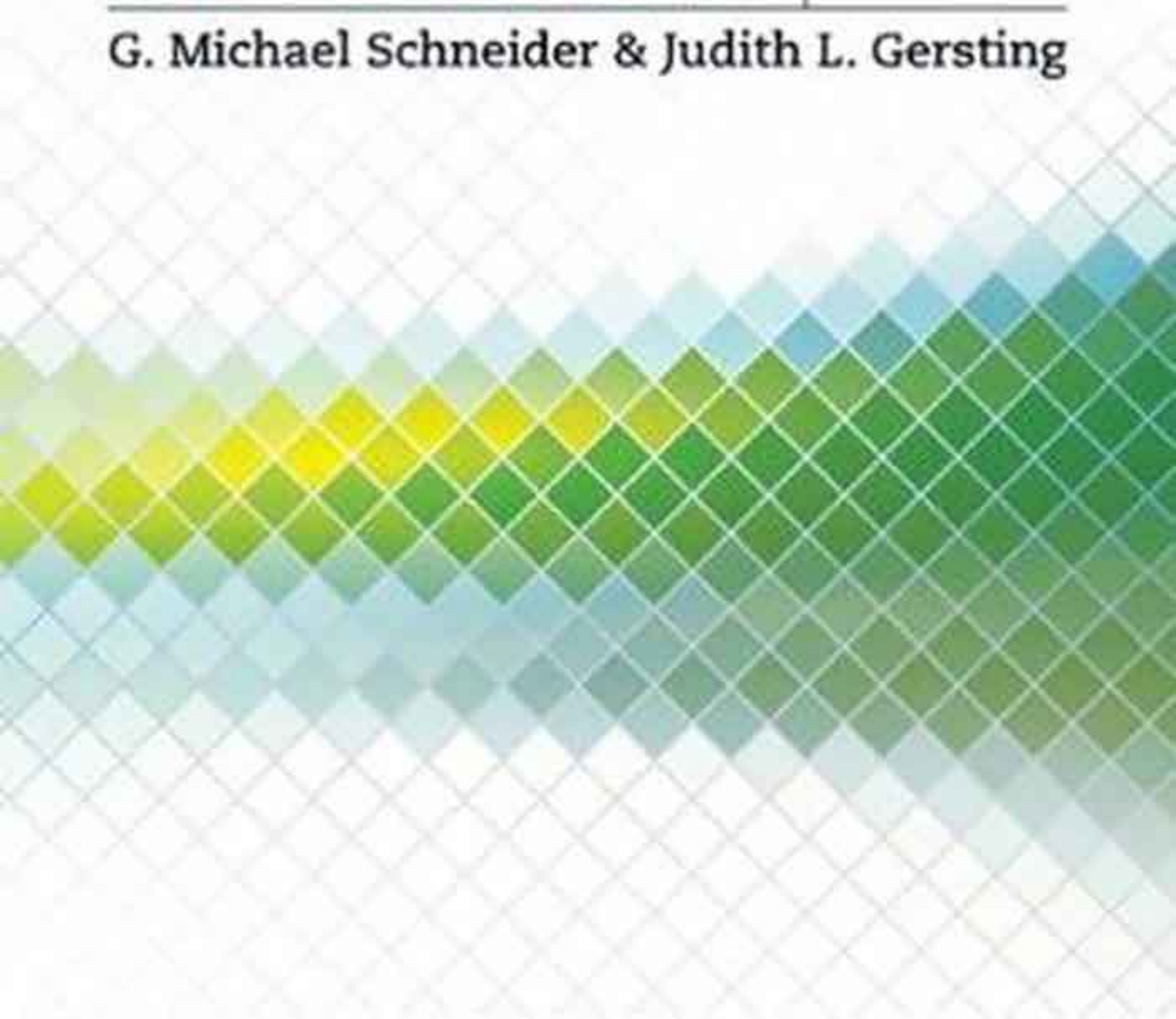


INVITATION TO **COMPUTER SCIENCE**

SEVENTH
EDITION

G. Michael Schneider & Judith L. Gersting



7TH EDITION

Invitation to Computer Science



7TH EDITION

Invitation to Computer Science

G. Michael Schneider
Macalester College

Judith L. Gersting
Indiana University-Purdue University
at Indianapolis

Contributing author:
Bo Brinkman
Miami University



Australia • Brazil • Japan • Korea • Mexico • Singapore • Spain • United Kingdom • United States

This is an electronic version of the print textbook. Due to electronic rights restrictions, some third party content may be suppressed. Editorial review has deemed that any suppressed content does not materially affect the overall learning experience. The publisher reserves the right to remove content from this title at any time if subsequent rights restrictions require it. For valuable information on pricing, previous editions, changes to current editions, and alternate formats, please visit www.cengage.com/highered to search by ISBN#, author, title, or keyword for materials in your areas of interest.

Important Notice: Media content referenced within the product description or the product text may not be available in the eBook version.

**Invitation to Computer Science,
Seventh Edition**

G. Michael Schneider and Judith L. Gersting

Product Director: Kathleen McMahon

Senior Content Developer: Alyssa Pratt

Development Editor: Deb Kaufmann

Marketing Manager: Eric LaScola

Manufacturing Planner: Julio Esperas

Art Director: Jack Pendleton

Production Management, Copyediting and
Composition: Integra Software Services
Pvt. Ltd.

Cover Photo: Studio-Pro/Gettyimages

© 2010, 2013, 2016 Cengage Learning

WCN: 02-200-203

ALL RIGHTS RESERVED. No part of this work covered by the copyright herein may be reproduced, transmitted, stored, or used in any form or by any means graphic, electronic, or mechanical, including but not limited to photocopying, recording, scanning, digitizing, taping, Web distribution, information networks, or information storage and retrieval systems, except as permitted under Section 107 or 108 of the 1976 United States Copyright Act, without the prior written permission of the publisher.

For product information and technology assistance, contact us at
Cengage Learning Customer & Sales Support, 1-800-354-9706

For permission to use material from this text or product, submit all
requests online at www.cengage.com/permissions.

Further permissions questions can be e-mailed to
permissionrequest@cengage.com.

Library of Congress Control Number: 2014957763

ISBN: 978-1-305-07577-1

Cengage Learning

20 Channel Center Street

Boston, MA 02210

USA

Cengage Learning is a leading provider of customized learning solutions with office locations around the globe, including Singapore, the United Kingdom, Australia, Mexico, Brazil, and Japan. Locate your local office at www.cengage.com/global.

Cengage Learning products are represented in Canada by Nelson Education, Ltd.

All images © 2016 Cengage Learning®. All rights reserved.

To learn more about Cengage Learning Solutions, visit www.cengage.com.

Purchase any of our products at your local college store or at our preferred online store www.cengagebrain.com.

To my wife, Ruthann, our children, Benjamin, Rebecca, and Trevor, grandson, Liam, and granddaughter, Sena.

G. M. S.

To my husband, John, and to: Adam and Francine; Jason, Cathryn, Sammie, and Johnny.

J. L. G.

Brief Contents

Chapter 1 An Introduction to
Computer Science 2

LEVEL 1 The Algorithmic Foundations of
Computer Science 42

Chapter 2 Algorithm Discovery and Design 44

Chapter 3 The Efficiency of Algorithms 92

LEVEL 2 The Hardware World 150

Chapter 4 The Building Blocks: Binary Numbers,
Boolean Logic, and Gates 152

Chapter 5 Computer Systems Organization 222

LEVEL 3 The Virtual Machine 278

Chapter 6 An Introduction to System Software
and Virtual Machines 280

Chapter 7 Computer Networks and Cloud
Computing 336

Chapter 8 Information Security 394

LEVEL 4 The Software World 432

Chapter 9 Introduction to High-Level Language
Programming 434

Chapter 10 The Tower of Babel 480

Chapter 11 Compilers and Language
Translation 540

Chapter 12 Models of Computation 586

LEVEL 5	Applications	634
	Chapter 13	Simulation and Modeling 636
	Chapter 14	Electronic Commerce, Databases, and Personal Privacy 668
	Chapter 15	Artificial Intelligence 708
	Chapter 16	Computer Graphics and Entertainment: Movies, Games, and Virtual Communities 752

LEVEL 6	Social Issues in Computing	782
	Chapter 17	Making Decisions about Computers, Information, and Society 784

Answers to Practice Problems 823

Index 867

Online Chapters

This text includes five language-specific online-only downloadable chapters on Ada, C++, C#, Java, and Python, available at www.cengagebrain.com (search for the ISBN of this book) and on the CourseMate for this text.

Contents



Preface to the Seventh Edition xxi

Chapter 1 An Introduction to Computer Science 2

1.1 Introduction 2

Special Interest Box: In the Beginning ... 5

1.2 The Definition of Computer Science 5

Special Interest Box: Abu Ja'far Muhammad ibn Musa Al-Khowarizmi (AD 780–850?) 9

1.3 Algorithms 11

1.3.1 The Formal Definition of an Algorithm 11

1.3.2 The Importance of Algorithmic Problem Solving 16

PRACTICE PROBLEMS 17

1.4 A Brief History of Computing 18

1.4.1 The Early Period: Up to 1940 18

Special Interest Box: The Original “Technophobia” 21

Special Interest Box: Charles Babbage (1791–1871)

Ada Augusta Byron, Countess of Lovelace (1815–1852) 23

1.4.2 The Birth of Computers: 1940–1950 24

Special Interest Box: John Von Neumann (1903–1957) 27

1.4.3 The Modern Era: 1950 to the Present 28

Special Interest Box: And the Verdict Is ... 28

Special Interest Box: The World's First Microcomputer 30

1.5 Organization of the Text 33

LABORATORY EXPERIENCE 1 37

EXERCISES 38

CHALLENGE WORK 40

ADDITIONAL RESOURCES 41

LEVEL 1

The Algorithmic Foundations of Computer Science 42

Chapter 2 Algorithm Discovery and Design 44

2.1 Introduction 44

2.2 Representing Algorithms 44

2.2.1 Pseudocode 44

2.2.2 Sequential Operations 47

	PRACTICE PROBLEMS	50
	2.2.3 Conditional and Iterative Operations	51
	<i>Special Interest Box: From Little Primitives Mighty Algorithms Grow</i>	60
	2.3 Examples of Algorithmic Problem Solving	60
	2.3.1 Example 1: Go Forth and Multiply	60
	PRACTICE PROBLEMS	61
	PRACTICE PROBLEMS	64
	2.3.2 Example 2: Looking, Looking, Looking	65
	LABORATORY EXPERIENCE 2	70
	2.3.3 Example 3: Big, Bigger, Biggest	70
	PRACTICE PROBLEMS	76
	LABORATORY EXPERIENCE 3	76
	2.3.4 Example 4: Meeting your Match	77
	2.4 Conclusion	83
	PRACTICE PROBLEMS	84
	EXERCISES	85
	CHALLENGE WORK	88
	ADDITIONAL RESOURCES	91
Chapter 3	The Efficiency of Algorithms	92
	3.1 Introduction	92
	3.2 Attributes of Algorithms	92
	PRACTICE PROBLEMS	97
	3.3 Measuring Efficiency	97
	3.3.1 Sequential Search	97
	3.3.2 Order of Magnitude—Order n	100
	<i>Special Interest Box: Flipping Pancakes</i>	102
	3.3.3 Selection Sort	102
	PRACTICE PROBLEM	103
	PRACTICE PROBLEMS	109
	3.3.4 Order of Magnitude—Order n^2	109
	<i>Special Interest Box: The Tortoise and the Hare</i>	113
	LABORATORY EXPERIENCE 4	114
	PRACTICE PROBLEM	115
	3.4 Analysis of Algorithms	115
	3.4.1 Data Cleanup Algorithms	115
	3.4.2 Binary Search	123
	PRACTICE PROBLEMS	123
	PRACTICE PROBLEMS	129
	LABORATORY EXPERIENCE 5	130
	3.4.3 Pattern Matching	130
	3.4.4 Summary	131
	PRACTICE PROBLEM	132
	3.5 When Things Get Out of Hand	132
	PRACTICE PROBLEMS	137
	3.6 Summary of Level 1	137

LABORATORY EXPERIENCE 6	138
EXERCISES	139
CHALLENGE WORK	149
ADDITIONAL RESOURCES	149

LEVEL 2 The Hardware World 150

Chapter 4	The Building Blocks: Binary Numbers, Boolean Logic, and Gates	152
4.1	Introduction	152
4.2	The Binary Numbering System	152
4.2.1	Binary Representation of Numeric and Textual Information	152
	<i>Special Interest Box: A Not So Basic Base</i>	158
4.2.2	Binary Representation of Sound and Images	165
	PRACTICE PROBLEMS	166
	PRACTICE PROBLEMS	175
4.2.3	The Reliability of Binary Representation	176
4.2.4	Binary Storage Devices	177
	<i>Special Interest Box: Moore's Law and the Limits of Chip Design</i>	182
4.3	Boolean Logic and Gates	183
4.3.1	Boolean Logic	183
	PRACTICE PROBLEMS	187
4.3.2	Gates	188
	<i>Special Interest Box: George Boole (1815–1864)</i>	192
4.4	Building Computer Circuits	193
4.4.1	Introduction	193
4.4.2	A Circuit Construction Algorithm	195
	PRACTICE PROBLEMS	199
4.4.3	Examples of Circuit Design and Construction	200
	LABORATORY EXPERIENCE 7	200
	LABORATORY EXPERIENCE 8	208
	PRACTICE PROBLEMS	209
	<i>Special Interest Box: Dr. William Shockley (1910–1989)</i>	209
4.5	Control Circuits	211
4.6	Conclusion	215
	EXERCISES	217
	CHALLENGE WORK	219
	ADDITIONAL RESOURCES	221
Chapter 5	Computer Systems Organization	222
5.1	Introduction	222
5.2	The Components of a Computer System	224
5.2.1	Memory and Cache	226
	<i>Special Interest Box: Powers of 10</i>	230

PRACTICE PROBLEMS	238
5.2.2 Input/Output and Mass Storage	239
PRACTICE PROBLEMS	244
5.2.3 The Arithmetic/Logic Unit	245
5.2.4 The Control Unit	250
PRACTICE PROBLEMS	256
5.3 Putting the Pieces Together—the Von Neumann Architecture	258
<i>Special Interest Box: An Alphabet Soup of Speed Measures: MHz, GHz, MIPS, and GFLOPS</i>	264
LABORATORY EXPERIENCE 9	265
5.4 Non-Von Neumann Architectures	265
<i>Special Interest Box: Speed to Burn</i>	269
5.5 Summary of Level 2	271
<i>Special Interest Box: Quantum Computing</i>	272
EXERCISES	273
CHALLENGE WORK	275
ADDITIONAL RESOURCES	277

LEVEL 3 The Virtual Machine 278

Chapter 6 An Introduction to System Software and Virtual Machines	280
6.1 Introduction	280
6.2 System Software	282
6.2.1 The Virtual Machine	282
6.2.2 Types of System Software	283
6.3 Assemblers and Assembly Language	286
6.3.1 Assembly Language	286
PRACTICE PROBLEMS	294
6.3.2 Examples of Assembly Language Code	295
PRACTICE PROBLEMS	299
LABORATORY EXPERIENCE 10	300
6.3.3 Translation and Loading	300
PRACTICE PROBLEMS	307
6.4 Operating Systems	308
6.4.1 Functions of an Operating System	308
<i>Special Interest Box: A Machine for the Rest of Us</i>	311
PRACTICE PROBLEMS	315
6.4.2 Historical Overview of Operating Systems Development	318
<i>Special Interest Box: Now That's Big!</i>	320
6.4.3 The Future	327
<i>Special Interest Box: Gesture-Based Computing</i>	330
EXERCISES	330
CHALLENGE WORK	333
ADDITIONAL RESOURCES	335

Chapter 7	Computer Networks and Cloud Computing	336
7.1	Introduction	336
7.2	Basic Networking Concepts	337
7.2.1	Communication Links	338
7.2.2	Local Area Networks	344
	<i>Special Interest Box: The Internet of Things</i>	345
	PRACTICE PROBLEMS	346
	PRACTICE PROBLEMS	349
7.2.3	Wide Area Networks	349
7.2.4	Overall Structure of the Internet	351
	<i>Special Interest Box: Firewalls</i>	354
7.3	Communication Protocols	356
7.3.1	Physical Layer	357
7.3.2	Data Link Layer	358
	PRACTICE PROBLEMS	362
7.3.3	Network Layer	363
	<i>Special Interest Box: I Can't Believe We've Run Out</i>	364
7.3.4	Transport Layer	366
	PRACTICE PROBLEMS	367
7.3.5	Application Layer	371
7.4	Network Services and Benefits	374
	LABORATORY EXPERIENCE 11	375
7.4.1	Interpersonal Communications	375
7.4.2	Social Networking	376
7.4.3	Resource Sharing	376
7.4.4	Electronic Commerce	378
7.5	Cloud Computing	379
7.6	A History of the Internet and the World Wide Web	382
7.6.1	The Internet	382
7.6.2	The World Wide Web	387
	<i>Special Interest Box: Geography Lesson</i>	388
	<i>Special Interest Box: Net Neutrality</i>	389
7.7	Conclusion	390
	EXERCISES	390
	CHALLENGE WORK	393
	ADDITIONAL RESOURCES	393
Chapter 8	Information Security	394
8.1	Introduction	394
8.2	Threats and Defenses	395
8.2.1	Authentication and Authorization	395
	<i>Special Interest Box: The Metamorphosis of Hacking</i>	396
	<i>Special Interest Box: Password Pointers</i>	400
	PRACTICE PROBLEMS	401
8.2.2	Threats from the Network	402
	<i>Special Interest Box: Beware the Trojan Horse</i>	403
	<i>Special Interest Box: your Money or your Files</i>	405

<i>Special Interest Box: Defense against the Dark Arts</i>	406
PRACTICE PROBLEM	407
8.2.3 White Hats vs. Black Hats	407
8.3 Encryption	407
8.3.1 Encryption Overview	407
<i>Special Interest Box: you've Been Hacked</i>	408
8.3.2 Simple Encryption Algorithms	409
PRACTICE PROBLEMS	412
LABORATORY EXPERIENCE 12	413
8.3.3 DES	413
<i>Special Interest Box: Hiding in Plain Sight</i>	413
8.3.4 Public-Key Systems	417
<i>Special Interest Box: your Secret Is Safe with Me</i>	419
PRACTICE PROBLEM	419
8.4 Web Transmission Security	420
<i>Special Interest Box: The Heartbleed Bug</i>	421
8.5 Embedded Computing	422
<i>Special Interest Box: Google Glass</i>	424
8.6 Conclusion	426
8.7 Summary of Level 3	426
EXERCISES	427
CHALLENGE WORK	429
ADDITIONAL RESOURCES	431

LEVEL 4 The Software World 432

Chapter 9	Introduction to High-Level Language Programming	434
9.1	The Language Progression	434
9.1.1	Where Do We Stand and What Do We Want?	434
9.1.2	Getting Back to Binary	437
9.2	A Family of Languages	439
	<i>Special Interest Box: Ada, C++, C#, Java, and Python Online Chapters</i>	439
9.3	Two Examples in Five-Part Harmony	440
9.3.1	Favorite Number	440
9.3.2	Data Cleanup (Again)	444
9.4	Feature Analysis	454
9.5	Meeting Expectations	454
9.6	The Big Picture: Software Engineering	463
9.6.1	Scaling Up	464
9.6.2	The Software Development Life Cycle	464
	<i>Special Interest Box: Vital Statistics for Real Code</i>	466
9.6.3	Modern Environments	472
9.6.4	Agile Software Development	473

<i>Special Interest Box: Healthcare.gov</i>	474
9.7 Conclusion	476
EXERCISES	477
CHALLENGE WORK	477
ADDITIONAL RESOURCES	479

Online Chapters

This text includes five language-specific online-only downloadable chapters on Ada, C++, C#, Java, and Python, available at www.cengagebrain.com (search for the ISBN of this book) and on the CourseMate for this text.

Chapter 10	The Tower of Babel	480
	10.1 Why Babel?	480
	10.2 Procedural Languages	482
	10.2.1 Plankalkül	482
	10.2.2 Fortran	482
	<i>Special Interest Box: Old Dog, New Tricks #1</i>	484
	PRACTICE PROBLEMS	485
	10.2.3 COBOL	485
	<i>Special Interest Box: Uncle Sam Wants Who?</i>	487
	PRACTICE PROBLEM	487
	10.2.4 C/C++	487
	PRACTICE PROBLEMS	491
	10.2.5 Ada	492
	PRACTICE PROBLEM	493
	10.2.6 Java	493
	10.2.7 Python	495
	PRACTICE PROBLEM	495
	10.2.8 C# and .NET	496
	PRACTICE PROBLEM	496
	<i>Special Interest Box: The "Popularity" Contest</i>	497
	<i>Special Interest Box: Old Dog, New Tricks #2</i>	499
	PRACTICE PROBLEM	500
	10.3 Special-Purpose Languages	500
	10.3.1 SQL	500
	10.3.2 HTML	501
	LABORATORY EXPERIENCE 13	504
	10.3.3 JavaScript	504
	<i>Special Interest Box: Beyond HTML</i>	505
	<i>Special Interest Box: PHP</i>	508
	PRACTICE PROBLEMS	508
	10.4 Alternative Programming Paradigms	509
	10.4.1 Functional Programming	510

	<i>Special Interest Box: It's All in How you LOOK, Look, Look, ... at It</i>	514
	PRACTICE PROBLEMS	516
	LABORATORY EXPERIENCE 14	516
	10.4.2 Logic Programming	516
	PRACTICE PROBLEMS	521
	10.4.3 Parallel Programming	522
	<i>Special Interest Box: New Dogs, New Tricks</i>	527
	PRACTICE PROBLEMS	528
	10.5 New Languages Keep Coming	528
	10.5.1 Go	528
	10.5.2 F#	530
	10.5.3 Swift	531
	10.6 Conclusion	532
	EXERCISES	534
	CHALLENGE WORK	537
	ADDITIONAL RESOURCES	539
Chapter 11	Compilers and Language Translation	540
	11.1 Introduction	540
	11.2 The Compilation Process	543
	11.2.1 Phase I: Lexical Analysis	544
	PRACTICE PROBLEMS	548
	11.2.2 Phase II: Parsing	548
	PRACTICE PROBLEMS	554
	PRACTICE PROBLEMS	565
	11.2.3 Phase III: Semantics and Code Generation	566
	PRACTICE PROBLEM	575
	11.2.4 Phase IV: Code Optimization	575
	LABORATORY EXPERIENCE 15	575
	<i>Special Interest Box: "Now I Understand," Said the Machine</i>	580
	11.3 Conclusion	581
	EXERCISES	582
	CHALLENGE WORK	585
	ADDITIONAL RESOURCES	585
Chapter 12	Models of Computation	586
	12.1 Introduction	586
	12.2 What Is a Model?	587
	12.3 A Model of a Computing Agent	588
	12.3.1 Properties of a Computing Agent	588
	PRACTICE PROBLEMS	589
	12.3.2 The Turing Machine	590
	<i>Special Interest Box: Alan Turing, Brilliant Eccentric</i>	591
	PRACTICE PROBLEMS	598
	12.4 A Model of an Algorithm	600

12.5	Turing Machine Examples	602
12.5.1	A Bit Inverter	603
	PRACTICE PROBLEMS	605
12.5.2	A Parity Bit Machine	605
12.5.3	Machines for Unary Incrementing	608
	PRACTICE PROBLEM	608
12.5.4	A Unary Addition Machine	612
	PRACTICE PROBLEMS	614
	LABORATORY EXPERIENCE 16	614
12.6	The Church–Turing Thesis	615
	<i>Special Interest Box: The Turing Award</i>	616
12.7	Unsolvable Problems	619
	<i>Special Interest Box: Couldn't Do, Can't Do, Never Will Be Able to ...</i>	624
	PRACTICE PROBLEMS	624
	LABORATORY EXPERIENCE 17	625
12.8	Conclusion	625
12.9	Summary of Level 4	626
	EXERCISES	627
	CHALLENGE WORK	631
	ADDITIONAL RESOURCES	633

LEVEL 5 Applications 634

Chapter 13	Simulation and Modeling	636
13.1	Introduction	636
13.2	Computational Modeling	636
13.2.1	Introduction to Systems and Models	636
13.2.2	Computational Models, Accuracy, and Errors	638
13.2.3	An Example of Model Building	642
	PRACTICE PROBLEMS	651
	LABORATORY EXPERIENCE 18	652
13.3	Running the Model and Visualizing Results	652
13.4	Conclusion	662
	<i>Special Interest Box: The Mother of All Computations!</i>	662
	EXERCISES	663
	CHALLENGE WORK	665
	ADDITIONAL RESOURCES	667
Chapter 14	Electronic Commerce, Databases, and Personal Privacy	668
14.1	Introduction	668
14.2	Ecommerce	669
	<i>Special Interest Box: Shopping on the Web</i>	670

14.2.1	The Vision Thing	671
14.2.2	Decisions, Decisions	672
14.2.3	Anatomy of a Transaction	673
	<i>Special Interest Box: A Rose by Any Other Name...</i>	675
14.2.4	Designing your Website	679
	<i>Special Interest Box: Less Is More</i>	681
14.2.5	Behind the Scenes	682
	PRACTICE PROBLEMS	683
14.2.6	Other Models	683
14.3	Databases	686
14.3.1	Data Organization	686
14.3.2	Database Management Systems	688
14.3.3	Other Considerations	695
	<i>Special Interest Box: Think Big!</i>	696
	PRACTICE PROBLEMS	697
	LABORATORY EXPERIENCE 19	698
14.4	Personal Privacy	698
	<i>Special Interest Box: What your Smartphone Photo Knows</i>	702
	<i>Special Interest Box: you Have the Right to Be Forgotten</i>	703
	PRACTICE PROBLEM	704
14.5	Conclusion	704
	EXERCISES	705
	CHALLENGE WORK	707
	ADDITIONAL RESOURCES	707

Chapter 15 Artificial Intelligence 708

15.1	Introduction	708
	<i>Special Interest Box: Victory in the Turing Test</i>	710
15.2	A Division of Labor	711
15.3	Knowledge Representation	713
	PRACTICE PROBLEMS	717
15.4	Recognition Tasks	717
	<i>Special Interest Box: Brain on a Chip?</i>	723
	LABORATORY EXPERIENCE 20	723
	PRACTICE PROBLEMS	724
15.5	Reasoning Tasks	724
15.5.1	Intelligent Searching	724
15.5.2	Swarm Intelligence	727
	<i>Special Interest Box: Robot Swarms</i>	728
15.5.3	Intelligent Agents	728
	<i>Special Interest Box: To Whom Am I Speaking?</i>	730
15.5.4	Expert Systems	731
	PRACTICE PROBLEMS	733
15.5.5	The Games We Play	734
	<i>Special Interest Box: Captured by CAPTCHA</i>	737

15.6	Robots and Drones	740
15.6.1	Robots	740
15.6.2	Drones	744
15.7	Conclusion	745
	ExERCISES	746
	CHALLENGE WORK	748
	ADDITIONAL RESOURCES	751

Chapter 16 Computer Graphics and Entertainment: Movies, Games, and Virtual Communities 752

16.1	Introduction	752
16.2	Computer-Generated Imagery (CGI)	754
16.2.1	Introduction to CGI	754
	<i>Special Interest Box: Computer Horsepower</i>	756
16.2.2	How It's Done: The Graphics Pipeline	757
16.2.3	Object Modeling	757
16.2.4	Object Motion	760
	PRACTICE PROBLEM	761
	PRACTICE PROBLEM	765
16.2.5	Rendering and Display	765
16.2.6	The Future of CGI	769
16.3	Video Gaming	769
	<i>Special Interest Box: The Good, the Bad, and the Ugly</i>	772
16.4	Multiplayer Games and Virtual Communities	773
16.5	Conclusion	776
	<i>Special Interest Box: The Computer Will See you Now</i>	777
16.6	Summary of Level 5	778
	ExERCISES	778
	CHALLENGE WORK	781
	ADDITIONAL RESOURCES	781

LEVEL 6 Social Issues in Computing 782

Chapter 17 Making Decisions about Computers, Information, and Society 784

17.1	Introduction	784
17.2	Case Studies	784
17.2.1	Case 1: Is Sharing Good?	784
	<i>Special Interest Box: Death of a Dinosaur</i>	789
	PRACTICE PROBLEMS	792
	<i>Special Interest Box: The Sound of Music</i>	793
17.2.2	Case 2: The Athens Affair—Privacy vs. Security	793
	<i>Special Interest Box: Hero or Traitor?</i>	795
	PRACTICE PROBLEMS	801

17.2.3	Case 3: Hackers—Public Enemies or Gadflies?	802
	PRACTICE PROBLEMS	807
17.2.4	Thinking Straight about Technology and Ethics	808
	<i>Special Interest Box</i> : Professional Codes of Conduct	808
17.2.5	Case 4: Genetic Information and Medical Research	809
17.3	Personal Privacy and Social Networks	814
	PRACTICE PROBLEMS	819
17.4	What We Covered and What We Did Not	820
17.5	Summary of Level 6	820
	EXERCISES	821
	ADDITIONAL RESOURCES	822
	<i>Answers to Practice Problems</i>	823
	<i>Index</i>	867

Preface to the Seventh Edition

Overview

This text is intended for a one-semester introductory course in computer science. It presents a breadth-first overview of the discipline that assumes no prior background in computer science, programming, or mathematics. It would be appropriate for a service course for students not majoring in computer science, as well as for schools that implement their introductory course for majors using a breadth-first approach that surveys the fundamental aspects of computer science and establishes a context for subsequent courses. It would be quite suitable for a high school computer science course as well. Previous editions of this text have been used in all these types of courses.

The Non-Majors Service Course

The introductory computer science service course (often called CS 0) has undergone numerous changes. In the 1970s and early 1980s, it was usually a class in Fortran, BASIC, or Pascal programming. In the mid-to-late 1980s, a rapid increase in computer use caused the service course to evolve into something called “computer literacy,” in which students learned about new applications of computing in fields such as business, medicine, law, and education. With the growth of personal computers and productivity software, a typical early to mid-1990s version of this course would spend a semester teaching students how to use word processors, databases, spreadsheets, and email. The most recent change was its evolution into a web-centric course in which students learned to design and implement webpages using technologies such as HTML, XML, ASP, and Java applets.

In many institutions, the computer science service course is evolving once again. There are two reasons for this change. First, virtually all college and high school students are familiar with personal computers and productivity software. They have been using word processors and search engines since elementary school and are familiar with social networks, online retailing, and email; many have designed webpages and even manage their own

websites and blogs. In this day and age, a course that focuses solely on applications will be of little or no interest.

But a more important reason for rethinking the structure of the CS 0 service course, and the primary reason why we authored this book, is the following observation:

Most computer science service courses do not teach students the foundations and fundamental concepts of computer science!

We believe quite strongly that, in addition to applications of information technology, students in a computer science service course should receive a solid grounding in the basic concepts of the discipline. This parallels the structure of introductory courses in biology, physics, and geology, which introduce the central concepts of the fields. Topics in this type of breadth-first computer science service course would not be limited to “fun” applications such as webpage creation, social networking, game design, and interactive graphics, but would also cover foundational issues such as algorithms, hardware, computer organization, system software, language models, theory of computation, and the social and ethical issues of computing. An introduction to these core ideas exposes students to the overall richness and beauty of the field and allows them not only to use computers and software effectively but also to understand and appreciate the basic ideas underlying the discipline of computer science. As a side benefit, students who complete such a course will have a much better idea of what a major or a minor in computer science would entail.

The First Course for Majors

Since the emergence of computer science as an academic discipline in the 1960s, the first course in the major (often called CS 1) has usually been an introduction to programming—from Fortran to BASIC to Pascal, and, later, C++, Java, and Python. But today there are numerous alternatives in the structure of CS 1, including the breadth-first overview. A first course for computer science majors using the breadth-first model emphasizes early exposure to the sub-disciplines of the field rather than placing exclusive emphasis on programming. This gives new majors a more complete and well-rounded understanding of their chosen field of study, including the many concepts and ways of thinking that are part of computer science.

Our book—intended for either majors or non-majors—is organized around this breadth-first approach as it presents a wide range of subject matter drawn from diverse areas of computer science. However, to avoid drowning students in a sea of seemingly unrelated facts and details, a breadth-first presentation must be carefully woven into a cohesive fabric, a theme, a “big picture” that ties together the individual topics and presents computer science as a unified and integrated discipline. To achieve this, our text divides the study of computer science into a hierarchy of topics, with each layer in the hierarchy building upon concepts presented in earlier chapters.

A Hierarchy of Abstractions

The central theme of this book is that *computer science is the study of algorithms*. Our hierarchy utilizes this definition by initially looking at the algorithmic foundations of computer science and then moving upward from this central theme to higher-level issues such as hardware, systems, software, applications, and ethics. Just as the chemist starts from the basic building blocks of protons, neutrons, and electrons and then builds upon these concepts to form atoms, molecules, and compounds, so, too, does our text build from elementary concepts to higher-level ideas.

The six levels in our computer science hierarchy are:

- Level 1. The Algorithmic Foundations of Computer Science
- Level 2. The Hardware World
- Level 3. The Virtual Machine
- Level 4. The Software World
- Level 5. Applications
- Level 6. Social Issues in Computing

Following an introductory chapter, Level 1 (Chapters 2–3) introduces “The Algorithmic Foundations of Computer Science,” the bedrock on which all other aspects of the discipline are built. It presents fundamental ideas such as the design of algorithms, algorithmic problem solving, abstraction, pseudocode, and iteration and illustrates these ideas using well-known examples. It also introduces the concepts of algorithm efficiency and asymptotic growth and demonstrates that not all algorithms are, at least in terms of running time, created equal.

The discussions in Level 1 assume that our algorithms are executed by something called a “computing agent,” an abstract concept for any entity that can carry out the instructions in our solution. However, in Level 2 (Chapters 4–5), “The Hardware World,” we now want our algorithms to be executed by “real” computers to produce “real” results. Thus begins our discussion of hardware, logic design, and computer organization. The initial discussion introduces the basic building blocks of computer systems—binary numbers, Boolean logic, gates, and circuits. It then shows how these elementary concepts can be combined to construct a real computer using the Von Neumann architecture, composed of processors, memory, and input/output. This level presents a simple machine language instruction set and explains how the algorithmic primitives of Level 1, such as assignment and conditional, can be implemented in machine language and run on the Von Neumann hardware of Level 2, conceptually tying together these two areas. It ends with a discussion of important new directions in hardware design—multicore processors and massively parallel machines.

By the end of Level 2 students have been introduced to basic concepts in logic design and computer organization, and they can appreciate the complexity inherent in these ideas. This complexity is the motivation for the material contained in Level 3 (Chapters 6–8), “The Virtual Machine.” This section describes how system software can create a more friendly,

user-oriented problem-solving environment that hides many of the ugly hardware details just described. Level 3 looks at the same problems discussed in Level 2, encoding and executing algorithms, but shows how this can be done much more easily in a virtual environment containing helpful tools like a graphical user interface, editors, translators, file systems, and debuggers. This section discusses the services and responsibilities of the operating system and how it has evolved. It investigates one of the most important virtual environments in current use—networks of computers. It shows how technologies such as Ethernet, the Internet, and the web link together independent systems via transmission media and communications software. This creates a virtual environment in which we seamlessly and transparently use not only the computer on our desk or in our hand but also computing devices located anywhere in the world. This transparency has progressed to the point where we can now use systems located “in the cloud” without regard for where they are, how they provide their services, and even whether they exist as real physical entities. Level 3 concludes with a look at one of the most important services provided by a virtual machine, namely information security, and describes algorithms for protecting the user and the system from accidental or malicious damage.

Once we have created this powerful user-oriented virtual environment, what do we want to do with it? Most likely we want to write programs to solve interesting problems. This is the motivation for Level 4 (Chapters 9–12), “The Software World.” Although this book should not be viewed as a programming text, it contains an overview of the features found in modern procedural programming languages. This gives students an appreciation for the interesting and challenging task of the computer programmer and the power of the problem-solving environment created by a modern high-level language. (Detailed introductions to five important high-level programming languages are available via online, downloadable chapters accessible through the CourseMate for this text, as well as at www.cengagebrain.com.) There are many different language models, so Level 4 also includes a discussion of other language types, including special-purpose languages such as SQL, HTML, and JavaScript, as well as the functional, logic, and parallel language paradigms. An introduction to the design and construction of a compiler shows how high-level languages can be translated into machine language for execution. This latter discussion ties together numerous ideas from earlier chapters, as we show how an algorithm (Level 1), expressed in a high-level language (Level 4), can be compiled and executed on a typical Von Neumann machine (Level 2) by using system software tools (Level 3). These “recurring themes” and frequent references to earlier concepts help reinforce the idea of computer science as an integrated set of topics. At the conclusion of Level 4, we introduce the idea of computability and unsolvability to show students that there are provable limits to what programs, computers, and computer science can achieve.

We now have a high-level programming environment in which it is possible to write programs to solve important problems. In Level 5 (Chapters 13–16), “Applications,” we take a look at a few important uses of computers in our modern society. There is no way to cover more than a fraction of

the many applications of computers and information technology in a single section. Indeed, there is hardly a field of study or an aspect of our daily lives that has not been impacted by advances in computation and communication. Instead, we focus on a small set of interesting applications that demonstrate important concepts, tools, and techniques of computer science. This includes applications drawn from the sciences and engineering (simulation and modeling), business and finance (ecommerce, databases), the social sciences (artificial intelligence), and everyday life (computer-generated imagery, video gaming, virtual communities). Our goal is to show students that these applications are not “magic boxes” whose inner workings are totally unfathomable. Rather, they are the direct result of building upon the core concepts of computer science presented in the previous chapters. We hope these discussions encourage readers to seek out information on applications specific to their own areas of interest.

Finally, we reach the highest level of study, Level 6 (Chapter 17), “Social Issues in Computing,” which addresses the social, ethical, and legal issues raised by pervasive computer technology. This section (written by contributing author Professor Bo Brinkman of Miami University) examines issues such as the theft of intellectual property, national security concerns aggravated by information technology, and the erosion of personal privacy caused by the popularity of social networks. This chapter does not attempt to provide easy solutions to these many-faceted problems. Instead, it focuses on techniques that students can use to think about ethical issues and reach their own conclusions. Our goal in this final section is to make students aware of the enormous impact that information technology is having on our society and to give them tools for making informed decisions.

This, then, is the hierarchical structure of our text. It begins with the algorithmic foundations of the discipline and works its way from lower-level hardware concepts through virtual machine environments, high-level languages, software, and applications, to the social issues raised by computer technology. This organizational structure, along with the use of recurring themes, enables students to view computer science as a unified and coherent field of study. The material in Chapters 1–12 is intended to be covered sequentially, but the applications discussed in Chapters 13–16 can be covered in any order and the social issues in Chapter 17 can be presented at any time.

What’s New in This Edition

This seventh edition of *Invitation to Computer Science* addresses a number of emerging issues in computer science. We have added significant new material on the Internet of Things, cloud computing, embedded computing, new models of electronic commerce, personal privacy, data mining, robots, and drones.

New and updated Special Interest Boxes highlight interesting historical vignettes, new developments in computing, biographies of important people in the field, and news items showing how computing is affecting our everyday lives. There are new in-chapter Practice Problems (with answers

provided at the end of the text) as well as new end-of-chapter Exercises. There are also new additions to the end-of-chapter Challenge Problems; these more complex questions can be used for longer assignments done either individually or by teams of students.

An Interactive Experience— CourseMate

This edition offers significantly enhanced supplementary material and additional resources available online through CourseMate. CourseMate is a valuable web resource containing an ebook with highlighting and note-taking capabilities, supplementary readings for each chapter, a glossary and flashcards of key technical terms, and links to interesting articles, helpful references, and relevant videos from across the web. The CourseMate encourages a truly interactive experience with study games, objective- and application-based quizzing, and hands-on exploration projects that speak students' language. Instructors may add CourseMate to the textbook package, or students may purchase CourseMate directly through www.cengagebrain.com.

An Experimental Science— Laboratory Software and Manual

Another important aspect of computer science education is the realization that, like physics, chemistry, and biology, computer science is an empirical, laboratory-based discipline in which learning comes not only from watching and listening but from doing and trying. Many ideas in computer science cannot be fully understood and appreciated until they are visualized, manipulated, and tested. Today, most computer science faculty see structured laboratories as an essential part of an introductory course. We concur, and this development is fully reflected in our approach to the material.

Associated with this text is a laboratory manual and custom-designed laboratory software that enables students to experiment with the concepts we present. The manual contains 20 laboratory experiences, closely coordinated with the main text, that cover all levels except Level 6. These labs give students the chance to observe, study, analyze, and/or modify an important idea or concept. For example, associated with Level 1 (the algorithmic foundations of computer science) are experiments that animate the algorithms in Chapters 2 and 3 and ask students to observe and discuss what is happening in these animations. There are also labs that allow students to measure the running time of these algorithms for different-sized data sets and discuss their behavior, thus providing concrete observations of an abstract concept like algorithmic efficiency. There are similar labs available for Levels 2, 3, 4, and 5 that highlight and clarify the material presented in the text.

Each of the lab manual experiments includes an explanation of how to use the software, a description of how to conduct the experiment, and problems for students to complete. For these lab projects, students can either work on their own or in teams, and the course may utilize either a closed-lab (formal, scheduled) or open-lab (informal, unscheduled) setting. The manual and software work well with all these laboratory models. The text contains “Laboratory Exercise” boxes that describe each lab and identify the point in the text where it would be most appropriate.

In this new seventh edition, the Laboratory Manual has been integrated into the CourseMate for this text. Instructors may add the CourseMate, including Lab Manual, to their textbook package; contact your sales representative for more information. Students may also purchase the CourseMate/Laboratory Manual directly through www.cengagebrain.com.

Programming and Online Language Modules

Programming concepts are presented in the text in the form of a survey of the features each high-level language provides and how they differ based on the computing tasks for which they were intended. Code examples are shown only to illustrate how algorithms can be embedded into the varying syntax of different languages. For instructors who want their students to have additional programming experience, online language modules for Ada, C++, C#, Java, and Python are available. Students may download any or all of these for free by going to the CourseMate for this text or to www.cengagebrain.com. At the CengageBrain home page, search for the ISBN of this book (found above the bar code on the back cover). At the Invitation page, click “Free Materials/Access Now”. These PDF documents can be read online, downloaded to the student’s computer, or printed and read on paper. Each chapter includes language-specific practice problems and exercises.

Computer science is a young and exciting discipline, and we hope that the new material in this edition, along with the laboratory projects and online modules, will convey this feeling of excitement to your students.

Instructor Resources

The following supplemental teaching tools are available when this book is used in a classroom setting. All supplements are available to instructors for download at sso.cengage.com.

Electronic Instructor’s Manual

The Instructor’s Manual follows the text chapter by chapter and includes material to assist in planning and organizing an effective, engaging course. The Instructor’s Manual includes Overviews, Chapter Objectives, Teaching

Tips, Quick Quizzes, Class Discussion Topics, Additional Projects, Additional Resources, and Key Terms. A sample syllabus is also available.

Solutions

Complete solutions to chapter exercises are provided.

Test Bank

Cengage Learning Powered by Cognero® is a flexible, online system that allows you to:

- Author, edit, and manage test bank content from multiple Cengage Learning solutions
- Create multiple test versions in an instant
- Deliver tests from your Learning Management System (LMS), your classroom, or anywhere you want

PowerPoint Presentations

Microsoft PowerPoint slides to accompany each chapter are available. Slides may be used to guide classroom presentation or to print as classroom handouts, or they may be made available to students for chapter review. Instructors may customize the slides to best suit their course with the complete Figure Files from the text.

Acknowledgments

The authors would like to thank Alyssa Pratt and Deb Kaufmann for their invaluable assistance in developing this new edition, as well as the reviewers for this edition, whose comments were very helpful.

Colt Mazeau, *American National University*
William Oblitey, *Indiana University of Pennsylvania*
Gary Smith, *Elmhurst College*
Michael Verdicchio, *The Citadel*

Any errors, of course, are the fault solely of the authors.

—G. Michael Schneider
Macalester College
schneider@macalester.edu

—Judith L. Gersting
Indiana University-Purdue University at Indianapolis
gersting@iupui.edu