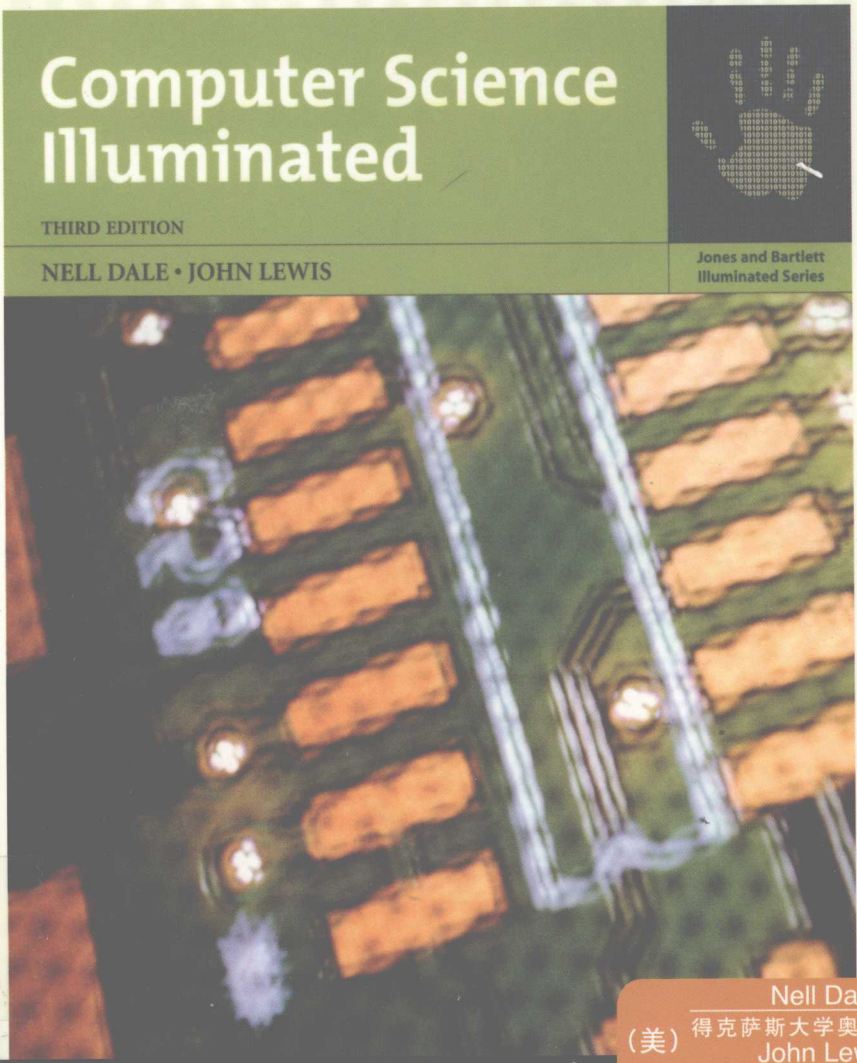
HZ BOOKS
华章教育

经 典 原 版 书 库

计算机科学概论

（英文版 · 第3版）



Nell Dale
（美）得克萨斯大学奥斯汀分校
John Lewis
维拉诺瓦大学
著

机械工业出版社
China Machine Press

经 典 原 版 书 库

计算机科学概论

（英文版 · 第3版）

Computer Science Illuminated

(Third Edition)

(美) Nell Dale 得克萨斯大学奥斯汀分校
John Lewis 维拉诺瓦大学 著

机械工业出版社
China Machine Press

Nell Dale and John Lewis: Computer Science Illuminated, Third Edition (ISBN-13: 978-0-7637-4149-5 ISBN-10: 0-7637-4149-3).

Copyright © 2007 by Jones and Bartlett Publishers, Inc.

All rights reserved.

Original English language edition published by Jones and Bartlett Publishers, Inc., 40 Tall Pine Drive, Sudbury, MA 01776.

This edition is licensed for distribution and sale in the People's Republic of China only, excluding Hong Kong, Taiwan and Macao and may not be distributed and sold elsewhere.

本书英文影印版由Jones and Bartlett Publishers, Inc. 授权出版。

此版本仅限在中华人民共和国境内（不包括中国香港、台湾、澳门地区）销售发行，未经授权的本书出口将被视为违反版权法的行为。

版权所有，侵权必究。

本书法律顾问　北京市展达律师事务所

本书版权登记号：图字：01-2008-0325

图书在版编目（CIP）数据

计算机科学概论（英文版·第3版）/（美）戴尔（Dale, N.），（美）刘易斯（Lewis, J.）著. —北京：机械工业出版社，2008.3

（经典原版书库）

书名原文：Computer Science Illuminated, Third Edition

ISBN 978-7-111-23516-3

Ⅰ.计… Ⅱ.①戴… ②刘… Ⅲ.计算机科学－英文 Ⅳ.TP3

中国版本图书馆CIP数据核字（2008）第020318号

机械工业出版社（北京市西城区百万庄大街22号 邮政编码 100037）

责任编辑：迟振春

北京京北制版厂印刷 · 新华书店北京发行所发行

2008年3月第1版第1次印刷

170mm×242mm · 41.25印张

标准书号：ISBN 978-7-111-23516-3

定价：66.00元

凡购本书，如有倒页、脱页、缺页，由本社发行部调换

本社购书热线：（010）68326294

Preface

Choice of Topics

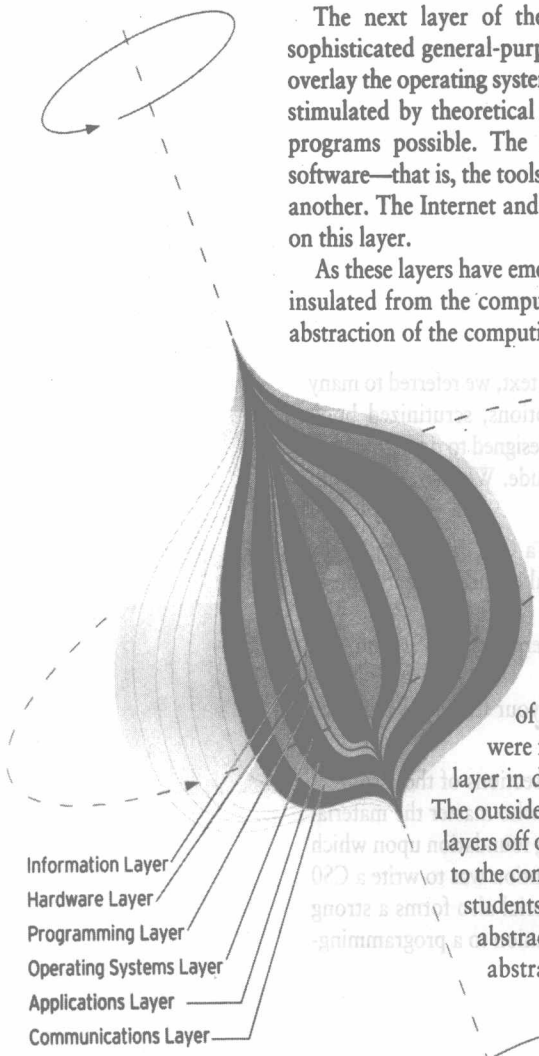
In putting together the outline of topics for this CS0 text, we referred to many sources. We looked at course catalogue descriptions, scrutinized book outlines, and administered an email questionnaire designed to find what you, our colleagues, thought such a course should include. We also asked both you and ourselves to make three lists:

- Four topics that students should master in a CS0 course if it is the only computer science course they will take during their college experience
- Four topics that you would like students entering your CS1 course to have mastered when they leave your class
- Four additional topics that you would like your CS1 students to be familiar with

The strong consensus that emerged from the intersections of these sources led to the working outline for this book. Students who master the material covered in this text before taking CS1 have a strong foundation upon which to continue in computer science. Although our intention was to write a CS0 text, our reviewers have pointed out that the material also forms a strong breadth-first background that can serve as a companion to a programming-language introduction to computer science.

Rationale for Organization

Chapter 1 of this book presents the history of hardware and software, showing how a computer system is like an onion. The computer, with its machine language, forms the heart of the onion, and layers of software and more sophisticated hardware have been added around this heart, layer by layer. First came machine language, part of the heart of this "onion." At the next layer, higher-level languages such as FORTRAN, Lisp, Pascal, C, C++, and Java were introduced parallel to the ever-increasing exploration of the programming process, using such tools as top-down design and object-oriented design. Over time, our understanding of the role of abstract data types and their implementations matured. The operating system, with its resource-management techniques, including files on ever-larger, faster secondary storage media, developed to surround and manage these programs.



The next layer of the computer system "onion" is composed of sophisticated general-purpose and special-purpose software systems that overlay the operating system. Development of these powerful programs was stimulated by theoretical work in computer science, which makes such programs possible. The final layer comprises networks and network software—that is, the tools needed for computers to communicate with one another. The Internet and the World Wide Web put the finishing touches on this layer.

As these layers have emerged over time, the user has become increasingly insulated from the computer system's hardware. Each layer provides an abstraction of the computing system beneath it. As each layer has evolved, users of the new layer have joined with users of the inner layers to create a very large workforce in the high-tech sector of our economy. This book is designed to provide an overview of the layers, introducing the underlying hardware and software technologies, thereby giving students an appreciation and understanding of all aspects of computing systems.

Having used history to describe the formation of the onion from the heart to the outer layers, we were faced with a design choice: We could look at each layer in depth from the inside out or from the outside in. The outside-in approach is very tempting. We could peel the layers off one at a time, moving from the most abstract layer to the concrete machine. However, research has shown that students understand concrete examples more easily than abstract ones, even when the students themselves are abstract thinkers. For this reason, we have chosen to begin with the concrete machine and examine the layers in the order in which they were created, trusting that a thorough understanding of one layer makes the transition to the next abstraction easier for students.

Changes in the Third Edition

The early editions of a new book are much like shakedown cruises for new ships: If the design is good, only minor problems occur that need tweaking. As a book—or a ship—gets older, however, new editions may call for a major

试读结束：需要全本请在线购买： www.ertongbook.com

overhaul. In planning for this revision, we asked our CS Education colleagues to give us feedback: What changes should we make? What kind of an overhaul is necessary? More than 50 of you shared your ideas with us. The consensus was that updating—not a major overhaul—was in order for the third edition. As a consequence, we have updated some of the biographies, replaced "old" tidbits with new ones, and reworked the "Ethical Issues" sections to keep them current. Content changes, including new sections on graphics, information security, computer security, cryptography, and e-commerce, are discussed later in this preface.

Several of you requested that we not put exercise answers in the back of the book. As with both of the previous editions, all of the exercise answers are available on the Web, password protected for use by the instructor.

Synopsis

Chapter 1 lays the groundwork for our approach to computer science—the computer system "onion"—by describing the rationale for this book's organization. Chapters 2 and 3 step back and examine a layer that is embodied in the physical hardware of the computer system. This *information layer* reflects how data is represented in the computer. Chapter 2 covers the binary number system and its relationship to other number systems such as decimal, the number system humans use on a daily basis. Chapter 3 investigates how we take the myriad types of data we manage—numbers, text, images, audio, and video—and represent them in a computer in binary format. We have added short discussions of the distinction between data and information and of the PNG image format.

Chapters 4 and 5 explore the *hardware layer*. Computer hardware includes devices such as transistors, gates, and circuits, all of which control the flow of electricity in fundamental ways. This core electronic circuitry gives rise to specialized hardware components such as the computer's central processing unit (CPU) and memory. Chapter 4 covers gates and electronic circuits. Chapter 5 covers the hardware components of a computer and describes how they interact within a von Neumann architecture. Of course, the ad at the beginning of Chapter 5 has been updated—and will probably be out of date by the time you read this preface!

Chapters 6 through 9 delve into various aspects of the *programming layer*. Chapter 6 examines the problem-solving process, both human and computer related. George Polya's human problem-solving strategies guide the discussion. Chapter 6 has been changed dramatically. The functionality of pseudocode is introduced as a way to write algorithms. Examples of both top-down design and object-oriented design are presented, along with detailed pseudocode descriptions. Chapter 7 covers the concepts of machine language and assembly language using Pep/7, a simulated computer. As part of this discussion, simple pseudocode algorithms are translated into both machine

code and assembly language. A looping example has been added as well. Chapter 8 focuses on the concepts underlying high-level programming languages. In this chapter, the pseudocode concepts are illustrated in brief examples from four programming languages: Ada, VB.NET, C++, and Java. Chapter 9 emphasizes the role of abstract data types and data structures in the programming process.

Chapters 10 and 11 cover the *operating system layer*. Chapter 10 discusses the resource-management responsibilities of the operating system and presents some of the basic algorithms used to implement these tasks. Chapter 11 covers file systems, including what they are and how they are managed by the operating system. A discussion of device drivers has been added to Chapter 10.

Chapters 12 through 14 deal with the *application layer*. This layer is made up of the general-purpose and specialized application programs that are available for the public to use to solve programs. In our coverage, we divide this layer into the subdisciplines of computer science upon which these programs are based. Chapter 12 examines information systems, Chapter 13 focuses on artificial intelligence, and Chapter 14 explores simulation, graphics, and other applications. We have added a major new section on information security to Chapter 12 that discusses confidentiality, integrity, and availability of data, as well as a subsection on cryptography. In addition, major new sections on computer graphics, e-commerce, and computer security appear in Chapter 14.

Chapters 15 and 16 cover the *communication layer*. Chapter 15 presents the theoretical and practical aspects of computers' communication with other computers. Chapter 16 discusses the World Wide Web and its influence on life today. A section on blogging has been added to Chapter 16.

Chapters 2 through 16 focus on what a computer can do and how it does it. Chapter 17 concludes by discussing the inherent limitations of computer hardware and software and by distinguishing the problems that can and cannot be solved using a computer. Big-O notation is presented as a way to talk about the efficiency of algorithms so that the categories of algorithms can be discussed, and the halting problem is presented to show that some problems are unsolvable.

The first and last chapters form a pair of "bookends": Chapter 1 describes what a computing system is and Chapter 17 explains what computing systems cannot do. The chapters in between look in depth at the layers that make up a computing system.

Why Not a Language?

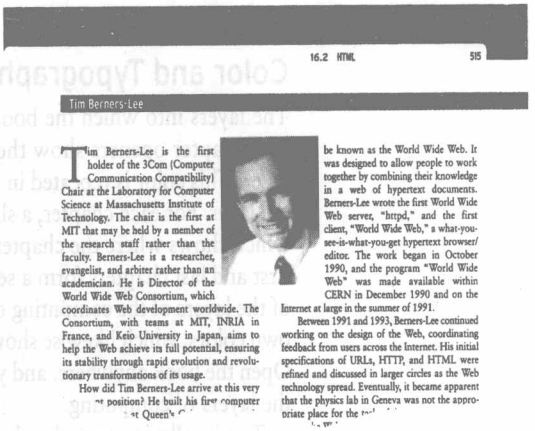
The original outline for this book included an "Introduction to Java" chapter. Some of our reviewers were ambivalent about including a

language at all; others wondered why Java was chosen and not C++. Ultimately, we decided to leave the choice of a specific programming language to the user. Introductory chapters, formatted in a manner consistent with the design of this book, are available for Java, C++, Visual Basic.NET, Python, Alice, and Pascal through Jones & Bartlett Publishers, Inc.

If the background of the students is such that they can master the introductory syntax and semantics of a language in addition to the background material in this book, please contact Jones & Bartlett Publishers or visit this textbook's website (http://csilluminated.jbpub.com/). As an alternative, one or all of these chapters can be used as enrichment for those students who have a stronger background.

Special Features

The third edition includes three special features intended to emphasize the history and breadth of computing as well as the moral obligations that come hand-in-hand with the advent of any new technology. First, each chapter includes a short biography of someone who has made a significant contribution to computing as we know it. The people honored in these sections range from those who contributed to the data layer, such as George Boole and Ada Lovelace, to those whose work enhanced the communication layer, such as Doug Engelbart and Tim Berners-Lee. These biographies are designed to give the students a taste of history and a sense of what kind of men and women contributed in the past and are contributing today to the world of computing.



16.2 HTML 515

Tim Berners-Lee

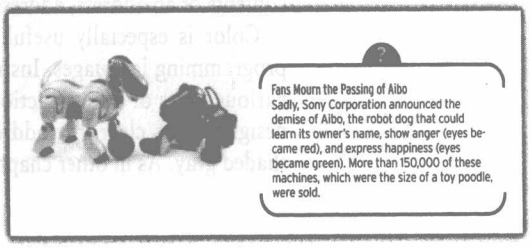
Tim Berners-Lee is the first holder of the 3Com (Computer Communication Compatibility) Chair at the Laboratory for Computer Science at Massachusetts Institute of Technology. The chair is the first at MIT that may be held by a member of the research staff rather than the faculty. Berners-Lee is a researcher, evangelist, and arbiter rather than an academician. He is Director of the World Wide Web Consortium, which coordinates Web development worldwide. The Consortium, with teams at MIT, INRIA in France, and Keio University in Japan, aims to help the Web achieve its full potential, ensuring its stability through rapid evolution and revolutionary transformations of its usage.

How did Tim Berners-Lee arrive at this very [illegible] position? He built his first computer [illegible] Queen's [illegible]

be known as the World Wide Web. It was designed to allow people to work together by combining their knowledge in a web of hypertext documents. Berners-Lee wrote the first World Wide Web server, "httpd," and the first client, "World Wide Web," a what-you-see-is-what-you-get hypertext browser/editor. The work began in October 1990, and the program "World Wide Web" was made available within CERN in December 1990 and on the Internet at large in the summer of 1991.

Between 1991 and 1993, Berners-Lee continued working on the design of the Web, coordinating feedback from users across the Internet. His initial specifications of URLs, HTTP, and HTML were refined and discussed in larger circles as the Web technology spread. Eventually, it became apparent that the physics lab in Geneva was not the appropriate place for the [illegible]

The second feature, which we call *callouts* for lack of a better word, are sidebar sections that highlight interesting tidbits of information from the past, the present, and the future. They are gleaned from history, from today's newspapers, and from the personal experiences of the authors. These little vignettes are designed



Fans Mourn the Passing of Aibo

Sadly, Sony Corporation announced the demise of Aibo, the robot dog that could learn its owner's name, show anger (eyes became red), and express happiness (eyes became green). More than 150,000 of these machines, which were the size of a toy poodle, were sold.

ETHICAL ISSUES Blogging

Like Websites, *blogs* have become ubiquitous virtually overnight. A blog is a Weblog or online journal. Most blogs are interactive and provide for feedback from readers. Whereas most bloggers write about mundane matters, the *blogosphere* has also emerged as a viable alternative news medium. Blogs are having a growing impact, sometimes supplementing or correcting reporting of the mainstream media. In 2004, blogs quickly exposed the inauthenticity of the documents used in a *60 Minutes* story about President George W. Bush's National Guard service. Many other blogs consistently provide a unique and unconventional perspective on the local and national news.

According to the *Wall Street Journal*, the audience for alternative media is expanding: "The number of Americans reading blogs jumped 58% in 2004 to an estimated 32 million people . . . with about 11 million looking to political blogs for news during the [2004] presidential campaign."[1]

But blogs are not just for online journalists or political commentators. Their use has also grown among doctors, lawyers, and teachers. Blogs have even become popular in the classroom. Many students have their own blogs where they record their impressions about teachers or other school-related information in a diary-like format. The use of student blogs has led to a new debate about the amount of control educators should exert over online classroom activities.

Of course, the blogosphere is not without its share of controversies. One such controversy erupted in

» continued

to amuse, to inspire, to intrigue, and, of course, to educate.

The third special feature is the "Ethical Issues" section that appears in each chapter. These sections are designed to illustrate that along with the advantages of computing come responsibilities for the consequences of its use. Privacy, hacking, viruses, and free speech are among the topics discussed in these sections. At the end of each chapter's exercises, a selection of "Thought Questions" cover these ethical issues as well as chapter content.

Color and Typography Are Signposts⊖

The layers into which the book is divided are color-coded within the text. The chapter openers show the onion, with the outside color showing the layer. This color is repeated in bars across the top of the pages pertaining to the layer. For each chapter, a slide appears on the side of the chapter opener, which shows where the chapter is within the layer. Earlier, we said that the first and last chapters form a set of "bookends." Although they are not part of the layers of the computing onion, we have given Chapters 1 and 17 their own colors, which likewise show up in the onion, the slide, and the color bar. Open the book anywhere, and you can immediately tell where you are within the layers of computing.

To visually separate the abstract from the concrete in the programming layer, we use different fonts for algorithms, including identifiers in running text, and program code. Using this signpost, you can tell at a glance whether the discussion is at the logical (algorithmic) level or at the programming language level. To clarify visually the distinction between an address and the contents of an address, addresses appear in red.

Color is especially useful in Chapter 7, which deals with low-level programming languages. Instructions are color-coded to differentiate the various parts of an instruction: The operation code is green, the register designation is clear, the addressing mode specifier is blue. Operands are shaded gray. As in other chapters, addresses appear in red.

⊖ 本书原版为彩色印刷，而影印版采用黑白印刷，有关颜色的问题请参考原书网站：http://csilluminated.jbpub.com/3e/。——编辑注

Website

A website has been established for this text that includes a wealth of additional information, both for the students and for the instructors:

http://csilluminated.jbpub.com/

Additional biographies, more information about some of the callouts, and updates that relate to ethical issues are available on this website. In addition, the site includes eLearning tools that provide a variety of exercises, such as crossword puzzles and digital labs for students.

For the instructor, answers to all exercises are available on the website. A wide selection of other exercises using new and innovative formats is available, along with PowerPoint presentations for each chapter.

Acknowledgments

You, our users, have been the most useful sources of information and advice during this revision. A heartfelt "thank you" to all 53 individuals who took the time to fill out our Web survey. We are also grateful to the reviewers of the first and second editions, as well as the following reviewers of this third edition:

Tim Bower, Kansas State University; Mikhail Brikman, Salem State College; Jacques Carette, McMaster University; Howard Francis, Pikeville College; Jim Jones, Graceland University; Murray Levy, West Los Angeles College; Lew Lowther, York University; Jeffrey McConnell, Canisius College; Richard Schlesinger, Kennesaw State University; Richard Spinello, Boston College; Herman Tavani, Rivier College; Amy Woszczynski, Kennesaw State University

Special thanks to Jeffrey McConnell of Canisius College, who wrote the graphics section in Chapter 14; Herman Tavani of Rivier College, who worked with us on the revision of the "Ethical Issues"; and Richard Spinello of Boston College, for his essay on the ethics of blogging. Thanks also to Bradley Miller and David Ranum of Luther College, who produced the Python chapter for us; Richard Schlesinger of Kennesaw State University, for contributing the independent VB.Net chapter; and Jose Garrido of Kennesaw State University, for creating the independent chapter introducing Alice programming.

I must also thank my tennis buddies for keeping me fit, my bridge buddies for keeping my mind alert, my family for keeping me grounded, and my dogs for their unconditional love.

—ND

I'd like to thank my family for their support.

—JL

Brief Contents

1 Laying the Groundwork 2
Chapter 1 The Big Picture 3

2 The Information Layer 32
Chapter 2 Binary Values and Number Systems 33
Chapter 3 Data Representation 53

3 The Hardware Layer 90
Chapter 4 Gates and Circuits 91
Chapter 5 Computing Components 119

4 The Programming Layer 146
Chapter 6 Problem Solving and Algorithm Design 147
Chapter 7 Low-Level Programming Languages 195
Chapter 8 High-Level Programming Languages 233
Chapter 9 Abstract Data Types and Algorithms 279

5 The Operating Systems Layer 322
Chapter 10 Operating Systems 323
Chapter 11 File Systems and Directories 353

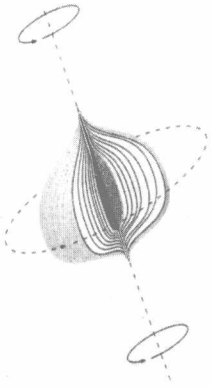
6 The Applications Layer 376
Chapter 12 Information Systems 377
Chapter 13 Artificial Intelligence 409
Chapter 14 Simulation, Graphics, and Other Applications 441

7 The Communications Layer 480
Chapter 15 Networks 481
Chapter 16 The World Wide Web 505

8 In Conclusion 530
Chapter 17 Limitations of Computing 531

试读结束：需要全本请在线购买： www.ertongbook.com

Contents



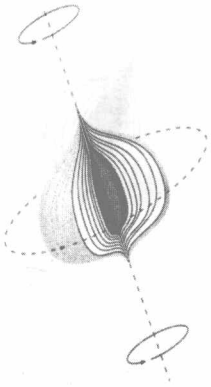
1 Laying the Groundwork 2

Chapter 1 The Big Picture 3

1.1 Computing Systems 4
Layers of a Computing System 4
Abstraction 6

1.2 The History of Computing 8
A Brief History of Computing Hardware 8
A Brief History of Computing Software 17
Predictions 23

1.3 Computing as a Tool and a Discipline 24
Summary 26
Ethical Issues: The Digital Divide 26
Key Terms 27
Exercises 27
Thought Questions 30



2 The Information Layer 32

Chapter 2 Binary Values and Number Systems 33

2.1 Numbers and Computing 34

2.2 Positional Notation 34
Binary, Octal, and Hexadecimal 36
Arithmetic in Other Bases 39
Power of 2 Number Systems 40
Converting from Base 10 to Other Bases 42
Binary Values and Computers 43
Summary 45
Ethical Issues: Computers and Homeland Security 46
Key Terms 46
Exercises 47
Thought Questions 51

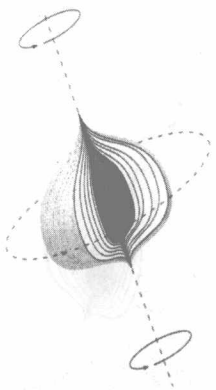
Chapter 3 Data Representation 53

3.1 Data and Computers 54
Analog and Digital Data 55
Binary Representations 57

3.2 Representing Numeric Data 59
Representing Negative Values 59
Representing Real Numbers 63
3.3 Representing Text 66
The ASCII Character Set 67
The Unicode Character Set 68
Text Compression 69
3.4 Representing Audio Data 73
Audio Formats 75
The MP3 Audio Format 76
3.5 Representing Images and Graphics 76
Representing Color 76
Digitized Images and Graphics 78
Vector Representation of Graphics 79
3.6 Representing Video 80
Video Codecs 80
Summary 82
Ethical Issues: MGM Studios, Inc. v. Grokster, LTD. 82
Key Terms 83
Exercises 84
Thought Questions 89

3 The Hardware Layer 90

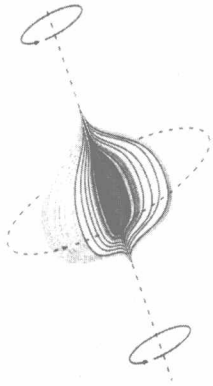
Chapter 4 Gates and Circuits 91
4.1 Computers and Electricity 92
4.2 Gates 94
NOT Gate 94
AND Gate 95
OR Gate 96
XOR Gate 96
NAND and NOR Gates 97
Review of Gate Processing 98
Gates with More Inputs 98
4.3 Constructing Gates 99
Transistors 99
4.4 Circuits 101
Combinational Circuits 102
Adders 105
Multiplexers 107
4.5 Circuits as Memory 108
4.6 Integrated Circuits 109
4.7 CPU Chips 110



Summary 110
Ethical Issues: Email Privacy 111
Key Terms 111
Exercises 112
Thought Questions 117

Chapter 5 Computing Components 119

5.1 Individual Computer Components 120
5.2 Stored-Program Concept 123
von Neumann Architecture 123
The Fetch-Execute Cycle 128
RAM and ROM 130
Secondary Storage Devices 131
Touch Screens 135
5.3 Non-von Neumann Architectures 137
Summary 139
Ethical Issues: Bioinformatics Research and the Case of deCODE Genetics, Inc. 140
Key Terms 142
Exercises 142
Thought Questions 145



4 The Programming Layer 146

Chapter 6 Problem Solving and Algorithm Design 147

6.1 Problem Solving 148
How to Solve Problems 149
Applying Polya's Problem-Solving Strategies 152
6.2 Algorithms 153
The Computer Problem-Solving Process 153
Following an Algorithm 155
Developing an Algorithm 157
6.3 Pseudocode 157
Following a Pseudocode Algorithm 158
Pseudocode Functionality 159
Pseudocode Example 162
6.4 Top-Down Design Methodology 166
A General Example 167
A Computer Example 169
Summary of Methodology 173
Testing the Algorithm 174
6.5 Object-Oriented Methodology 175
Object Orientation 175

Design Methodology 175
General Example 179
Computer Example 180
6.6 Important Threads 183
Information Hiding 184
Abstraction 184
Naming Things 185
Programming Languages 186
Testing 186
Summary 187
Ethical Issues: Licensing Computer Professionals 188
Key Terms 189
Exercises 189
Thought Questions 193

Chapter 7 Low-Level Programming Languages 195

7.1 Computer Operations 196
7.2 Levels of Abstraction 196
7.3 Machine Language 197
Pep/7: A Virtual Computer 198
7.4 A Program Example 204
Problem and Algorithm 205
A Program 206
7.5 Assembly Language 211
Pep/7 Assembly Language 211
Pseudo-operations 212
Assembly-Language Version of Program Hello 213
A New Program 214
Decision Making in Assembly Language 217
A Program with a Loop 219
7.6 Other Important Threads 222
Abstraction 222
Testing 222
Test-Plan Implementation 223
Summary 225
Ethical Issues: Software Piracy and Copyrighting 226
Key Terms 226
Exercises 226
Thought Questions 230

Chapter 8 High-Level Programming Languages 233

8.1 Translation Process 234
Compilers 234
Interpreters 235

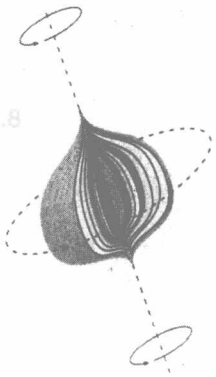
8.2 Programming Language Paradigms 236
8.3 Functionality of Imperative Languages 239
Boolean Expressions 239
Strong Typing 241
Input/Output Structures 245
Control Structures 247
Composite Data Types 264
8.4 Functionality of Object-Oriented Languages 267
Encapsulation 267
Inheritance 268
Polymorphism 269
Summary 270
Ethical Issues: Open-Source Software Development 271
Key Terms 272
Exercises 273
Thought Questions 277

Chapter 9 Abstract Data Types and Algorithms 279

9.1 Abstract Data Types 280
9.2 Implementation 281
Array-Based Implementations 281
Linked Implementation 283
9.3 Lists 285
Basic List Operations 286
Additional List Operations 290
9.4 Sorting 291
Selection Sort 291
Bubble Sort 293
Quicksort 295
9.5 Binary Search 300
9.6 Stacks and Queues 302
Stacks 302
Queues 304
Implementation 304
9.7 Trees 304
Binary Trees 305
Binary Search Trees 308
Other Operations 313
Graphs 314
9.8 Programming Libraries 314
Summary 315
Ethical Issues: Computer Hoaxes and Scams 316

Key Terms 317
Exercises 317
Thought Questions 321

5 The Operating Systems Layer 322



Chapter 10 Operating Systems 323

10.1 Roles of an Operating System 324
Memory, Process, and CPU Management 326
Batch Processing 327
Timesharing 328
Other OS Factors 329

10.2 Memory Management 330
Single Contiguous Memory Management 331
Partition Memory Management 333
Paged Memory Management 335

10.3 Process Management 337
The Process States 337
The Process Control Block 339

10.4 CPU Scheduling 339
First-Come, First-Served 340
Shortest Job Next 341
Round Robin 341
Summary 343
Ethical Issues: Digital Rights Management and the Sony Rootkit Controversy 344
Key Terms 345
Exercises 345
Thought Questions 351

Chapter 11 File Systems and Directories 353

11.1 File Systems 354
Text and Binary Files 354
File Types 355
File Operations 357
File Access 358
File Protection 359

11.2 Directories 360
Directory Trees 361
Path Names 362

11.3 Disk Scheduling 365
First-Come, First-Served Disk Scheduling 367
Shortest-Seek-Time-First Disk Scheduling 368

试读结束：需要全本请在线购买： www.ertongbook.com