

EIGHTH EDITION

Computer Science Illuminated

Nell Dale, PhD

University of Texas, Austin, Texas

John Lewis, PhD

Virginia Tech, Blacksburg, Virginia



JONES & BARTLETT
LEARNING



World Headquarters
Jones & Bartlett Learning
25 Mall Road
Burlington, MA 01803
978-443-5000
info@jblearning.com
www.jblearning.com

Jones & Bartlett Learning books and products are available through most bookstores and online booksellers. To contact Jones & Bartlett Learning directly, call 800-832-0034, fax 978-443-8000, or visit our website, www.jblearning.com.

Substantial discounts on bulk quantities of Jones & Bartlett Learning publications are available to corporations, professional associations, and other qualified organizations. For details and specific discount information, contact the special sales department at Jones & Bartlett Learning via the above contact information or send an email to specialsales@jblearning.com.

Copyright © 2025 by Jones & Bartlett Learning, LLC, an Ascend Learning Company

All rights reserved. No part of the material protected by this copyright may be reproduced or utilized in any form, electronic or mechanical, including photocopying, recording, or by any information storage and retrieval system, without written permission from the copyright owner.

The content, statements, views, and opinions herein are the sole expression of the respective authors and not that of Jones & Bartlett Learning, LLC. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not constitute or imply its endorsement or recommendation by Jones & Bartlett Learning, LLC and such reference shall not be used for advertising or product endorsement purposes. All trademarks displayed are the trademarks of the parties noted herein. *Computer Science Illuminated, Eighth Edition* is an independent publication and has not been authorized, sponsored, or otherwise approved by the owners of the trademarks or service marks referenced in this product.

There may be images in this book that feature models; these models do not necessarily endorse, represent, or participate in the activities represented in the images. Any screenshots in this product are for educational and instructive purposes only. Any individuals and scenarios featured in the case studies throughout this product may be real or fictitious but are used for instructional purposes only.

27514-8

Production Credits

Vice President, Product Management: Marisa R. Urbano
Vice President, Content Strategy and Implementation:
Christine Emerton
Sr. Product Manager, Cybersecurity: Ray Chew
Manager, Content Strategy: Orsolya Gall
Content Strategist: Sara Holloway
Content Coordinator: Samantha Gillespie
Director, Project Management and Content Services: Karen Scott
Manager, Program Management: Kristen Rogers
Project Manager: Belinda Thresher
Project Manager, Navigate: Erin Bosco
Senior Digital Project Specialist: Angela Dooley
Marketing Manager: Mark Adamiak

Content Services Manager: Colleen Lamy
Manufacturing Buyer: Bob Valentine
Composition: Straive
Project Management: Straive
Cover Design: Kristin Parker
Text Design: Kristin Parker
Media Development Editor: Faith Brosnan
Rights & Permissions Manager: John Rusk
Rights Specialist: Liz Kincaid
Cover Image (Title Page, Part Opener, Chapter Opener):
© Yuichiro Chino
Printing and Binding: Lakeside Crawfordsville

Library of Congress Cataloging-in-Publication Data

Names: Dale, Nell (Nell B.) author. | Lewis, John, 1963- author.

Title: Computer science illuminated / Nell B. Dale, PhD, The University of Texas at Austin, John Lewis, PhD, Virginia Tech.

Description: Eighth edition. | Burlington, Massachusetts : Jones & Bartlett Learning, [2025] | Includes bibliographical references and index. |

Summary: "Computer Science Illuminated is designed for the introductory, breadth-first course, providing students with an overall introduction to the field of computing. It is also appropriate for AP Computer Science Principles course. The authors provide a unique and innovative layered approach, moving through the levels from an organized, language-neutral perspective"— Provided by publisher.

Identifiers: LCCN 2023035411 | ISBN 9781284275070 (paperback)

Subjects: LCSH: Computer science.

Classification: LCC QA76 .D285 2025 | DDC 004—dc23/eng/20230819

LC record available at <https://lccn.loc.gov/2023035411>

6048

Printed in the United States of America

27 26 25 24 23 10 9 8 7 6 5 4 3 2 1

© Artur Debat/Getty Images; © Alan Dyer/Stocktrek Images/Getty Images

*To all the students who will use
this book: It is written for you.*

—**Nell Dale**

*To my wife, Sharon, and
our children, Justin, Kayla,
Nathan, and Samantha.*

—**John Lewis**



Nell Dale, the University of Texas at Austin

Well-respected in the field of computer science education, Nell Dale has served on the faculty of the University of Texas at Austin for more than 25 years and has authored over 40 undergraduate computer science textbooks. After receiving her BS in Mathematics and Psychology from the University of Houston, Nell entered the University of Texas at Austin, where she earned her MA in Mathematics and her PhD in Computer Science. Nell has made significant contributions to her discipline through her writing, research, and service. Nell's contributions were recognized in 1996 with the ACM SIGCSE Award for Outstanding Contributions in Computer Science Education and in 2001 with the ACM Karl V. Karlstrom Outstanding Educator Award. She was elected an ACM Fellow in 2010. In 2013, she received the IEEE Taylor L. Booth Education Award. Nell has retired from full-time teaching, giving her more time to write, travel, and play tennis and bridge. She lives in Austin, Texas.

John Lewis, Virginia Tech

John Lewis is a leading educator and author in the field of computer science. He has written a market-leading textbook on Java software and program design. After earning his PhD in Computer Science, John spent 14 years at Villanova University in Pennsylvania. He now teaches computing at Virginia Tech, his alma mater, and works on textbook projects out of his home. He has received numerous teaching awards, including the University Award for Teaching Excellence and the Goff Award for Outstanding Teaching. His professional interests include object-oriented technologies, multimedia, and software engineering. In addition to teaching and writing, John actively participates in the ACM Special Interest Group on Computer Science Education (SIGCSE) and finds time to spend with his family and in his workshop.

BRIEF CONTENTS

© Artur Debat/Getty Images, © Alan Dyer/Stocktrek Images/Getty Images

1 Laying the Groundwork 2

Chapter 1 The Big Picture 3

2 The Information Layer 34

Chapter 2 Binary Values and Number Systems 35

Chapter 3 Data Representation 55

3 The Hardware Layer 92

Chapter 4 Gates and Circuits 93

Chapter 5 Computing Components 121

4 The Programming Layer 148

Chapter 6 Low-Level Programming Languages and Pseudocode 149

Chapter 7 Problem Solving and Algorithms 189

Chapter 8 Abstract Data Types and Subprograms 239

Chapter 9 Object-Oriented Design and High-Level Programming Languages 279

5 The Operating Systems Layer 326

Chapter 10 Operating Systems 327

Chapter 11 File Systems and Directories 359

6 The Applications Layer 384

Chapter 12 Information Systems 385

Chapter 13 Artificial Intelligence 419

Chapter 14 Simulation, Graphics, Gaming, and Other Applications 453

7 The Communications Layer 494

Chapter 15 Networks 495

Chapter 16 The World Wide Web 525

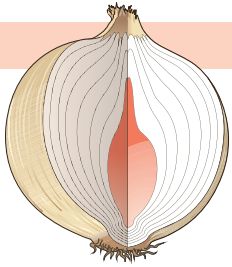
Chapter 17 Computer Security 557

8 In Conclusion 586

Chapter 18 Limitations of Computing 587

CONTENTS

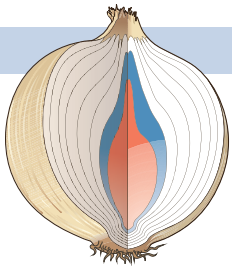
© Amir Debat/Getty Images; © Alan Dyer/Stocktrek Images/Getty Images



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 9
 - A Brief History of Computing Hardware 9
 - A Brief History of Computing Software 19
 - Predictions 25
- 1.3 Computing as a Tool and a Discipline 26
 - The Big Ideas of Computing 27
 - Summary 28
 - Ethical Issues: Digital Divide 30
 - Key Terms 31
 - Exercises 31
 - Thought Questions 33



2 The Information Layer 34

Chapter 2 Binary Values and Number Systems 35

- 2.1 Numbers and Computing 36
- 2.2 Positional Notation 36
 - Binary, Octal, and Hexadecimal 38
 - Arithmetic in Other Bases 41
 - Power-of-2 Number Systems 42
 - Converting from Base 10 to Other Bases 44
 - Binary Values and Computers 45
 - Summary 48
 - Ethical Issues: The FISA Court 49
 - Key Terms 49
 - Exercises 50
 - Thought Questions 53

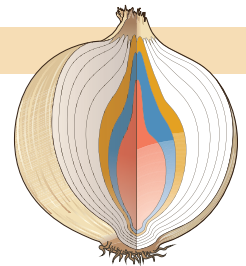
Chapter 3 Data Representation 55

- 3.1 Data and Computers 56
 - Analog and Digital Data 57

- Binary Representations 59
- 3.2** Representing Numeric Data 61
 - Representing Negative Values 61
 - Representing Real Numbers 65
- 3.3** Representing Text 68
 - The ASCII Character Set 69
 - The Unicode Character Set 70
 - Text Compression 71
- 3.4** Representing Audio Data 77
 - Audio Formats 79
 - The MP3 Audio Format 79
- 3.5** Representing Images and Graphics 80
 - Representing Color 80
 - Digitized Images and Graphics 82
 - Vector Representation of Graphics 83
- 3.6** Representing Video 84
 - Video Codecs 84
 - Summary 85
 - Ethical Issues: The Fallout from Snowden's Revelations 86
 - Key Terms 86
 - Exercises 87
 - Thought Questions 91

3 The Hardware Layer 92

- Chapter 4** Gates and Circuits 93
 - 4.1** Computers and Electricity 94
 - 4.2** Gates 96
 - NOT Gate 96
 - AND Gate 97
 - OR Gate 98
 - XOR Gate 98
 - NAND and NOR Gates 99
 - Review of Gate Processing 100
 - Gates with More Inputs 101



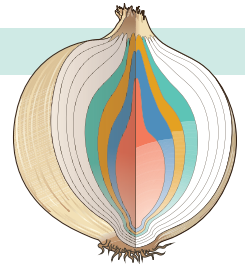
4.3	Constructing Gates	101
	Transistors	102
4.4	Circuits	104
	Combinational Circuits	104
	Adders	108
	Multiplexers	110
4.5	Circuits as Memory	111
4.6	Integrated Circuits	112
4.7	CPU Chips	113
	Summary	113
	Ethical Issues: Codes of Ethics	114
	Key Terms	116
	Exercises	116
	Thought Questions	119
Chapter 5	Computing Components	121
5.1	Individual Computer Components	122
5.2	The Stored-Program Concept	125
	von Neumann Architecture	125
	The Fetch–Execute Cycle	130
	RAM and ROM	133
	Secondary Storage Devices	133
	Touch Screens	137
5.3	Embedded Systems	139
5.4	Parallel Architectures	140
	Parallel Computing	140
	Classes of Parallel Hardware	142
	Summary	143
	Ethical Issues: Is Privacy a Thing of the Past?	144
	Key Terms	144
	Exercises	145
	Thought Questions	147

4 The Programming Layer 148**Chapter 6** Low-Level Programming Languages and Pseudocode 149

- 6.1** Computer Operations 150
- 6.2** Machine Language 150
 - Pep/9: A Virtual Computer 151
 - Input and Output in Pep/9 157
- 6.3** A Program Example 157
 - Pep/9 Simulator 158
 - Another Machine-Language Example 160
- 6.4** Assembly Language 161
 - Pep/9 Assembly Language 162
 - Numeric Data, Branches, and Labels 164
 - Loops in Assembly Language 168
- 6.5** Expressing Algorithms 169
 - Pseudocode Functionality 169
 - Following a Pseudocode Algorithm 173
 - Writing a Pseudocode Algorithm 175
 - Translating a Pseudocode Algorithm 178
- 6.6** Testing 180
 - Summary 181
 - Ethical Issues: Software Piracy 183
 - Key Terms 184
 - Exercises 184
 - Thought Questions 187

Chapter 7 Problem Solving and Algorithms 189

- 7.1** How to Solve Problems 190
 - Ask Questions 191
 - Look for Familiar Things 191
 - Divide and Conquer 192
 - Algorithms 192
 - Computer Problem-Solving Process 194
 - Summary of Methodology 195
 - Testing the Algorithm 196



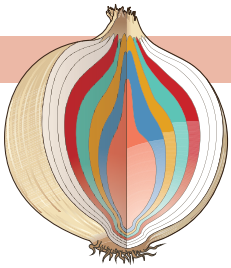
7.2	Algorithms with Simple Variables	197
	An Algorithm with Selection	197
	Algorithms with Repetition	198
7.3	Composite Variables	204
	Arrays	204
	Records	205
7.4	Searching Algorithms	206
	Sequential Search	206
	Sequential Search in a Sorted Array	207
	Binary Search	210
7.5	Sorting	212
	Selection Sort	213
	Bubble Sort	216
	Insertion Sort	218
7.6	Recursive Algorithms	219
	Subprogram Statements	219
	Recursive Factorial	221
	Recursive Binary Search	222
	Quicksort	222
7.7	Important Threads	226
	Information Hiding	226
	Abstraction	227
	Naming Things	228
	Testing	229
	Summary	229
	Ethical Issues: Open-Source Software	230
	Key Terms	232
	Exercises	232
	Thought Questions	237
Chapter 8	Abstract Data Types and Subprograms	239
8.1	What Is an Abstract Data Type?	240
8.2	Stacks	240
8.3	Queues	241
8.4	Lists	242

8.5	Trees	245
	Binary Trees	245
	Binary Search Trees	247
	Other Operations	253
8.6	Graphs	254
	Creating a Graph	256
	Graph Algorithms	257
8.7	Subprograms	263
	Parameter Passing	264
	Value and Reference Parameters	266
	Summary	269
	Ethical Issues: Workplace Monitoring	271
	Key Terms	272
	Exercises	272
	Thought Questions	277
Chapter 9	Object-Oriented Design and High-Level Programming Languages	279
9.1	Object-Oriented Methodology	280
	Object Orientation	280
	Design Methodology	281
	Example	284
9.2	Translation Process	289
	Compilers	290
	Interpreters	290
9.3	Programming Language Paradigms	293
	Imperative Paradigm	293
	Declarative Paradigm	294
9.4	Functionality in High-Level Languages	296
	Boolean Expressions	297
	Data Typing	299
	Input/Output Structures	303
	Control Structures	305
9.5	Functionality of Object-Oriented Languages	311
	Encapsulation	311
	Classes	312

Inheritance 314
 Polymorphism 315

9.6 Comparison of Procedural and Object-Oriented Designs 316

Summary 317
 Ethical Issues: Hoaxes and Scams 319
 Key Terms 320
 Exercises 321
 Thought Questions 325



5 The Operating Systems Layer 326

Chapter 10 Operating Systems 327

10.1 Roles of an Operating System 328

Memory, Process, and CPU Management 330
 Batch Processing 331
 Timesharing 332
 Other OS Factors 333

10.2 Memory Management 334

Single Contiguous Memory Management 336
 Partition Memory Management 337
 Paged Memory Management 339

10.3 Process Management 342

The Process States 342
 The Process Control Block 343

10.4 CPU Scheduling 344

First Come, First Served 345
 Shortest Job Next 346
 Round Robin 346

Summary 348
 Ethical Issues: Medical Privacy: HIPAA 350
 Key Terms 351
 Exercises 352
 Thought Questions 357

Chapter 11 File Systems and Directories 359

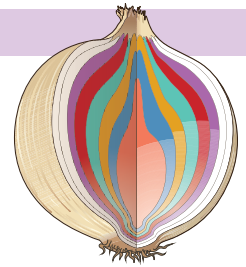
11.1 File Systems 360

Text and Binary Files 360
 File Types 361
 File Operations 363

- File Access 364
- File Protection 365
- 11.2** Directories 366
 - Directory Trees 367
 - Path Names 369
- 11.3** Disk Scheduling 371
 - First-Come, First-Served Disk Scheduling 373
 - Shortest-Seek-Time-First Disk Scheduling 373
 - SCAN Disk Scheduling 374
 - Summary 376
 - Ethical Issues: Privacy: Opt-In or Opt-Out? 378
 - Key Terms 379
 - Exercises 379
 - Thought Questions 383

6 The Applications Layer 384

- Chapter 12** Information Systems 385
 - 12.1** Managing Information 386
 - 12.2** Spreadsheets 387
 - Spreadsheet Formulas 389
 - Circular References 394
 - Spreadsheet Analysis 395
 - 12.3** Database Management Systems 397
 - The Relational Model 397
 - Relationships 401
 - Structured Query Language 402
 - Database Design 403
 - 12.4** E-Commerce 405
 - Cryptocurrency 406
 - Blockchain 406
 - Non-Fungible Tokens (NFTs) 407
 - 12.5** Big Data 408
 - Summary 409
 - Ethical Issues: Politics and the Internet 411
 - Key Terms 413
 - Exercises 413
 - Thought Questions 417

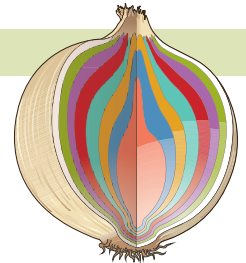


Chapter 13	Artificial Intelligence	419
13.1	Thinking Machines	420
The Turing Test	421	
Aspects of AI	423	
13.2	Knowledge Representation	423
Semantic Networks	425	
Search Trees	427	
13.3	Expert Systems	430
13.4	Neural Networks	432
Biological Neural Networks	432	
Artificial Neural Networks	433	
13.5	Natural-Language Processing	435
Voice Synthesis	436	
Voice Recognition	438	
Natural-Language Comprehension	439	
13.6	Robotics	440
The Sense–Plan–Act Paradigm	441	
Subsumption Architecture	441	
Physical Components	443	
Summary	445	
Ethical Issues: Initial Public Offerings	447	
Key Terms	448	
Exercises	448	
Thought Questions	451	
Chapter 14	Simulation, Graphics, Gaming, and Other Applications	453
14.1	What Is Simulation?	454
Complex Systems	454	
Models	455	
Constructing Models	455	
14.2	Specific Models	457
Queuing Systems	457	
Meteorological Models	461	
Computational Biology	466	

- Other Models 467
- Computing Power Necessary 467
- 14.3** Computer Graphics 468
 - How Light Works 470
 - Object Shape Matters 472
 - Simulating Light 472
 - Modeling Complex Objects 474
 - Getting Things to Move 480
- 14.4** Gaming 482
 - History of Gaming 483
 - Creating the Virtual World 484
 - Game Design and Development 485
 - Game Programming 486
 - Summary 487
 - Ethical Issues: Gaming as an Addiction 489
 - Key Terms 490
 - Exercises 490
 - Thought Questions 493

7 The Communications Layer 494

- Chapter 15** Networks 495
 - 15.1** Networking 496
 - Types of Networks 497
 - Internet Connections 500
 - Packet Switching 502
 - 15.2** Open Systems and Protocols 504
 - Open Systems 504
 - Network Protocols 505
 - TCP/IP 506
 - High-Level Protocols 507
 - MIME Types 508
 - Firewalls 509
 - 15.3** Network Addresses 510
 - Domain Name System 511
 - Who Controls the Internet? 514

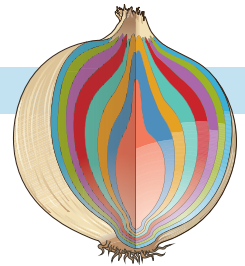


15.4	Cloud Computing	515
	Summary	516
	Ethical Issues: The Effects of Social Networking	519
	Key Terms	520
	Exercises	521
	Thought Questions	523
Chapter 16	The World Wide Web	525
16.1	Spinning the Web	526
	Search Engines	527
	Instant Messaging	528
	Weblogs	529
	Cookies	530
	Web Analytics	530
16.2	HTML and CSS	531
	Basic HTML Elements	535
	Tag Attributes	536
	More About CSS	537
	More HTML5 Elements	540
16.3	Interactive Web Pages	541
	Java Applets	541
	Java Server Pages	542
16.4	XML	543
16.5	Social Network Evolution	547
	Summary	548
	Ethical Issues: Gambling and the Internet	551
	Key Terms	552
	Exercises	552
	Thought Questions	555
Chapter 17	Computer Security	557
17.1	Security at All Levels	558
	Information Security	558

- 17.2** Preventing Unauthorized Access 560
 - Passwords 561
 - CAPTCHA 563
 - Fingerprint and Facial Recognition 564
 - Two-Factor Authentication 565
- 17.3** Malicious Code 565
 - Antivirus Software 566
 - Security Attacks 567
- 17.4** Cryptography 569
- 17.5** Protecting Your Information Online 572
 - Corporate Responsibility 574
 - Security and Portable Devices 574
 - WikiLeaks 575
 - Summary 578
 - Ethical Issues: Blogging and Journalism 580
 - Key Terms 581
 - Exercises 582
 - Thought Questions 585

8**In Conclusion 586****Chapter 18** Limitations of Computing 587

- 18.1** Hardware 588
 - Limits on Arithmetic 588
 - Limits on Components 594
 - Limits on Communications 595
- 18.2** Software 596
 - Complexity of Software 596
 - Current Approaches to Software Quality 597
 - Notorious Software Errors 602
- 18.3** Problems 604
 - Comparing Algorithms 605
 - Turing Machines 612
 - Halting Problem 615
 - Classification of Algorithms 617



Summary	619
Ethical Issues: Therac-25: Anatomy of a Disaster	620
Key Terms	621
Exercises	621
Thought Questions	624
Glossary	625
Endnotes	651
Index	661

PREFACE

© Artur Debat/Getty Images, © Alan Dyer/Stocktrek Images/Getty Images

Choice of Topics

In putting together the outline of topics for this CS0 text, we used many sources. We looked at course catalogue descriptions and book outlines, and we administered a questionnaire designed to find out what you, our colleagues, thought should be included in such a course. We asked you and ourselves to do the following:

- List four topics that you feel students should master in a CS0 course if this is the only computer science course they will take during their college experience.
- List four topics that you would like students entering your CS1 course to have mastered.
- List 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 formed the working outline for this book. It serves as a core CS0 text, providing a breadth-first introduction to computing. It is appropriate for use for a course embracing the AP Computer Science Principles curriculum and alternatively as a companion or lead-in to a programming intensive course.

Rationale for Organization

This book begins with the history of hardware and software, showing how a computer system is like an onion. The processor and its machine language form the heart of the onion, and layers of software and more sophisticated hardware have been added around this heart, layer by layer. 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

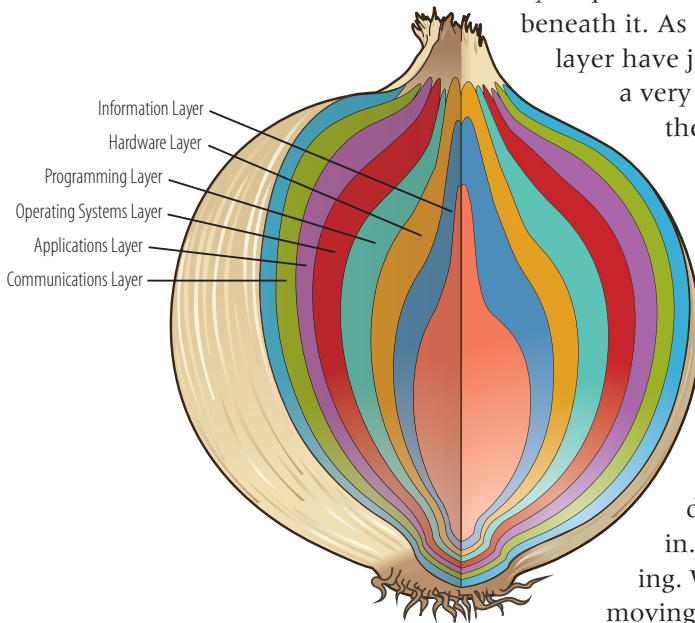
overlay the operating system. Development of these powerful programs was stimulated by theoretical work in computer science, which makes such programs possible. The final layer comprises networks and network software—that is, the tools needed for computers to communicate with one another. The Internet and the World Wide Web put the finishing touches to this layer, and this text culminates with a discussion of security issues affecting our interaction online.

As these layers have grown over the years, the user has become increasingly insulated from the computer system's hardware. Each of these layers provides an abstraction of the computing system beneath it. As each layer has evolved, users of the new layer have joined with users of inner layers to create a very large workforce in the high-tech sector of

the global economy. This book is designed to provide an overview of the layers, introducing the underlying hardware and software technologies, in order to give students an appreciation and understanding of all aspects of computing systems.

Having used history to describe the formation of the onion from the inside out, we were faced with a design choice: We could look at each layer in depth from the inside out or the outside in. The outside-in approach was 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. Thus, 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 the students.



Changes in the Eighth Edition

As always when planning a revision, we asked our colleagues, including many current users of the text, to give us feedback. We appreciate the many thoughtful and insightful responses we received. A variety of changes have been made throughout the book with the *Eighth Edition*.

The summary of the ACM Code of Ethics was changed in Chapter 4 to be a more representative set of guidelines. The laptop specification in Chapter 5 was completely updated with the corresponding changes throughout the discussion of hardware and software components, including an up-to-date emphasis on solid-state disks.

Three new sections were added to Chapter 12: Cryptocurrency, Blockchain (moved from Chapter 17 and updated), and NFTs.

In Chapter 17, the section on Fingerprint Analysis was updated to include Facial Recognition and a new section on Two-Factor Authentication was added.

In addition, the special features throughout the book have been revised and augmented. The “Ethical Issues” sections at the end of each chapter have been brought up to date, addressing the ever-changing state of those issues. Several “Did You Know?” sidebars have been updated. Finally, the biographical sketches throughout the book have been updated.

As with each edition, the entire text has been reviewed for opportunities to improve the coverage, presentation, and examples used. Updated (and sometimes streamlined) text throughout the book helps to clarify the topics presented.

Synopsis

Chapter 1 lays the groundwork, as described in the “Rationale for This Book’s Organization” section above. **Chapters 2** and **3** step back and examine a layer that is embodied in the physical hardware. We call this the “information layer” because it 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 one we 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.

Chapters 4 and 5 discuss 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 like the computer’s central processing unit (CPU) and memory. Chapter 4 covers gates and electronic circuits; Chapter 5 focuses on the hardware components of a computer and how they interact within a von Neumann architecture.

Chapters 6 through 9 examine aspects of the programming layer. Chapter 6 explores the concepts of both machine-language and assembly-language programming using Pep/9, a simulated computer. We discuss the functionality of pseudocode as a way to write algorithms. The concepts of looping and selection are introduced here, expressed in pseudocode, and implemented in Pep/9.

Chapter 7 examines the problem-solving process as it relates to both humans and computers. George Polya’s human problem-solving strategies guide the discussion. Top-down design is presented as a way to design simple algorithms. We choose classic searching and sorting algorithms as the context for the discussion of algorithms. Because algorithms operate on data, we examine ways to structure data so that it can be more efficiently processed. We also introduce subalgorithm (subprogram) statements.

Chapter 8 explores abstract data types and containers: composite structures for which we know only properties or behaviors. Lists, sorted lists, stacks, queues, binary search trees, and graphs are discussed. The section on subalgorithms is expanded to include reference and value parameters and parameter passing.

Chapter 9 covers the concepts of high-level programming languages. Because many prominent high-level languages include functionality associated with object-oriented programming, we detour and

first present this design process. Language paradigms and the compilation process are discussed. Pseudocode concepts are illustrated in brief examples from four programming languages: Python, Visual Basic .NET, C++, and Java.

Chapters 10 and **11** cover the operating systems 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 focuses on file systems, including what they are and how they are managed by the operating system.

Chapters 12 through **14** cover the applications layer. This layer is made up of the general-purpose and specialized application programs that are available to the public for solving programs. We divide this layer into the subdisciplines of computer science upon which these programs are based. Chapter 12 examines information systems; Chapter 13 examines artificial intelligence; and Chapter 14 examines simulation, graphics, gaming, and other applications.

Chapters 15 through **17** cover the communications layer. Chapter 15 presents the theoretical and practical aspects of computers communicating with each other. Chapter 16 discusses the World Wide Web and the various technologies involved. Chapter 17 examines computer security and keeping information protected in the modern information age.

Chapters 2 through 17 are about what a computer can do and how. **Chapter 18** concludes the text with a discussion of the inherent limitations of computer hardware and software, including the problems that can and cannot be solved using a computer. We present Big-O notation as a way to talk about the efficiency of algorithms so that the categories of algorithms can be discussed, and we use the halting problem to show that some problems are unsolvable.

The first and last chapters form bookends: Chapter 1 describes what a computing system is and Chapter 18 cautions about what a computing system is not. The chapters between take an in-depth look at the layers that make up a computing system.

Why Not a Language?

Instead of championing a specific programming language such as Java, C++, or something else, we decided to leave the choice to the user. Introductory chapters, formatted in a manner consistent with the design of this book, are available online for Java, C++, JavaScript, Visual Basic, .NET, Python, SQL, Ruby, Perl, Alice, and Pascal.

If the students have enough knowledge and experience to master the introductory syntax and semantics of a language in addition to the background material in this book, simply have the students download the appropriate chapter. As an alternative, one or all of these chapters can be used to enrich the studies of those who have stronger backgrounds.

Special Features

We have included three special features in this text in order to emphasize the history and breadth of computing as well as the moral obligations that come with new technology.

Biographies

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 who have contributed to the communication layer, such as Doug Engelbart and Tim Berners-Lee. These biographies give students a taste of history and introduce them to the men and women who are pioneers in the world of computing.

10.4 CPU Scheduling

349

Steve Jobs

Born in 1955, Steve Jobs is probably best known for founding Apple Computer together with Steve Wozniak and Ronald Wayne in 1976. At the time, most computers were either mainframes (sometimes as large as a small room) or minicomputers (about the size of a refrigerator), often anything but user-friendly, and almost exclusively used by big businesses. Jobs had a vision of a personal computer that would be accessible to everyone. He is often credited with democratizing the computer.

Jobs and Wozniak designed the Apple I in Jobs's bedroom and built it in the garage of his parents' house. Jobs and Wozniak sold their prize possessions (a Volkswagen microbus and a Hewlett-Packard scientific calculator, respectively) to raise the \$1300 capital with which they founded their company. Four years later, Apple went public. At the end of the first day of trading, the company had a market value of \$1.2 billion.

Jobs headed the team that developed the Apple Macintosh (named after the McIntosh apple), perhaps the most famous of the Apple computers. The Macintosh was the first commercially successful computer to be launched with a graphical user interface. After the launch

Having been ousted from the company he founded, Jobs began another computer company, NeXT, which was purchased by Apple in 1996 for \$402 million. Not only did the acquisition bring Jobs back to his original company, but it also made him CEO of Apple. Under his renewed leadership, Apple launched the iMac, which has been described as the "gold standard of desktop computing."

In 1986, Jobs moved into the field of computer-generated animation when he bought a computer graphics company and renamed it *Pixar*. Pixar has produced a number of box office hits, including *A Bug's Life*, *Toy Story*, *Monsters, Inc.*, and *Finding Nemo*.

Jobs, himself a university dropout, gave the 2005 commencement address at Stanford University, in which he imparted the following piece of career advice to the graduates: "You've got to find what you love."

In 2007 Jobs was named the most powerful person in business by *Fortune* magazine, and Governor Arnold Schwarzenegger inducted Jobs into the California Hall of Fame. In August 2011 Jobs resigned as CEO of Apple; he was elected Chairman of the Board. Tim Cook took over as CEO of Apple. After a long and successful



© Christoph Bechtel/Photo

© Artur Debat/Getty Images; © Alan Dyer/Stocktrek Images/Getty Images

Did You Know

Our second feature (the “Did You Know?” sections indicated by a question mark) comprises sidebars that include interesting tidbits of information from the past, present, and future. They are garnered from history, current events, and the authors’ personal experiences. These little vignettes are designed to amuse, inspire, intrigue, and, of course, educate.

Ethical Issues

Our third feature is an “Ethical Issues” section that is included in each chapter. These sections illustrate the fact that along with the advantages of computing come responsibilities for and consequences of its use. Privacy, hacking, viruses, and free speech are among the topics discussed. Following the exercises in each chapter, a “Thought Questions” section asks stimulating questions about 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 opening spread for each chapter shows an image of the onion in which the outermost color corresponds to the current layer. This color is repeated in header bars and section numbers throughout the layer. Each opening spread also visually indicates where the chapter is within the layer and the book.

We have said that the first and last chapters form bookends. Although they are not part of the layers of the computing onion, these chapters are

?

Virtual games and national security

U.S. and British spies have infiltrated the fantasy world of virtual games. A 2008 National Security Agency (NSA) document declared that virtual games provide a “target-rich communication network” that allows intelligence suspects a way to communicate and “hide in plain sight.”⁴



ETHICAL ISSUES

The FISA Court

The United States Foreign Intelligence Surveillance Court is a U.S. federal court that was established under the Foreign Intelligence Surveillance Act of 1978 (FISA). The Court handles requests by federal law enforcement agencies for surveillance warrants against suspected foreign intelligence agents operating inside the United States.⁴

Before 2013, when Edward Snowden leaked that the Court had ordered a subsidiary of Verizon to provide detailed call records to the National Security Agency (NSA), most people had never heard of the FISA Court.

The FISA Court comprises 11 judges who sit for 7-year terms. The Chief Justice of the Supreme Court appoints the judges, without confirmation. An application for an electronic surveillance warrant is made before one of the judges. The court may amend this application before granting the warrant. If the application is denied, the government may not take the same request to another judge. If

the U.S. Attorney General determines that an emergency exists, he or she may authorize the electronic surveillance but must notify a Court judge not more than 72 hours after the authorization. The USA PATRIOT Act of 2001 expanded the time periods during which surveillance may be authorized.⁵

In December 2012, President Obama signed the FISA Amendments Act Reauthorization Act of 2012, which extended Title VII of FISA until December 31, 2017. In January, 2018, President Trump reauthorized it again through 2023.

Title VII of FISA, added by the FISA Amendments Act of 2008, created separate procedures for targeting suspected foreign intelligence agents, including non-U.S. persons and U.S. persons reasonably believed to be outside the United States.⁶

Note that the stated intent of the FISA Court is to protect the United States as well as the rights of U.S. citizens.

color coded like the others. 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. You know at a glance whether the discussion is at the logical (algorithmic) level or at the programming-language level. In order to distinguish visually between an address and the contents of an address, we color addresses in orange.

Color is especially useful in Chapter 6, “Low-Level Programming Languages and Pseudocode.” Instructions are color coded to differentiate the parts of an instruction. The operation code is blue, the register designation is clear, and the addressing mode specifier is green. Operands are shaded gray. As in other chapters, addresses are in orange.

Instructor’s Resources

For the instructor, slides in PowerPoint format, a test bank, and answers to the book’s end-of-chapter exercises are available for free download at <http://go.jblearning.com/CSI8e>.

ACKNOWLEDGMENTS

© Artur Debat/Getty Images; © Alan Dyer/Stocktrek Images/Getty Images

Our adopters have been most helpful during this revision. To those who took the time to respond to our online survey: Thanks to all of you. We are also grateful to the reviewers of the previous editions of the text:

Warren W. Sheaffer, Saint Paul College
Tim Bower, Kansas State University
Terri Grote, St. Louis Community College
Susan Glenn, Gordon State College
Simon Sultana, Fresno Pacific University
Shruti Nagpal, Worcester State University
Sandy Keeter, Seminole State College of Florida
S. Monisha Pulimood, The College of New Jersey
Roy Thelin, Ridge Point High School
Robert Yacobellis, Loyola University Chicago
Richard Croft, Eastern Oregon University
Raymond J. Curts, George Mason University
Mia Moore, Clark Atlanta University
Melissa Stange, Lord Fairfax Community College
Matthew Hayes, Xavier University of Louisiana
Marie Arvi, Salisbury University
Linda Ehley, Alverno College
Keson Khieu, Sierra College
Keith Coates, Drury University
Joon Kim, Brentwood School
Joe Melanson, Corning Painted Post
Joan Lucas, College at Brockport, the State University of New York
Jeffrey L. Lehman, Huntington University
Janet Helwig, Dominican University
James Thomas Davis, A. Crawford Mosley High School
James Hicks, Los Angeles Southwest College
Jacquilin Porter, Henry Ford College and Baker College
Homer Sharafi, Northern Virginia Community College
Gil Eckert, Monmouth University
Gary Monnard, St. Ambrose University
David Klein, Hofstra University
David Adams, Grove City College
Christopher League, Long Island University Brooklyn

Carol Sweeney, Villa Maria Academy High School
Brian Bradshaw, Thiel College
Bill Cole, Sierra College-Rocklin
Aparna Mahadev, Worcester State University
Anna Ursyn, University of Northern Colorado
Ann Wojnar, Beaumont School

Special thanks to Jeffrey McConnell of Canisius College, who wrote the graphics section in Chapter 14; Herman Tavani of Rivier College, who helped us revise the “Ethical Issues” sections; Richard Spinello of Boston College for his essay on the ethics of blogging; and Paul Toprac, Associate Director of Game Development at the University of Texas at Austin, for his contributions on gaming.

We appreciate and thank our reviewers and colleagues who provided advice and recommendations for the content in this *Eighth Edition*:

Tim Bower, Kansas State University
Brian Bradshaw, Thiel College
Keith Coates, Drury University
Raymond J. Curts, George Mason University
James Thomas Davis, A. Crawford Mosley High School
Gil Eckert, Monmouth University
Susan Glenn, Gordon State College
Terri Grote, St. Louis Community College
Matthew Hayes, Xavier University of Louisiana
Sandy Keeter, Seminole State College of Florida
Keson Khieu, Sierra College
Joon Kim, Brentwood School
David Klein, Hofstra University
Christopher League, Long Island University Brooklyn
Jeffrey L. Lehman, Huntington University
Joan Lucas, College at Brockport, the State University of New York
Joe Melanson, Corning Painted Post
Gary Monnard, St. Ambrose University
Shruti Nagpal, Worcester State University
Jacquilin Porter, Henry Ford College and Baker College
Homer Sharafi, Northern Virginia Community College

Melissa Strange, Lord Fairfax Community College
Simon Sultana, Fresno Pacific University
Carol Sweeney, Villa Maria Academy High School
Roy Thelin, Ridge Point High School
Anna Ursyn, University of Northern Colorado
Ann Wojnar, Beaumont School

Of course, this book wouldn't be possible without the hard work of the skilled professionals at Jones & Bartlett Learning. They include: Ray Chew, Director of Product Management; Orsolya Gall, Manager of Content Strategy; Sara Holloway, Content Strategist; Belinda Thresher, Project Manager; Angela Dooley, Digital Project Specialist; Faith Brosnan, Media Development Editor; Liz Kincaid, Rights Specialist; and Kristin Parker, Cover & Text Designer. Thank you!

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

—ND

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

—JL

SPECIAL FEATURES

© Amal Debat/Getty Images, © Alan Dyer/Stocktrek Images/Getty Images

Interspersed throughout *Computer Science Illuminated, Eighth Edition*, are two special features of note: Ethical Issues and Biographies. A list of each is provided below for immediate access.

ETHICAL ISSUES

- Digital Divide (Chapter 1, p. 30)
- The FISA Court (Chapter 2, p. 49)
- The Fallout from Snowden's Revelations (Chapter 3, p. 86)
- Codes of Ethics (Chapter 4, p. 114)
- Is Privacy a Thing of the Past? (Chapter 5, p. 146)
- Software Piracy (Chapter 6, p. 185)
- Open-Source Software (Chapter 7, p. 232)
- Workplace Monitoring (Chapter 8, p. 273)
- Hoaxes and Scams (Chapter 9, p. 321)
- Medical Privacy: HIPAA (Chapter 10, p. 352)
- Privacy: Opt-In or Opt-Out? (Chapter 11, p. 380)
- Politics and the Internet (Chapter 12, p. 411)
- Initial Public Offerings (Chapter 13, p. 447)
- Gaming as an Addiction (Chapter 14, p. 489)
- The Effects of Social Networking (Chapter 15, p. 519)
- Gambling and the Internet (Chapter 16, p. 551)
- Blogging and Journalism (Chapter 17, p. 580)
- Therac-25: Anatomy of a Disaster (Chapter 18, p. 620)

BIOGRAPHIES

Ada Lovelace, the First Programmer (Chapter 1, p. 14)

Grace Murray Hopper (Chapter 2, p. 46)

Bob Bemer (Chapter 3, p. 72)

George Boole (Chapter 4, p. 95)

John Vincent Atanasoff (Chapter 5, p. 127)

Konrad Zuse (Chapter 6, p. 181)

George Polya (Chapter 7, p. 195)

John von Neumann (Chapter 8, p. 248)

Edsger Dijkstra (Chapter 9, p. 309)

Steve Jobs (Chapter 10, p. 349)

Tony Hoare (Chapter 11, p. 377)

Daniel Bricklin (Chapter 12, p. 395)

Herbert A. Simon (Chapter 13, p. 424)

Ivan Sutherland (Chapter 14, p. 462)

Doug Engelbart (Chapter 15, p. 503)

Tim Berners-Lee (Chapter 16, p. 538)

Mavis Batey (Chapter 17, p. 576)

Alan Turing (Chapter 18, p. 613)