

 语码转换式**双语**教学系列教材

总主编 蔡明德 副总主编 黎树斌 刘玉彬 总主审 范圣第

计算机科学与工程

COMPUTER SCIENCE AND ENGINEERING

主编 李锡祚



大连理工大学出版社

语码转换式双语法教学系列教材

总主编 蔡明德 副总主编 黎树斌 刘玉彬 总主审 范圣第

计算机科学与工程

COMPUTER SCIENCE AND ENGINEERING

主 编 李锡祚
副主编 宋海玉 王玲芬
主 审 宋 敏



大连理工大学出版社

图书在版编目(CIP)数据

计算机科学与工程/李锡祚主编. —大连:大连理工大学出版社, 2008. 7

(语码转换式双语教学系列教材)

ISBN 978-7-5611-4450-3

I. 计… II. 李… III. 计算机科学—双语教学—教材
IV. TP3

中国版本图书馆 CIP 数据核字(2008)第 138883 号

大连理工大学出版社出版

地址:大连市软件园路 80 号 邮政编码:116023

发行:0411-84708842 邮购:0411-84703636 传真:0411-84701466

E-mail: dulp@dulp.cn URL: <http://www.dulp.cn>

大连理工印刷有限公司印刷 大连理工大学出版社发行

幅面尺寸:183mm×233mm	印张:14	字数:433 千字
2008 年 7 月第 1 版	2008 年 7 月第 1 次印刷	

责任编辑:汪会武 千 川	责任校对:娜 婉
封面设计:波 朗	

ISBN 978-7-5611-4450-3

定 价:23.00 元

总序

PREFACE

2008年的盛夏,我们为广大师生奉上这套语码转换式双语教学系列教材。

语码转换式双语教学以不影响学科授课进度为前提,根据学生实际、专业特点、学年变化及社会需求等,随教学适时适量地渗透英语专业语汇、语句或语段,“润物细无声”般地扩大学生专业语汇,提高学生专业英语能力。这一模式适于所有学生,适于各学年,适于除英语外的各门学科课程,受到学生的普遍欢迎。

为保证语码转换式双语教学有计划、系统、高效而科学地持续性运作,减少教学的随意性和盲目性,方便师生的教与学,也使语码转换式双语教学的检查和考核工作有据可依,我们编写了这套语码转换式双语教学系列教材。

本套教材的全部内容一律采用汉英双语编写。

教材按专业组册,涵盖所有主干专业课和专业基础课,力求较为全面地反映各学科领域的知识体系。

依据当代语言学关于词汇场的理论,把同一知识体系中具有关联语义特征的内容编排在一起。课程内容编写以中文版教材章节为单位,以中文版教材章节为序,每门课以一本中文教材为蓝本,兼顾其他同类教材内容。

教材以渗透基本常用专业词汇尤其是短语为重点,注意了体现学科发展的新词、新语。同时考虑课程需求及专业特点,不同课程在不同程度上灵活地渗透了各章节的重要概念、定义,章节内容概述或体现章节内容主旨的语句及语段。各册教材还编写了体现各自专业特点的渗透内容,如:例题及解题方法,课程试卷,课程的发生、发展及前沿概述,公式,图示,实验原理,合同文本,案例分析,法条等。

部分课程补充了中文教材未能体现的先进理论、先进工艺、先进材料或先进方法的核心内容,弥补了某些中文教材内容的相对滞后性。部分教材还概括性地介绍了国内外学术发展的趋势、动态、研究方法和理论及编者的科研成果。

考虑学生汉译英的方便,各册都编写了词汇与短语索引。

语码转换式双语教学系列教材尚属尝试性首创,是多人辛勤耐心劳作的结果。尽管在编写过程中,我们一边使用一边修改,力求教材的实用性、知识性、先进性融为一体,希望它们能为学生专业语汇积累,为英文原版教材学习

扫除部分语言障碍,为专业资料阅读和专业内容英汉互译能力的提高起到重要作用;尽管编者在教材编写过程中都在实践语码转换式双语教学,但由于我们缺乏经验、学识水平和占有资料有限,加上为使学生尽早使用教材,编写时间仓促,教材中肯定在内容编写、译文处理、分类体系等方面存在缺点、疏漏和失误,恳请各方专家和广大师生对本套教材提出批评和建议,以期再版时更加完善。

在教材的编写过程中,大量中外出版物中的内容给了我们重要启示和权威的参考帮助,在此,我们谨向有关资料的编、著者表示诚挚的谢意。

编 者
2008 年 7 月

前言

FOREWORD

随着高等教育国际化进程的加快,“双语教学”作为一种新的教育教学方式,成为我国教育教学改革的热点,并被认为是培养国际化人才的一个重要途径。对于计算机专业,由于英语作为主要的专业技术描述语言,所以实施双语教学对学生日后进入全球化信息科学与技术领域进行国际交流,对提高学生的国际竞争力,培养国际化计算机专业人才具有重要意义。针对我国计算机专业双语教学的实际情况,为了使更多的学生都能在贯穿大学四年的各门课程教学和符合自己水平的不同层次的双语教学中受益,改变目前双语教育只能面向极少数学生的精英培养模式为普遍式培养模式,我们构建了一个以人为本、符合认知规律、省时高效的双语教学模式,即在课堂教学中,依托课程内容,采取语码转换式教学,不断渗透双语。为此,我们编写了这部计算机专业双语教学教材。

本书以中英文对译形式汇集了计算机类专业主要课程的常用词汇、短语、基本概念和定义等内容。这些课程包括:程序设计基础、面向对象方法、计算机组织与结构、离散数学、数据结构与算法、操作系统、数据库概论、编译技术、软件工程、计算机网络、JAVA技术、软件分析与技术、微机计算机接口技术、XML技术、嵌入式系统、人工智能、计算机图形学、人机交互技术、企业级软件开发、软件质量保证与测试、信息保障与安全、动漫与数码娱乐基础、游戏设计与编程等。书中对基本概念和定义给出了详细的中英文说明,并在每章节以*标出重点词汇,以示必须掌握。

为了方便读者,书中词汇、短语、概念和定义按照相应的中文教材章节中出现的次序进行编排,并在书后提供了中文索引,以便快速查阅。

本书共汇编了近5000个词条,包括词汇、短语、概念和定义。这些词条直接选自英文文献、最新原版教材及专业手册等,有些词汇或术语的翻译,不是机械地按英文直译,而是按专业术语或习惯给出译文,保证了本书具有较好的适用性。在附录中,还增加了VC++开发环境常见错误解析、Oracle数据库常见错误信息和Java错误信息,这使本书不仅适合于理论教学还适合于实践教学。此外,附录中的如何写英文摘要也给读者在论文发表、摘要写作方面给以经验及示范。因此,本书对于读者循序渐进地掌握计算机专业英语词汇以及专业英语的表达方式,阅读计算机专业文献,开展国际学术交流具有一定的裨益。

本书由李锡祚担任主编,并负责全书统稿。参加编写的人员有王晓强(第一章)、宋海玉(第二章)、王立明(第三章)、姜楠(第四、二十一章)、李锡祚(第五、九章)、王玲芬(第六章)、何丽君(第七章、附录二)、王朋杰(第八、二十章)、云健(第十章)、赵晶莹(第十一、十二、十九章、附录三)、李灵华(第十三章)、郭海(第十四章)、魏巍(第十五章)、刘爽(第

十六章)、米守房(第十七章)、包书哲(第十八章)、李威(第二十二、二十三章、附录一)。

在编写本教材的过程中,参考了大量国内外有价值的参考文献,这些资料在参考文献中已经列出。在此特向这些书的编者致以崇高的敬意。

由于编者水平有限,本书所收录的计算机科学与工程类词汇、短语、概念和定义还很不全面,而且错误和不足之处在所难免,深望读者批评指正。

编者

2008年5月

使用说明

1. 正文中*的含义:表示该词条在本篇章中为重要词条,要求学生必须掌握。
2. 查阅方法:本书可从两方面进行查阅。一种是按照课程的章节顺序进行查阅。另一种是按索引法,即按照词条的拼音顺序查找词条的出处,再查阅正文。

例 (1)程序流程图 9-3,9-3 即指该词条在第九章第三节中出现。
(2)操作数 1-2,3-7 即指该词条出现两次,在第一章第二节中出现,第三章第七节中也出现,并在两处均有中英文对照。

目录

CONTENTS

>> 第一章 程序设计基础

- 第一节 计算机语言概述 / 1
- 第二节 C 语言基础 / 2
- 第三节 程序设计基础 / 3
- 第四节 C 语言基本控制结构 / 3
- 第五节 抽象数据类型 / 3
- 第六节 函数 / 4
- 第七节 预处理命令 / 5
- 第八节 指针 / 5
- 第九节 文件系统 / 6

>> 第二章 面向对象方法

- 第一节 基于过程程序设计 / 7
- 第二节 类与对象 / 8
- 第三节 类的初始化、赋值与析构 / 9
- 第四节 继承与多态 / 10
- 第五节 运算符重载 / 13
- 第六节 模板与泛型 / 13
- 第七节 异常 / 13
- 第八节 I/O 流 / 14

>> 第三章 计算机组织与结构

- 第一节 概论 / 15
- 第二节 数字逻辑基础 / 15
- 第三节 计算机中的数据表示及运算方法 / 18
- 第四节 计算机中的运算部件 / 19
- 第五节 控制器 / 19
- 第六节 存储技术 / 20
- 第七节 汇编语言程序设计 / 22
- 第八节 输入输出系统 / 27

>> 第四章 离散数学

- 第一节 基础知识 / 29
- 第二节 逻辑 / 29
- 第三节 计数 / 31
- 第四节 关系和图 / 32
- 第五节 函数 / 33
- 第六节 序关系与结构 / 34
- 第七节 树 / 35
- 第八节 图形理论 / 36
- 第九节 半群与群 / 37

>> 第五章 数据结构与算法

- 第一节 绪论 / 40
- 第二节 线性表 / 40
- 第三节 栈和队列 / 41

- 第四节 串和数组 / 42
- 第五节 树和二叉树 / 43
- 第六节 图和广义表 / 44
- 第七节 查找 / 45
- 第八节 排序 / 46
- 第九节 文件 / 47

>> 第六章 操作系统

- 第一节 绪论 / 48
- 第二节 进程管理 / 48
- 第三节 CPU 调度 / 50
- 第四节 存储管理 / 51
- 第五节 设备管理 / 52
- 第六节 文件系统 / 53
- 第七节 案例研究 / 54

>> 第七章 数据库概论

- 第一节 绪论 / 56
- 第二节 关系数据库 / 57
- 第三节 关系数据库标准语言 SQL / 57
- 第四节 关系系统 / 58
- 第五节 关系数据理论 / 58
- 第六节 数据库设计 / 59
- 第七节 数据库恢复技术 / 59
- 第八节 并发控制 / 60
- 第九节 数据库安全性 / 61
- 第十节 数据库完整性 / 61
- 第十一节 面向对象数据库系统 / 62
- 第十二节 分布式数据库系统 / 62
- 第十三节 Oracle 数据库 / 62

>> 第八章 编译技术

- 第一节 概论 / 65
- 第二节 词法分析 / 65
- 第三节 上下文无关文法及语法分析 / 67
- 第四节 自顶向下的语法分析 / 69
- 第五节 自底向上的语法分析 / 71
- 第六节 语义分析 / 72
- 第七节 运行时环境 / 73
- 第八节 代码生成 / 75

>> 第九章 软件工程

- 第一节 软件工程概述 / 76
- 第二节 需求分析 / 77
- 第三节 软件设计 / 78
- 第四节 编码 / 79

- 第五节 测试 /80
- 第六节 面向对象方法 /81
- 第七节 软件发布 /82
- 第八节 软件维护 /82
- 第九节 项目管理 /83

>> 第十章 计算机网络

- 第一节 计算机网络概述 /85
- 第二节 数据通信与物理层 /86
- 第三节 数据链路层与局域网 /87
- 第四节 网络互联与 TCP/IP 协议 /89
- 第五节 因特网 /92
- 第六节 网络安全 /93

>> 第十一章 JAVA 技术

- 第一节 JAVA 语言与面向对象程序设计 /95
- 第二节 简单 JAVA 程序 /95
- 第三节 数据运算、控制流和数组 /96
- 第四节 类、包和接口 /96
- 第五节 JAVA 基础类及常用算法 /97
- 第六节 图形用户界面 /97
- 第七节 JAVA 高级编程 /98

>> 第十二章 软件分析与设计

- 第一节 绪论 /100
- 第二节 用例模型 /100
- 第三节 领域模型 /101
- 第四节 设计模型 /101
- 第五节 实现模型 /103

>> 第十三章 微型计算机接口技术

- 第一节 微处理器及其信号 /104
- 第二节 存储器 /104
- 第三节 微型计算机输入输出接口 /105
- 第四节 微型计算机的中断系统 /106
- 第五节 可编程接口芯片 /106
- 第六节 DMA 传输 /108
- 第七节 数模与模数转换 /109
- 第八节 现代微型计算机 /110
- 第九节 PC 系列微机外部设备接口 /111
- 第十节 微型计算机总线 /113
- 第十一节 Windows 输入输出程序设计 /113

>> 第十四章 XML 技术

- 第一节 XML 标准概览 /114

- 第二节 XML 文档规则 /115
- 第三节 定义文档内容 /116
- 第四节 XML 标准 /117

>> 第十五章 嵌入式系统

- 第一节 嵌入式系统介绍 /120
- 第二节 ARM 技术概述 /121
- 第三节 ARM 指令集 /121
- 第四节 Thumb 指令集 /122
- 第五节 基于 ARM 的嵌入式程序设计 /122
- 第六节 嵌入式计算平台 /123

>> 第十六章 人工智能

- 第一节 绪论 /124
- 第二节 知识的表示与推理 /124
- 第三节 专家系统 /126
- 第四节 知识发现 /126
- 第五节 机器学习 /127
- 第六节 分布式人工智能 /128
- 第七节 进化计算 /129

>> 第十七章 计算机图形学

- 第一节 绪论 /131
- 第二节 计算机图形系统及其设备 /131
- 第三节 生成直线和圆弧的算法 /132
- 第四节 变换 /132
- 第五节 层次结构 /133
- 第六节 交互技术 /133
- 第七节 光栅图形的扫描转换与区域填充 /134
- 第八节 隐藏面和隐藏线的消除 /134
- 第九节 曲线曲面的表示 /135
- 第十节 三维实体造型 /136
- 第十一节 真实感图形的基本理论与算法 /136
- 第十二节 计算机动画 /137
- 第十三节 科学计算可视化 /137

>> 第十八章 人机交互技术

- 第一节 概述 /139
- 第二节 人机交互模型 /139
- 第三节 人机系统交互界面的构架 /140
- 第四节 人机界面的设计 /141
- 第五节 标志需要和建立需求 /141
- 第六节 设计、制作原型和构建 /142
- 第七节 以用户为中心的交互设计 /142
- 第八节 评估入门 /143

目录

C CONTENTS

>> 第十九章 企业级软件开发

- 第一节 绪论 / 144
- 第二节 客户层与表示层 / 145
- 第三节 业务层 / 146
- 第四节 其他常见核心技术 / 147

>> 第二十章 软件质量保证与测试

- 第一节 软件质量保证 / 148
- 第二节 软件测试 / 149

>> 第二十一章 信息保障与安全

- 第一节 信息安全概述 / 154
- 第二节 密码技术 / 155
- 第三节 消息认证与数字签名 / 157
- 第四节 身份认证 / 158
- 第五节 访问控制 / 159
- 第六节 攻击 / 161
- 第七节 入侵检测 / 162

>> 第二十二章 动漫与数码娱乐基础

- 第一节 游戏概述 / 164
- 第二节 游戏设计 / 165
- 第三节 游戏美工 / 165

第四节 游戏管理 / 166

>> 第二十三章 游戏设计与编程

- 第一节 游戏引擎 / 167
- 第二节 图形程序设计 / 167
- 第三节 控制设备程序设计 / 167
- 第四节 音效程序设计 / 168

>> 附录一 VC++ 6.0 开发环境常见错误信息 / 169

>> 附录二 Oracle 数据库常见错误信息 / 171

>> 附录三 Java 错误信息 / 176

>> 附录四 英文摘要范例 / 177

>> 参考文献 / 179

>> 索引 / 183

第一章 程序设计基础

Chapter 1 Programming Fundamentals

第一节 计算机语言概述

Section 1 Introduction to Computer Language

★ 数据 data
★ 内存 memory
操作系统 Operating System (OS)
软件 software
硬件 hardware
键盘 keyboard
在线帮助 online help
★ 输入输出 input/output (I/O)
★ 输入 input
★ 输出 output
图形用户界面 Graphical User Interface (GUI)
计算机编程语言 computer programming language
文件扩展名 filename extension
根目录 root directory
文件夹/目录 folder/directory
通配符 wildcard character
只读 read-only
中央处理单元 Central Processing Unit (CPU)

美国信息交换标准码(ASCII 码) American Standard Code for Information Interchange (ASCII)
指令 instruction
随机存取存储器 Random Access Memory (RAM)
只读存储器 Read-Only Memory (ROM)
美国国家标准协会 American National Standards Institute (ANSI)
国际标准化组织 International Standards Organization (ISO)
★ 二进制码 binary code
低级语言 low-level language
机器语言 machine language
汇编语言 assembly language
★ 高级语言 high-level language
面向对象的编程语言 Object-Oriented Programming Language (OOPL)
寄存器 register
菜单 menu
向导 wizard

1 快捷键 keyboard shortcut: A combination of keys, such as Ctrl+C, that allows the user to activate a program function without clicking a series of menu options.

使用户无需点击多个菜单项就能调用系统功能的组合键,如 Ctrl+C。

2 ★ 比特 bit: A bit is the smallest unit of information handled by a computer. A bit can hold one value, either a 0 or an 1. Eight bits comprise a byte that can represent a letter or a number.

比特是计算机处理的最小信息单位,一个比特可以表示一个0或1。8个比特构成一个字节,可表示一个字母或一个数字。

3 ★ 字节 byte: An eight-bit unit of information that represents a single character.

由8个比特构成的信息单元,可以表示一个字符。

4 可执行文件 executable file: A file, usually with an .exe extension, containing instructions that tell a computer how to perform a specific task.

含有计算机指令并告诉计算机如何执行特定任务的文件,通常有.exe的扩展名。

5 千字节 Kilobyte (KB): 1KB approximately equals to one thousand bytes; exactly 1,024 bytes.

1KB大概等于1千个字节,准确数字为1024个字节。

6 兆字节 Megabyte (MB): 1MB approximately equals to one million bytes; exactly 1,048,576 bytes.

1MB大概等于一百万字节,准确数字为1 048 576个字节。

7 千兆字节 Gigabyte (GB): 1GB approximately equals to one billion bytes; exactly 1 073 741 824 bytes.

1GB大概等于10亿个字节,准确数字为1 073 741 824个字节。

- 8** 算术逻辑单元 Arithmetic-Logic Unit (ALU): ALU is one part of CPU, which performs arithmetic and logical instructions.
CPU 的一个组成部分,用于处理算术和逻辑指令。

第二节 C 语言基础

Section 2 The Foundation of C Language

* 编辑 edit
 * 编译 compile
 * 链接 link
 运行 run
 编译程序/编译器 compiler
 编辑器 editor
 连接器/连接程序 linker
 源程序/源代码 source code
 目标代码 object code
 * 注释 comment/remark
 * 换行符 newline character (\n)
 * 库函数 library function
 模块 module
 * 调试 debug
 * 语法 syntax
 运行错误 run-time error
 * 逻辑错误 logical error
 * 变量名 variable name
 * 变量值 variable value
 数据类型 data type
 * 操作数 operand
 常量 constant
 字面常量/直接常量 literal constant
 * 符号常量 symbolic constant
 * 小写 lowercase
 * 大写 uppercase

* 大小写敏感/区分大小写 case-sensitive
 下划线 underline
 分号 semicolon (;)
 变量声明 variable declaration
 变量定义 variable definition
 整型 integer
 浮点型 floating-point
 补码 complement
 字符 character
 字符串 string
 运算符 operator
 * 赋值运算符 assignment operator
 * 算术运算符 mathematical operator
 * 算术表达式 mathematical expression
 * 类型转换 type conversion
 自动类型转换 auto type conversion
 * 强制类型转换 cast
 * 自增运算符 increment operator
 * 自减运算符 decrement operator
 赋值表达式 assignment expression
 逗号运算符 comma operator
 逗号表达式 comma expression
 * 运算符优先级 operator precedence
 * 单目运算符 unary operator
 * 双目运算符 binary operator
 三目运算符 ternary operator

- 1** 可移植性 portability: Portability means that a program written for one computer system can be compiled and run on another system with little or no modification.

可移植性意味着,为一个计算机系统编写的程序,可以在只作很小的修改或不作修改的情况下就在另一个系统上编译和运行。

- 2** * 变量 variable: A variable is a named memory block. By using a variable's name, we can refer to the data stored in the memory block.

变量是一个被命名了的内存块。通过使用变量名,我们可以操作存储在这块内存块上的数据。

- 3** * 表达式 expression: In C language, an expression is anything that evaluates to a numeric value. Expression consists of simple expression and complex expression. Simple expression consists of some single items: a simple variable, literal constant or symbolic constant; Complex expression consists of simple expressions connected by operators.

在 C 语言中,表达式是可以求出数值的任何内容。表达式分为简单表达式和复合表达式两种。其中简单表达式由一些简单的项构成,包括简单变量、字面常量、符号常量等;而复合表达式则是由运算符连接简单表达式而形成的。

第三节 程序设计基础

Section 3 The Foundation of Programming

算法描述 description of algorithm

流程图 flowchart

自顶向下方法 top-down approach

模块化程序设计 modular programming

结构化程序设计 structured programming

❶ ★ 算法 algorithm: An algorithm is a method or procedure for carrying out a task.

算法是完成一项任务所采用的方法或步骤。

❷ ★ 数据结构 data structure: In computer science, a data structure is a way of storing data in a computer so that it can be used efficiently.

在计算机科学中,数据结构是指在计算机上存储数据的一种方式,它可以使数据的使用更加有效。

第四节 C 语言基本控制结构

Section 4 The Basic Control Structure of C Language

★ 复合语句 compound statement

空语句 null statement

转义符 escape character (\)

转义序列 escape sequence

格式控制/格式控制字符串 format control string

顺序结构 sequence structure

★ 选择结构 choose structure

★ 循环结构 loop structure

程序控制语句 program control statement

关系表达式 relational expression

if 语句 if statement

逻辑表达式 logical expression

条件运算符 conditional operator

while 语句 while statement

循环 loop

循环体 loop body

循环计数器 loop counter

★ 嵌套 nesting

❶ ★ 语句 statement: A statement is a complete direction instructing the computer to carry out some tasks. In C language, statements are usually written one per line, and always end with a semicolon. 语句是使计算机执行一些任务的一条完整指令。在 C 语言中,语句写在一行之内,并且以一个分号结束。

❷ ★ 关系运算符 relational operator: Relational operators always perform comparisons between their operands.

关系运算符通常在它们的操作数间进行比较。

❸ ★ 逻辑运算符 logical operator: Logical operators operate on true/false expressions. C language uses 0 and 1 to represent false and true respectively, and that any nonzero value is interpreted as being true.

逻辑运算符用于操作表示真或假的表达式中。在 C 语言中,使用 1 和 0 来分别代表真和假;另外,任何非零值也都被解释为真。

❹ ★ 嵌套循环 nested loop: The term, nested loop, refers to a loop that is contained within another loop.

嵌套循环是指一个循环包含在另一个循环之中。

第五节 抽象数据类型

Section 5 Abstract Data Type

数组元素 element of an array

数组初始化 array initialization

二维数组 two-dimensional array

多维数组 multidimensional array

★ 成员选择运算符 member selection operator

(->) or (.)

抽象数据类型 abstract data type (ADT)

结构体数组 array of structure

★ 枚举 enumeration

位运算符 bitwise operator

按位取反运算符 one's complement operator (~)

按位与运算符 bitwise AND operator (&)

按位同或运算符 bitwise inclusive OR operator

❶ ★ 数组 array: An array is a collection of data storage locations, each having the same data type and the same name. Each storage location in an array is called an array element and is distinguished from each other by a subscript.

数组是许多数据存储单位的一个集合, 这些数据存储单位有相同的名字和数据类型。数组中的每一个数据存储单位被称为数组元素, 它们相互之间通过下标进行区别。

❷ 下标 subscript: Subscript (or index) is a number following the array name, enclosed in brackets.

下标 (或称索引) 是一个紧跟在数组名后的、方括号内的数字。

★ 一维数组 single-dimensional array: A single-dimensional array has only a single subscript.

只有一个下标的数组称为一维数组。

❸ ★ 结构体 structure: A structure is a collection of one or more variables grouped under a single name for easy manipulation. The variables in a structure, unlike those in an array, can be of different data types. A structure can contain any data types supported by C language, including arrays and other structures. Each variable within a structure is called a member of the structure.

结构体可以看做是一个或多个变量的集合; 这些变量拥有同一个名字, 以便更容易操作。与数组中的变量不同, 结构体中的变量可以有不同的数据类型。结构体可以包含 C 语言中所有的数据类型, 包括数组和其他结构体。在结构体中的每一个变量都可以被称为结构体的一个成员。

❹ ★ 联合体 union: Unions are similar to structures, a union is declared and used in the same ways as a structure is. A union differs from a structure in that only one of its members can be used at a time. The reason for this is simple. All the members of a union occupy the same area of memory.

联合体与结构体很相似, 它们以同样的方式被声明和使用。联合体和结构体的区别之处在于, 在某一时刻, 联合体仅仅有一个成员可以被使用。原因很简单, 联合体中的所有成员占有内存中的同一块区域。

(|)

按位异或运算符 bitwise exclusive OR operator (^)

★ 二进制 binary

左移位运算符 left shift operator

右移位运算符 right shift operator

位段 bit field

第六节 函数

Section 6 Function

模块化 modularization

★ 函数定义 function definition

★ 函数原型 function prototype

函数声明 function declaration

无参函数 no argument function

函数头 function header

函数体 function body

参数列表 parameter list

空函数 null function

传参 pass argument

★ 返回值 return value

★ 被调用函数 called function

主调函数 calling function

★ 递归调用 recursive call

递归函数 recursive function

全局变量 global variable

存储类型 storage type

作用域 scope

生存期 lifetime

寄存器变量 register variable

自动变量 automatic variable

静态变量 static variable

外部变量 external variable

外部函数 external function

❶ ★ 函数 function: Functions are independent sections of code that perform specific tasks. When our program needs a task performed, it calls the function that performs that task.

函数是一些执行特定任务的独立代码块。当程序需要去完成一个指定的任务时, 程序就调用某个特

定的函数去完成这项任务。

■ **结构化程序设计** structured programming: Structured programming is a method of program design that emphasizes a modular, top-down approach.

结构化程序设计是一种程序设计方法,它强调模块化和自顶向下的设计思想。

■ **★ 实参** argument: An argument is an actual value passed to the function by the calling program. Each time a function is called, it can be passed different arguments.

实参是当函数被程序调用的时候被传递给函数的实际值。每一次函数被调用的时候,传递给它的实参值都可能是不同的。

■ **★ 形参** parameter: A parameter is an entry in a function header; it serves as a “placeholder” for an argument.

形参是函数头中的一个入口,它为实参保留一个“占位符”。

■ **局部变量** local variables: Variables declared in a function are called local variables. The term local means that the variables are private to that particular function and are distinct from other variables of the same name declared elsewhere in the program.

声明在一个函数内的变量被称为局部变量。局部意味着变量对于某个特定的函数来说是私有的,并且与程序中其他的同名变量相区别。

■ **★ 递归** recursion: The term recursion refers to a situation in which a function calls itself.

递归是指一个函数自己调用自己的情况。

第七节 预处理命令

Section 7 Preprocessor Directive

* 预处理命令 preprocessor directive

★ 宏 macro

★ 宏展开 expand a macro

★ 标识符 identifier

标准路径 standard directory

当前路径 current directory

条件编译 conditional compilation

重定义 redefinition

■ **★ 预处理器** preprocessor: C language implements some certain facilities by means of a preprocessor, which is conceptionally a separate first step in compilation. The two most frequently used features are “#include”, to include the contents of a file during compilation, and “#define”, to replace a token by an arbitrary sequence of characters.

C语言通过预处理器完成一些特定的功能;从概念上来说,预处理器是编译过程中单独执行的一个步骤。两个经常使用的预处理器是“文件包含”,即#include(用来在编译期间将指定的文件内容包含进当前文件)和“宏替换”,即#define(用一个任意字符序列替换一个标记)。

■ **★ 宏替换预处理命令** macro substitution preprocessor directive (#define): In C language, the #define preprocessor directive has two usages: creating symbolic constants and creating macros.

在C语言中,#define预处理命令有两个用途:创建符号常量和创建宏。

■ **★ 文件包含预处理命令** file inclusion preprocessor directive (#include): We use the #include preprocessor directive to include header files in our program. When it encounters the #include directive, the preprocessor reads the specified file and inserts it at the location of the directive.

我们使用#include预处理命令把所需的头文件加入程序中。当程序遇到#include预处理命令时,预处理器读入特定的文件,并将它插入到预处理命令指定的位置。

第八节 指针

Section 8 Pointer

地址运算符 address operator (&)

间接引用运算符 indirection operator (*)

函数指针 function pointer

★ 地址 address

★ 内存地址 memory address

★ 指向 point to

指针赋值 pointer assignment

指针比较 pointer comparison

直接存储 direct access

间接存储 indirect access

★ 链表 linked list

★ 结点 node

★ 插入 insert

★ 删除 delete

★ 堆 heap

★ 栈 stack

动态内存分配 dynamic memory allocation

静态内存分配 static memory allocation

1 ★ 指针 pointer: A pointer is the address of a memory block.

一个内存块的地址称为指针。

2 ★ 指针变量 pointer variable: A pointer variable is a variable that stores pointer.

指针变量是一个存储指针的变量。

第九节 文件系统

Section 9 File System

文本流 text stream

★ 磁盘 disk

二进制流 binary stream

★ 文件名 filename

★ 缓冲区 buffer

打开文件 open a file

关闭文件 close a file

1 ★ 文件 file: A file is a stream of bytes stored as a single unit.

文件是指存储在一个单位中的字节流。

2 文件指针 file pointer: In C language, file pointer points to a structure that contains information about the file. The information consists of the location of a buffer, the current character position in the buffer, whether the file is being read or written, and whether errors or end of file have occurred, etc.

在 C 语言中,文件指针指向一个包含文件信息的结构体。这些信息包括:缓冲区的位置、缓冲区中当前字符的位置、文件读写状态、是否出错或者文件是否已经达到尾部等等。