比较容易，学员所设计的程序质量也得到了很大提高。

教材副标题“离散数学、数据结构、C++编程实训”表明，本教材强调要同时进行两个方面的教学：一是要求学员掌握一点离散数学和数据结构的常识，能自觉地应用理论指导编程实践；二是要求学员尽可能按照软件工业一线编程人员的要求完成编程作业，从中掌握科学的、严密的程序设计方法，养成规范化作业的概念和技能。具体要求如下：

• 在基础理论方面，要求学员熟悉离散数学的基本概念，培养一定的抽象思维的能力，着重掌握应用这些基本概念解决实际问题的方法。在编排形式上，对于一些重要的或者难于理解的概念，专门设立了标题为“导读”的内容，进行专门讲解。教材特地提供了一个名词索引，将概念出现的章节一一列出来，以便于掌握概念之间的内在联系。

• 在实践技能方面，采用了基于案例的方法，目的是使学员完整地掌握开发软件的方法、步骤、技术，养成实施规范化工程作业的习惯。为此，教材中特别以“规范化编程实践”的标题列出了一些建议遵守的准则。此外，还对编程中经常易犯的错误以“经验与提醒”的形式作了特别提示。

对有关数学的习题，教材大部分都给出了参考答案和解题提示；对有关编程作业的习题则

提供了范例程序。软件技术领域的许多问题都存在不止一种解决方法，因此希望学员不要束缚自己的思路，要充分发挥自己的分析能力，这对于解决实际的能力的培养尤为重要。

通过团队协作分析和解决问题是将来从业所必须具备的能力，因此建议学员通过小组讨论和交流，对习题进行分析。无论是对培养个体应用知识的能力，还是对培养团队工作精神来说，这都是一种好方式。

本教材建议教学时数为60。建议学时分配为：用18学时学习离散数学理论部分；30学时用于学习C++语言和进行编程作业的实践训练（数据结构的内容已穿插在其中）；最后用12学时集中完成课程作业。大量的编程练习对学习软件技术来说十分关键，因此课外练习编程时间应不少于60学时。

1.6 书后配盘

在书后配盘中有本教材的网络多媒体版本。将光盘放入计算机光驱中，点击光驱设备符号或者用鼠标右键点击光驱设备符号，打开右键菜单，从中选择“自动播放”，教材首页便可自动在浏览器中打开。

将光盘中全部内容拷贝到Web服务器中，还可以成为一个网站。

与书面教材相比，光盘有以下优点：多媒体形式中有许多便于自学的素材，如在计算机上进行编程作业过程的视频演示以及对一些数据结构的工作原理的动画描述等；多媒体教材最大的优势还在于它提供了一种立体化的知识结构，通过点击网页中的链接，可将有关概念融会贯通。需要注意的是，该多媒体教材要求学时较多，凡光盘内容超出本教材之处均作为选学内容。

1.7 软件技术教育

为了帮助学员理解如何学习软件技术，下面对软件技术教育做些介绍。技术（technique）这个词在有些词典里的意思是指完成技术工作细节所必须具备的能力。例如，机械加工业的一线工作人员必须具备按统一的工艺流程加工零件的技能，才能生产出合格产品。程序编写员就是软件行业的一线工作人员，他们必须掌握的技术就是按规范化的作业方法，编写标准化的程序代码，提供规范化的文档和进行规范化的测试。

很久以来一直有一种误解，以为技巧在计算机软件开发中起着重要作用。诚然，在计算机发展的早期，巧妙的设计可以克服资源（内存和CPU）不足所带来的一些限制。但是，现在的程序设计已有完善的理论作为指导，软件开发也正在以越来越大的工业化规模进行。在这种形势下，对编写程序的个体来说，重要的不再是如何巧妙地利用计算机资源，而是用规范化的作业方式，编写出高度规范化的程序代码。

现在很多书中把软件技术和软件工程混为一谈，但它们是有区别的。第一，像所有其他工程一样，软件工程是组织和管理群体的方法，注重的是完成任务的群体活动，包括任务分解、活动组织和协调；而软件技术是软件开发一线编程人员所必须具备的、用规范化作业方式编写出高度规范化的程序代码的技能。第二，软件技术和软件工程是由不同的职能人员去实施的。对一般软件技术人员来说，他们的职责在大多数情况下只是按设计说明编写程序代码，而不是

从头策划一个项目。第三，软件技术和软件工程的区别还体现在两者质量保证措施不同。软件技术所关注的是具体程序代码的质量（即数据定义和算法的正确性），而软件工程关注的是软件开发全过程的质量。因此，软件技术教育的重点就是使从事软件开发的所有个体都充分认识软件开发作业的工程性质，摆脱手工作坊式的软件开发模式，掌握从事规范化编程作业的观念和技能。

下面摘录了一篇关于印度软件业的报导，从中可以进一步体会什么是个人应掌握的软件技术，什么是软件工程应该考虑的问题，从而更有目的地学习软件技术。

看看印度软件人(原载 2000 年 7 月 17 日文汇报，原著者：戴表)

（这是一位中国软件公司老板对印度软件人所下的评语。从中我们或许可以知道一些近年来印度软件业迅速在世界崛起的原因）

我在工作中，接触到印度软件人和他们开发出来的软件。从整个体系架构上看，他们的作品非常清晰，按照我们的要求实现了全部功能，而且相当稳定，但是打开具体代码一看，拖沓冗长。我们自己的一些程序员据此说他们水平低。但是，印度人能把软件整体把握得很好，并弄出相当好的设计文档。我们对某些特定的开发工具可能是精通的，却无法保证把一个软件稳当、完整地开发出来。

举个简单的例子：软件中需要一个列表，用来表示我们处理的事务。该表在业务繁忙的时候将变得很大。中国人总是用双向链表，抱着《数据结构》书在那里写链表的类，而我们雇的那些印度人根据情况，有时就抛弃链表，直截了当开一个大数组。为什么印度人不用链表？他们说，你们的设备最少都装备了 512MB 内存，占用一些没什么，而数组方式访问方便、效率高。他们做事情就这么简单明了，跟哥伦布竖鸡蛋一样。

印度人并不随意马虎

我对印度软件业的几点感受是：

1．流程重于项目。

2．质量监察独立于研发部门，专门检查开发部门的开发流程是不是按照既定流程走。如果质量管理人员觉得流程不对，他会直接上报高层，项目肯定就此停止。

3．所有的材料(包括草稿)都有文档。

4．准备工作做得很充分，详细文档要求达到拿到这个文档就可以编码的程度。一般写文档时间占 60%，而编码时间相对较少。

5．有各种详细的 Review（同行评审），包括项目组内、项目组之间、客户的……

6．计划详细，几十天的大项目竟能精确到小时级。

让中国高手惊呆了

印度人做事的总体风格很书面、很理论。我们招聘印度人，给应聘者出了一份与国内差不多的试卷，有基本概念和编程题目。等到他们完成后，我们这些自认的中国高手简直惊呆了！他们做的编程题目简直像是有统一答案，程序结构、注释、变量命名就不用说了，连表达方式都极其类似！反观所谓的中国高手，每个人都有自己的一套。到了新的岗位，先把前任的程序贬损一通，然后自己再开发有更多问题的代码来代替。我曾经统计了一下我的公司，一个软件中有 4 个以上的 CSocket 版本。每个人都觉得别人做得差，自己再搞一套。中国人就是这个样子，还振振有辞“我们这样有创造性”。

不依赖任何一个个人

印度软件公司的编程人员流动率(包括内部项目之间的流动)高达 30%，但他们可以让高中生编代码，可以想见其整体的文档水平之高。他们的产品不依赖任何一个个人，谁都可以立即辞职，而产品的开发还会正常进行。

习 题

1. 为什么说数学是工程技术的基础？
2. 软件技术和软件工程有什么区别？
3. 说说自己心目中的软件开发人员应该具备什么素质？
4. 印度软件人的故事给自己学习软件技术以什么启示？

第2章 命题逻辑

知识点：

命题逻辑是谓词逻辑的子集，它们都是指导程序设计的重要理论。从便于学习的角度出发，一般教材都是从命题逻辑开始介绍数理逻辑，就像在学习实数概念以前先学习自然数的概念一样。本章的学习目的为：

① 熟悉命题逻辑的基本概念并能熟练进行命题逻辑演算。

② 能对实际问题进行抽象，建立命题模型。既能将自然语言表示为命题公式，又能用自然语言描述命题表达式。

数理逻辑是用数学方法研究人们思维活动的一门学科，它将研究对象的形式关系抽象为可运算的符号体系，因此数理逻辑又称为符号逻辑，它是计算机应用和理论研究的基本工具，内容很丰富。本教材针对非计算机专业工程技术人员设计程序的需要，着重介绍命题逻辑的一些最基本的概念，以便能应用它们来设计程序逻辑控制语句（见 6.4.7 节）。

2.1 命题逻辑的基本概念

所有的数学学科都是建立在定义、定律、定理的体系基础上的。由于数学学科本身的特点，阐述定义、定律和定理的语句显得要比一般自然语言中的语句难于理解，因此，除了反复阅读有关文字叙述以外，更重要的是通过将概念应用于实际问题去逐渐加深理解。特别是命题逻辑，与日常生活有密切关系，所以一定要多联系实际来理解有关概念。

2.1.1 命题

语句是人们进行思维推理、表达内容的基本形式，在命题逻辑中称这种表达判断的陈述句为命题。命题是命题逻辑中的一个语法单位。

定义 2-1：

表示判断的陈述句称之为命题，其内容的真实性必须是惟一确定的，称为命题的值或真值。

定义 2-2：

原子命题是不能再分解的命题。真值确定的原子命题称为命题常量，真值可变化的原子命题称为命题变元。

命题不同于自然语言之处在于，它必须有惟一确定的真值。所谓真值是指命题语句所陈述的内容为真或为假。当一个命题为真时，用 T（或 TRUE）表示；若命题的值为假，则用 F（或 FALSE）表示。为了便于程序设计，一般也用二进制数 1 表示命题为真，0 表示命题为假。

命题一般用大写字母或带下标的大写字母来表示，如 P、P_1 等。命题变元没有确定的真假

值，在命题演算中可以代表任何命题。

例 2-1：

以下是命题的例子。

A：地球是球形的(T)

P_1：太阳绕着地球转(F)

Q_2：3+3=6 (T)

而 $x+3>0$ 不是命题，因其没有确定的真假值。

例 2-2：设 P 为命题变元，在没给它赋值之前，它没有值。

如果令

P：月球上没有生物

则 P 的值为真。

而如果令

P：地球是方的

则命题变元 P 的值为假。

导读：

① 若将命题变元比做普通数学里的变量，则命题常量就相当于普通数学中的常量。

② 所有的数学学科都是有共性的，通过对比以往所学过的数学概念，可以帮助理解离散数学的有关概念。

2.1.2 复合命题

学习数学最重要的就是学习如何进行运算。如普通数学用加、减、乘、除运算符把运算数组成数学运算式，用根号表示开方运算。在命题逻辑中，原子命题通过命题运算符构成的逻辑运算式，称为复合命题。主要的命题运算符有 6 个：否定、合取、析取、异或、蕴含、等价。

复合命题的值与组成它的原子命题的值有关，它们之间的取值关系，也就是运算符的运算规则可用真值表来表示，它可直观地描述命题运算的结果，是进行命题运算的重要工具。

定义 2-3（否定运算符¬）：

设 P 为一命题，则命题¬P 称为“P 的否命题”，读作“非 P”，符号¬称为否定运算符。规定当且仅当 P 是假时¬P 是真。

命题¬P 取值的规则如表 2-1 所示。

表 2-1 表示命题否定运算规则的真值表

P	¬P
0	1
1	0

例 2-3：

（1）给定命题 A：今天气温低于摄氏 30 度

则命题¬A 的意思是：今天气温不低于摄氏 30 度。

（2）给定命题 B：机器运行不正常

则命题$\neg B$的意思是：机器运行正常。

否定运算符的作用只是否定所依附的语句，而不管其原意如何。因此表示否定内容的语句不一定就是否定命题，如例 2-3 中命题 B 本身表达的是一个否定的内容，而其否定式表示的却是一个肯定的内容。

定义 2-4（合取运算符$\wedge$）：

设 P、Q 为两个命题，则命题 $P \wedge Q$ 称为“P、Q 的合取”，读作“P 与 Q”，符号$\wedge$称为合取运算符。合取运算的规定是：当且仅当 P 和 Q 同时是真时，$P \wedge Q$ 为真。

命题 $P \wedge Q$ 的取值规则如表 2-2 所示。

表 2-2 表示命题合取运算规则的真值表

P	Q	$P \wedge Q$
0	0	0
0	1	0
1	0	0
1	1	1

例 2-4：给定命题

A：地球是方的

B：今天晴天

则命题 $A \wedge B$ 的意思是：地球是方的且今天是晴天。

本例特意用上述不成立的事实组成命题，以强调复合命题关注的只是形式。

定义 2-5（析取运算符$\vee$）：

设 P、Q 为两个命题，则命题 $P \vee Q$ 称为“P、Q 的析取”，读作“P 或 Q”，符号$\vee$称为析取运算符。析取运算的规定是：当且仅当 P 和 Q 中至少一个为真时，$P \vee Q$ 为真。

命题 $P \vee Q$ 的取值规则如表 2-3 所示。

表 2-3 表示命题析取运算规则的真值表

P	Q	$P \vee Q$
0	0	0
0	1	1
1	0	1
1	1	1

例 2-5：给定命题

A：地球是方的

B：今天晴天

则命题 $A \vee B$ 的意思是：地球是方的或者今天是晴天。

与前例一样，本例子旨在说明复合命题关注的只是形式，所以也有意选取了上述似乎并不成立的事实组成命题。

定义 2-6（异或运算符∇）：

设 P、Q 为两个命题，则命题 $P \nabla Q$ 称为“P、Q 的异或”，读作“P 异或 Q”，符号∇称为异或运算符。异或运算的规定是：当且仅当 P 和 Q 中只有一个为真时，$P \nabla Q$ 为真。

命题 $P \nabla Q$ 的取值规则如表 2-4 所示。