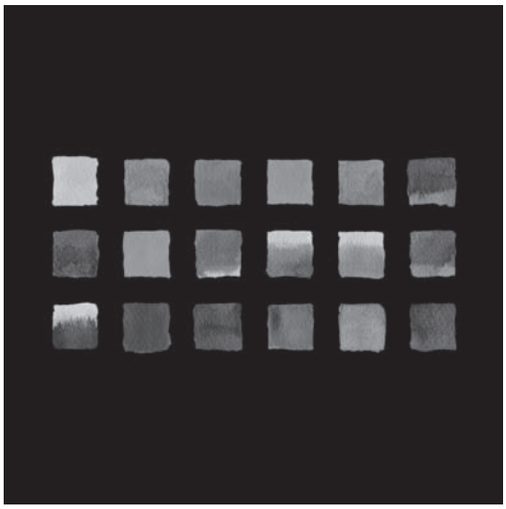
Data Structures and Abstractions with Java™

Fifth Edition



Frank M. Carrano
University of Rhode Island
Timothy M. Henry
New England Institute of Technology

Pearson

330 Hudson Street, NY NY 10013

Senior Vice President Courseware Portfolio Management: Marcia J. Horton
Director, Portfolio Management: Engineering, Computer Science & Global Editions: Julian Partridge
Executive Portfolio Manager: Tracy Johnson
Portfolio Management Assistant: Meghan Jacoby
Managing Content Producer: Scott Disanno
Content Producer: Lora Friedenthal
Rights and Permissions Manager: Ben Ferrini
Manufacturing Buyer, Higher Ed, Lake Side Communications Inc (LSC): Maura Zaldivar-Garcia
Inventory Manager: Bruce Bounty
Product Marketing Manager: Yvonne Vannatta
Field Marketing Manager: Demetrius Hall
Procurement Specialist: Maura Zaldivar-Garcia
Marketing Assistant: Jon Bryant
Cover Designer: Black Horse Designs
Cover Photo: Mustafahacalaki/E+/Getty Images
Printer/Binder: LSC Communications, Inc.
Full-Service Project Management: Billu Suresh, SPi Global

Credits and acknowledgments borrowed from other sources and reproduced, with permission, in this textbook appear on the appropriate page within text.

Library of Congress Cataloging-in-Publication Data

Names: Carrano, Frank M., author. | Henry, Timothy M., author.
Title: Data structures and abstractions with Java / Frank M. Carrano, University of Rhode Island, Timothy M. Henry, New England Institute of Technology.
Description: Fifth edition. | Boston : Pearson Education, Inc., 2018. | Includes index.
Identifiers: LCCN 2018000065 | ISBN 9780134831695 (alk. paper) | ISBN 0134831691 (alk. paper)
Subjects: LCSH: Data structures (Computer science) | Java (Computer program language)
Classification: LCC QA76.9.D33 C37 2018 | DDC 005.7/3--dc23 LC record available at https://lccn.loc.gov/2018000065

10 9 8 7 6 5 4 3 2 1

Pearson

ISBN 10: 0-13-483169-1
ISBN 13: 978-0-13-483169-5

Welcome to the fifth edition of *Data Structures and Abstractions with Java*, a book for an introductory course in data structures, typically known as CS-2.

I wrote this book with you in mind—whether you are an instructor or a student—based upon my experiences during more than three decades of teaching undergraduate computer science. I wanted my book to be reader friendly so that students could learn more easily and instructors could teach more effectively. To this end, you will find the material covered in small pieces—I call them "segments"—that are easy to digest and facilitate learning. Numerous examples that mimic real-world situations provide a context for the new material and help to make it easier for students to learn and retain abstract concepts. Many illustrations clarify complicated ideas. Included are over 60 video tutorials to supplement the instruction and help students when their instructor is unavailable.

I am happy to work again with my colleague and co-author of the fourth edition, Dr. Timothy Henry. Together we have continued to enhance the presentation with our focus on design decisions for both specifications and implementations of various data structures, as well as our emphasis on safe and secure programming practices.

We hope that you enjoy reading this book. Like many others before you, you can learn—or teach —data structures in an effective and sustainable way.

Warm regards,
Frank M. Carrano

Using This Flexible and Unique Textbook

This book's organization, sequencing, and pace of topic coverage make teaching and learning easier by

- Focusing the reader's attention on one concept at a time
- Providing flexibility in the order in which you can cover topics
- Clearly distinguishing between the specification and implementation of abstract data types (ADTs)
- Separating the relevant coverage of Java into Java Interludes, which you can use as needed

To accomplish this approach, we have organized the material into 30 chapters, composed of small, numbered segments that deal with one concept at a time. Each chapter focuses on either the specification and use of an ADT or its various implementations. You can choose to cover the specification of an ADT followed by its implementations, or you can treat the specification and use of several ADTs before you consider any implementation issues. The book's organization makes it easy for you to choose the topic order that you prefer.

Our use of Java Interludes to treat the pertinent aspects of Java clearly separate our coverage of data structures from Java-specific issues. These interludes occur between chapters throughout the book as needed. Our focus, however, is on data structures not Java. You can see the titles of these interludes, as well as their placement between chapters, in the table of contents that follows.

Table of Contents at a Glance

The following brief table of contents shows the overall composition of the book. Notice the Introduction, Prelude, and nine Java Interludes. Further details—including a chapter-by-chapter description—are given later in this preface.

What's New?

This new edition of *Data Structures and Abstractions with Java* enhances the previous edition and continues its pedagogical approach to make the material accessible to students at the introductory level. The coverage that you enjoyed in previous editions is still here. As is usual for us, we have read every word of the previous edition and made changes to improve clarity and correctness. No chapter or interlude appears exactly as it did before. Our changes are motivated by reader suggestions and our own desire to improve the presentation.

In this new edition, we

- Added coverage of recursion in a new chapter that introduces grammars, languages, and backtracking.
- Continued our introduction to safe and secure programing practices.
- Added additional Design Decisions, Notes, Security Notes, and Programming Tips throughout the book.
- Added new exercises and programming projects, with an emphasis in areas of gaming, e-commerce, and finance to most chapters.
- Adjusted the order of some topics.
- Refined our terminology, presentation, and word choices to ease understanding.
- Revised the illustrations to make them easier to read and to understand.
- Renamed Self-Test Questions as Study Questions and moved their answers to online. We encourage our students to discuss their own answers with a study partner or group.
- Included the appendix about Java classes within the book instead of leaving it online.
- Reduced the amount of Java code given in the book.
- Ensured that all Java code is Java 9 compliant.

Connect with Us

We are always available to instructors and students who use our books. Your comments, suggestions, and corrections will be greatly appreciated. Please e-mail us at

`carrano@acm.org` or `timhenry@acm.org`

You can also find us on Twitter, our websites, or Facebook:

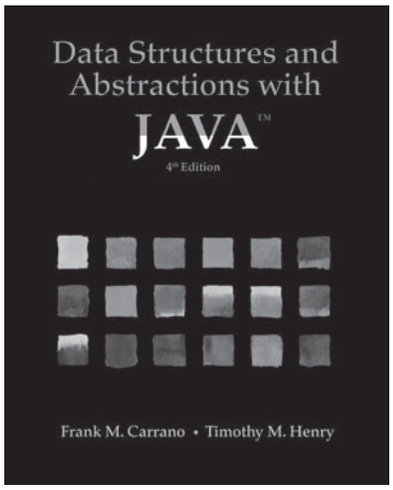
- Twitter: `twitter.com/makingCSreal`
- Websites: `www.makingCSreal.com` and `timothyhenry.net`
- Facebook: `www.facebook.com/makingCSreal`

A Note to Students

The topics that we cover in this book deal with the various ways of organizing data so that a given application can access and manipulate data in an efficient way. These topics are fundamental to your future study of computer science, as they provide you with the foundation of knowledge required to create complex and reliable software. Whether you are interested in designing video games or software for robotic controlled surgery, the study of data structures is vital to your success. Even if you do not study all of the topics in this book now, you are likely to encounter them later. We hope that you will enjoy reading the book, and that it will serve as a useful reference tool for your future courses.

After looking over this preface, you should read the Introduction. There you will quickly see what this book is about and what you need to know about Java before you begin. The Prelude discusses class design and the use of Java interfaces. We use interfaces throughout the book. Appendixes A, B, and C review `javadoc` comments, Java classes, and inheritance. Java Interludes occur throughout the book and cover advanced aspects of Java as they are needed. Supplemental material is available online to review the basics of Java and its file input and output, to define important terms, and to answer the study questions. Note that inside the front and back covers you will find Java's reserved words, its primitive data types, the precedence of its operators, and a list of Unicode characters.

Please be sure to browse the rest of this preface to see the features that will help you in your studies.



Features to Enhance Learning

Each chapter begins with a table of contents, a list of prerequisite portions of the book that you should have read, and the learning objectives for the material to be covered. Other pedagogical elements appear throughout the book, as follows:

Notes Important ideas are presented or summarized in highlighted paragraphs and are meant to be read in line with the surrounding text.

Security Notes Aspects of safe and secure programming are introduced and highlighted in this new feature.

A Problem Solved Large examples are presented in the form of "A Problem Solved," in which a problem is posed and its solution is discussed, designed, and implemented.

Design Decisions To give readers insight into the design choices that one could make when formulating a solution, "Design Decision" elements lay out such options, along with the rationale behind the choice made for a particular example. These discussions are often in the context of one of the "A Problem Solved" examples.

Examples Numerous examples illuminate new concepts.

Programming Tips Suggestions to improve or facilitate programming are presented as soon as they become relevant.

Study Questions Questions are posed throughout each chapter, integrated within the text, that reinforce the concept just presented. These "study" questions help readers to understand the material, since answering them requires pause and reflection. We suggest that you discuss these questions and their answers with others before consulting our solutions, which are available to you online.

VideoNotes Online tutorials are a Pearson feature that provides video support to the presentation given throughout the book. They offer students another way to recap and reinforce key concepts. VideoNotes allow for self-paced instruction with easy navigation, including the ability to select, play, rewind, fast-forward, and stop within each video. Unique VideoNote icons appear throughout this book whenever a video is available for a particular concept or problem. A detailed list of the VideoNotes for this text and their associated locations in the book can be found on page xxiv of this preface. VideoNotes are free with the purchase of a new textbook. To purchase access to VideoNotes, please go to

`pearsonhighered.com/carrano`

Exercises and Programming Projects Further practice is available by solving the exercises and programming projects at the end of each chapter. Unfortunately, we cannot give readers the answers to these exercises and programming projects, even if they are not enrolled in a class. Only instructors who adopt the book can receive selected answers from the publisher. For help with these exercises and projects, you will have to contact your instructor.

Accessing Instructor and Student Resource Materials

The following items are available on the publisher's website at pearsonhighered.com/carrano:

- Java code as it appears in the book
- A link to any misprints that have been discovered since the book was published
- Links to additional online content, which is described next

Instructor Resources

The following protected material is available to instructors who adopt this book by logging onto Pearson's Instructor Resource Center (IRC), accessible from pearsonhighered.com/carrano:

- Instructor's Guide
- PowerPoint lecture slides that have ADA compatible descriptive text for all images
- Instructor solutions manual
- Lab manual and solutions
- Java source code for instructors
- Figures from the book
- Test bank

Additionally, the resources available to students and described next, are available in the IRC to instructors.

Please contact your Pearson sales representative for an instructor access code. Contact information is available at `pearsonhighered.com/replocator`.

Student Resources

The following material is available to students by logging onto the Companion Website accessible from pearsonhighered.com/carrano:

- Instructional VideoNotes
- A survey of basic Java (Supplement 1)
- An overview of file I/O (Supplement 2)
- A glossary of terms (Supplement 3)
- Answers to Study Questions (Supplement 4)

Students must use the access card located in the front of the book to register for and then enter the Companion Website. Students without an access code can purchase access from the Companion Website by following the instructions listed there.

Note that documentation for the Java Class Library is available at `docs.oracle.com/javase/9/docs/api/`.

Content Overview

Readers of this book should have completed a programming course, preferably in Java. The appendixes and online supplements cover the essentials of Java that we assume readers will know. You can use this material as a review or as the basis for making the transition to Java from another programming language.

- **Introduction:** We begin by setting the stage for the data organizations that we will study by looking at some everyday examples.
- **Prelude:** The Prelude discusses the design of classes. Among the topics we cover are preconditions, postconditions, assertions, interfaces, and the Unified Modeling language (UML). Design is an important aspect of our presentation.
- **Chapters 1 through 3:** We introduce the bag as an abstract data type (ADT). By dividing the material across several chapters, we clearly separate the specification, use, and implementation of the bag. For example, Chapter 1 specifies the bag and provides several examples of its use. This chapter also introduces the ADT set. Chapter 2 covers implementations that use arrays, while Chapter 3 introduces chains of linked nodes and uses one in the definition of a class of bags.

  In a similar fashion, we separate specification from implementation throughout the book when we discuss various other ADTs. You can choose to cover the chapters that specify and use the ADTs and then later cover the chapters that implement them. Or you can cover the chapters as they appear, implementing each ADT right after studying its specification and use. A list of chapter prerequisites appears later in this preface to help you plan your path through the book.

  Chapter 2 does more than simply implement the ADT bag. It shows how to approach the implementation of a class by initially focusing on core methods. When defining a class, it is often useful to implement and test these core methods first and to leave definitions of the other methods for later. Chapter 2 also introduces the concept of safe and secure programming, and shows how to add this protection to your code.
- **Java Interludes 1 and 2:** The first Java interlude introduces generics, so that we can use it with our first ADT, the bag. This interlude immediately follows Chapter 1. Java Interlude 2 introduces exceptions and follows Chapter 2. We apply this material to the implementations of the ADT bag.
- **Chapter 4:** Here we introduce the efficiency and complexity of algorithms, a topic that we integrate into future chapters.
- **Chapters 5 and 6:** Chapter 5 discusses stacks, giving examples of their use, and Chapter 6 implements the stack using an array, a chain of linked nodes, and a vector.
- **Java Interlude 3:** This Java interlude shows how the programmer can write new exception classes. In doing so, it shows how to extend an existing class of exceptions. It also introduces the `finally` block.
- **Chapters 7 and 8:** Chapter 7 discusses queues, deques, and priority queues, and Chapter 8 considers their implementations. It is in this latter chapter that we introduce circularly linked and doubly linked chains. Chapter 8 also uses the programmer-defined class `EmptyQueueException`.
- **Chapter 9:** Next, we present recursion as a problem-solving tool and its relationship to stacks. Recursion, along with algorithm efficiency, is a topic that is revisited throughout the book.
- **Chapters 10, 11, and 12:** The next three chapters introduce the ADT list. We discuss this collection abstractly and then implement it by using an array and a chain of linked nodes.

- **Java Interlude 4 and Chapter 13:** Included in our coverage of Java iterators are the standard interfaces `Iterator`, `Iterable`, and `ListIterator`. Chapter 13 then shows ways to implement an iterator for the ADT list. It considers and implements Java's iterator interfaces `Iterator` and `ListIterator`.
- **Chapter 14:** This new chapter offers more coverage of recursion, including languages, grammars, and backtracking.
- **Java Interlude 5:** This interlude provides the Java concepts needed for the sorting methods that we are about to present. It introduces the standard interface `Comparable`, generic methods, bounded type parameters, and wildcards.
- **Chapters 15 and 16:** The next two chapters discuss various sorting techniques and their relative complexities. We consider both iterative and recursive versions of these algorithms.
- **Java Interlude 6:** This interlude discusses mutable and immutable objects, a topic relevant to the preceding chapters about sorting and the following one about sorted lists.
- **Chapters 17 and 18 and Java Interlude 7:** Continuing the discussion of a list, Chapter 17 introduces the sorted list, looking at two possible implementations and their efficiencies. Chapter 18 shows how to use the list as a superclass for the sorted list and discusses the general design of a superclass. Although inheritance is reviewed in Appendix C, the relevant particulars of inheritance—including protected access, abstract classes, and abstract methods—are presented in Java Interlude 7 just before Chapter 18.
- **Chapter 19:** We then examine some strategies for searching an array or a chain in the context of a list or a sorted list. This discussion is a good basis for the sequence of chapters that follows.
- **Java Interlude 8:** Before we get to the next chapter, we quickly cover in this interlude situations where more than one generic data type is necessary.
- **Chapters 20 through 23:** Chapter 20 covers the specification and use of the ADT dictionary. Next, Chapter 21 presents implementations of the dictionary that are linked or that use arrays. Chapter 22 introduces hashing, and Chapter 23 uses hashing as a dictionary implementation.
- **Chapters 24 and 25:** Chapter 24 discusses trees and their possible uses. Included among the several examples of trees is an introduction to the binary search tree and the heap. Chapter 25 considers implementations of the binary tree and the general tree.
- **Java Interlude 9:** Java Interlude 9 discusses cloning. We clone an array, a chain of linked nodes, and a binary node. We also investigate a sorted list of clones. Although this material is important, you can treat it as optional, as it is not required in the following chapters.
- **Chapters 26 through 28:** Chapter 26 focuses on the implementation of the binary search tree. Chapter 27 shows how to use an array to implement the heap. Chapter 28 introduces balanced search trees. Included in this chapter are the AVL, 2-3, 2-4, and red-black trees, as well as B-trees.
- **Chapters 29 and 30:** Finally, we discuss graphs and look at several applications and two implementations.
- **Appendixes A through C:** The appendixes provide supplemental coverage of Java. As we mentioned earlier, Appendix A considers programming style and comments. It introduces `javadoc` comments and defines the tags that we use in this book. Appendix B discusses Java classes, and Appendix C expands this topic by looking at composition and inheritance.

Acknowledgments

Our sincere appreciation and thanks go to the following reviewers for carefully reading the previous edition and making candid comments and suggestions that greatly improved the work:

Prakash Duraisamy—*Miami University*
David Gries—*Cornell University*
Hakam Alomari—*Miami University*
Jack Pope—*North Hennepin Community College*
Anna Rafferty—*Carleton College*

Special thanks go to our support team at Pearson Education Computer Science during the lengthy process of revising this book: Executive Portfolio Manager Tracy Johnson, Portfolio Management Assistant Meghan Jacoby, Managing Content Producer Scott Disanno, and Content Producer Lora Friedenthal have always be a great help to us in completing our projects. Our long-time copy editor, Rebecca Pepper, ensured that the presentation is clear, correct, and grammatical. Thank you so much!

Our gratitude for the previously mentioned people does not diminish our appreciation for the help provided by many others. Tim Henry produced the VideoNotes and the lecture slides for this edition. Professor Charles Hoot of the Oklahoma City University created the lab manual, Professor Kathy Liszka from the University of Akron created the collection of test questions, and Joe Erickson and Jesse Grabowski provided the solutions to many of the programming projects. Thank you again to the reviewers of the previous editions of the book:

Reviewers for the fourth edition:

Tony Allevato—*Virginia Polytechnic Institute and State University*
Mary Boelk—*Marquette University*
Suzanne Buchele—*Southwestern University*
Kevin Buffardi—*Virginia Polytechnic Institute and State University*
Jose Cordova—*University of Louisiana at Monroe*
Greg Gagne—*Westminster College*
Victoria Hilford—*University of Houston*
Jim Huggins—*Kettering University*
Shamim Kahn—*Columbus State University*
Kathy Liszka—*University of Akron*
Eli Tilevich—*Virginia Polytechnic Institute and State University*
Jianhua Yang—*Columbus State University*
Michelle Zhu—*Southern Illinois University*

Reviewers for the third edition:

Steven Andrianoff—*St. Bonaventure University*
Brent Baas—*LeTourneau University*
Timothy Henry—*New England Institute of Technology*
Ken Martin—*University of North Florida*
Bill Siever—*Northwest Missouri State University*
Lydia Sinapova—*Simpson College*
Lubomir Stanchev—*Indiana University*
Judy Walters—*North Central College*
Xiaohui Yuan—*University of North Texas*

Acknowledgments

Reviewers for the second edition:

Harold Anderson—*Marist College*
Razvan Andonie—*Central Washington University*
Tom Blough—*Rensselaer Polytechnic Institute*
Chris Brooks—*University of San Francisco*
Adrienne Decker—*University at Buffalo, SUNY*
Henry Etlinger—*Rochester Institute of Technology*
Derek Harter—*Texas A&M University*
Timothy Henry—*New England Institute of Technology*
Robert Holloway—*University of Wisconsin, Madison*
Charles Hoot—*Oklahoma City University*
Teresa Leyk—*Texas A&M University*
Robert McGlinn—*Southern Illinois University, Carbondale*
Edward Medvid—*Marymount University*
Charles Metzler—*City College of San Francisco*
Daniel Zeng—*University of Arizona*

Reviewers for the first edition:

David Boyd—*Valdosta State University*
Dennis Brylow—*Purdue University*
Michael Croswell—*Industry trainer/consultant*
Matthew Dickerson—*Middlebury College*
Robert Holloway—*University of Wisconsin, Madison*
John Motil—*California State University, Northridge*
Bina Ramamurthy—*University at Buffalo, SUNY*
David Surma—*Valparaiso University*

We continue to appreciate the many others who helped during previous editions. They include Alan Apt, Steve Armstrong, James Blanding, Lianne Dunn, Bob Englehardt, Mike Giacobbe, Toni Holm, Charles Hoot, Brian Jepson, Rose Kernan, Christianna Lee, Patrick Lindner, John Lovell, Vince O'Brien, Patty Roy, Walt Savitch, Ben Schomp, Heather Scott, Carole Snyder, Chirag Thakkar, Camille Trentacoste, Nate Walker, and Xiaohong Zhu.

Finally, we thank our families and friends—Doug, Joanne, Tita, Bobby, Ted, Nancy, Sue, Tom, Bob, Maybeth, Marge, and Lorraine—for giving us lives away from computers.

Thank you, everyone, for your expertise and good cheer.

Frank M. Carrano
Timothy M. Henry

Contents

Table of Contents

Table of Contents

VideoNotes Directory

This table lists the VideoNotes that are available online. The page numbers indicate where in the book each VideoNote has relevance.

VideoNotes

VideoNotes

Chapter Prerequisites

Each chapter and appendix assumes that the reader has studied certain previous material. This list indicates those prerequisites. Numbers represent chapter numbers, letters reference appendixes, and "JI" precedes each interlude number. You can use this information to plan a path through the book.

		Prerequisites
Prelude	Designing Classes	Supplement 1, A, B, C
Chapter 1	Bags	Prelude, C
Java Interlude 1	Generics	Prelude
Chapter 2	Bag Implementations That Use Arrays	Prelude, 1
Java Interlude 2	Exceptions	Supplement 1, B, C
Chapter 3	A Bag Implementation That Links Data	1, 2, JI2
Chapter 4	The Efficiency of Algorithms	B, 2, 3
Chapter 5	Stacks	Prelude, 1, JI2
Chapter 6	Stack Implementations	2, 3, 4, 5
Java Interlude 3	More About Exceptions	C, JI2
Chapter 7	Queues, Deques, and Priority Queues	Prelude, 5
Chapter 8	Queue, Deque, and Priority Queue Implementations	2, 3, 6, 7
Chapter 9	Recursion	B, 2, 3, 4, 5
Chapter 10	Lists	Introduction, B, Prelude, JI2, 6
Chapter 11	A List Implementation That Uses an Array	Prelude, 2, 4, 10
Chapter 12	A List Implementation That Links Data	3, 8, 10, 11
Java Interlude 4	Iterators	JI2, 10
Chapter 13	Iterators for the ADT List	11, 12, JI4
Chapter 14	Problem Solving with Recursion	5, 9
Java Interlude 5	More About Generics	JI1
Chapter 15	An Introduction to Sorting	3, 4, 9, JI5
Chapter 16	Faster Sorting Methods	4, 9, JI5, 15
Java Interlude 6	Mutable and Immutable Objects	C, 10
Chapter 17	Sorted Lists	4, 9, 10, 12, JI6
Java Interlude 7	Inheritance and Polymorphism	Prelude, C, 6
Chapter 18	Inheritance and Lists	C, 10, 11, 12, 17, JI7
Chapter 19	Searching	4, 9, 10, 11, 12, 17
Java Interlude 8	Generics Once Again	B, JI5
Chapter 20	Dictionaries	10, JI4, 13, 19, JI8
Chapter 21	Dictionary Implementations	3, 4, 10, 11, 12, JI4, 19, 20

		Prerequisites
Chapter 22	Introducing Hashing	20, 21
Chapter 23	Hashing as a Dictionary Implementation	4, 11, 12, JI4, 20, 21, 22
Chapter 24	Trees	5, 9, 12, JI4, 19
Chapter 25	Tree Implementations	C, JI2, 5, 7, 12, 24
Java Interlude 9	Cloning	JI5, JI6, 17, 25, B, C
Chapter 26	A Binary Search Tree Implementation	C, 9, 20, 24, 25
Chapter 27	A Heap Implementation	2, 11, 24
Chapter 28	Balanced Search Trees	24, 25, 26
Chapter 29	Graphs	5, 7, 24
Chapter 30	Graph Implementations	5, 7, 10, JI4, 13, 20, 24, 29
Appendix A	Documentation and Programming Style	Some knowledge of Java
Appendix B	Java Classes	Supplement 1
Appendix C	Creating Classes from Other Classes	B
Supplement 1	Java Basics	Knowledge of a programming language
Supplement 2	File Input and Output	Prelude, JI2, Supplement 1