



Object-Oriented Data Structures Using Java, Fourth Edition Transition Guide

By Nell Dale • Daniel T. Joyce • Chip Weems
ISBN-13: 9781284089097
Hardcover with Navigate 2 Advantage Access
820 Pages • ©2018

Main Updates

- **NEW** - Includes chapters on the Map ADT and the Collection ADT
- **NEW** - Sections highlighting variations on the standard data structures, including a look at how the structures are supported by the Java Standard Library
- **NEW** - New sections and examples introduce important topics such as image generation, recursive processing of arrays and linked lists, fractals, games, and text analysis
- **NEW** – Modern new design enhances the look and feel of the text
- **UPDATED** - Current and clarified exposition throughout
- **STUDENT FAVORITE** - Numerous exercises throughout allow students to experience a variety of applications on the concepts within the chapter, which vary in level of difficulty

Major Chapter Changes Are Listed Below

Chapter 1: Getting Organized

- Chapter 1 still introduces classes, objects, applications, inheritance, packages, data structures, structuring mechanisms (the array and the reference), and complexity analysis
- The two short sections about software engineering and object orientation (previously Sections 1.1 and 1.2) were dropped
- A new two page sub-section addressing inheritance-based polymorphism was added to Section 1.2 Organizing Classes (previously 1.4)
- Section 1.3 Exceptional Situations (previously 3.3) was moved here from Chapter 3 and now uses the Date class within its examples
- Section 1.5 Basic Structuring Mechanisms (previously 1.6) was supplemented to contain two pages of information about direct and indirect addressing, plus an example related to two dimensional arrays involving image generation
- Section 1.6 Comparing Algorithms: Order of Growth Analysis (previously 1.7), while following the same basic pattern as in the third edition, now introduces the basic concepts with examples involving a guessing game and includes a new example algorithm - selection sort

Chapter 2: The Stack ADT

- This chapter merges the previous Chapters 2 (Abstract Data Types) and 3 (The Stack ADT)
- We dropped the StringLog ADT example which formed the basis of the previous Chapter 2 and in its place use the Stack ADT to introduce the ideas of data abstraction and the three perspectives of ADTs
- As in the previous Chapter 3 we provide a formal specification of the stack using generics, array- and link-based implementations, and example applications – balanced expressions and postfix expression evaluation
- We now define a single stack interface, rather than stack, bounded stack, and unbounded stack interfaces – note that inheritance of interfaces is now covered in the new Section 4.7 Queue Variations
- A new two page description of interface-based polymorphism was added to the section dealing with Java interfaces
- The previous Section 2.4 Software Testing was dropped – many of the concepts covered there are now found in a feature "Testing ADTs" within Section 2.5 Array-Based Stack Implementations
- The new Section 2.10 Stack Variations discusses alternate definitions for stack operations and introduces the Java Library Stack Class and the Collections Framework

Chapter 3: Recursion

- This chapter contains all of the previous Chapter 4 Recursion except we dropped the previous Section 4.4 Counting Blobs
- New sections of examples using recursion were added:
 - Section 3.3 Recursive Processing of Arrays featuring binary search
 - Section 3.4 Recursive Processing of Linked Lists featuring traversals and transformations
 - Section 3.6 Fractals featuring a T-Square fractal

Chapter 4: The Queue ADT

- This chapter contains all of the previous Chapter 5 The Queue ADT except we dropped the previous Section 5.5 Application: The Card Game of War
- We now define a single queue interface, rather than queue, bounded queue, and unbounded queue interfaces
- Section 4.3 Array-Based Queue Implementations now includes a short discussion of amortized analysis
- We added the new Section 4.4 An Interactive Test Driver providing an example of the use of the ArrayBoundedQueue class
- The new Section 4.7 Queue Variations discusses alternate ways of handling exceptional situations, defines the Glass queue (previously defined within the average waiting time case study) and Double-Ended queues, introduces doubly-linked lists (previously introduced in Section 7.2 Double-Linked Lists) and discusses Java Library queue support

Chapter 5: The Collection ADT

- This new chapter presents a basic Collection ADT and uses a great deal of material from the previous Chapter 6 The List ADT
- This chapter uses the some basic implementation approaches used in the previous Chapter 6 The List ADT, including the approaches to implementing the array-based collection (new Section 5.2 previously a sub-section of 6.4), sorted array-based collection (new Section 5.5 previously a subsection of 6.4), and linked collection (new Section 5.6 previously a subsection of 6.7)
- Section 5.3 Application: Vocabulary Density is new and is used several times moving forward in the text, for example it is used in Section 5.5 to demonstrate efficiency gains afforded by the sorted array- based implementation
- Section 5.4 Comparing Objects Revisited was previously Section 6.1 with additional examples related to a FamousPerson class added to help convey the idea of the "key" of an element
- The new Section 5.7 Collection Variations discusses the Java Library Collection interface plus introduces two new

ADTs – the Bag and the Set.

Chapter 6: The List ADT

- This chapter is closely related to the previous Chapter 6 of the same name – however there are several important changes due to the relationship of lists to the collections defined in the new Chapter 5 and to a change in our approach to iteration
- There is only a single list interface defined – it extends both the CollectionInterface from the new Chapter 5 and the Library's Iterable interface, plus defines index related operations
- In previous editions we defined our own iteration related operations "reset" and "getNext" – in this edition we use an approach similar to that used by the library – this new approach is described in detail in Section 6.2 List Implementations
- The new Section 6.3 Applications: Card Deck and Games provides an implementation of a deck of cards using a list and sample applications that use the deck of cards (one of which is the previous example "Poker" from Edition 3 Section 6.5)
- Section 6.4 Sorted Array-Based List Implementation now
 - includes a discussion about Insertion Sort
 - introduces the concept of unsupported operations
 - introduces the Comparator interface (previously covered in Edition 3 Section 10.4 More Sorting Considerations)
- The new Section 6.5 List Variations discusses Java Library lists plus variations of linked lists (most of this material is an abridged version of material previously found in Edition 3 Chapter 7 More Lists – that chapter was dropped)

Chapter 7: The Binary Search Tree ADT

- This chapter is directly related to the previous Chapter 8 Binary Search Trees except
 - we added breadth first and depth first traversals of general trees
 - the interface now extends the newly defined CollectionInterface of Chapter 5 with just a few simple operations
 - we explain the idea of a snapshot of an ADT in relation to the tree traversals
 - we include a constructor that allows the client to pass a Comparator, as introduced in Section 6.4
 - we return to the text analysis experiment introduced in Section 5.3 as part of our performance analysis subsection
 - the previous Section 8.9 A Nonlinked representation of Binary Trees was moved to Chapter 9
- The new Section 7.10 Tree Variations describes many different variations of tree structures

Chapter 8: The Map ADT

- This new chapter presents The Map ADT also known as a symbol table, dictionary, or associative array. Two implementations are developed, one that uses an ArrayList and the other that uses a hash table
- A large part of the chapter is devoted to the concept of hashing – most of this material is from Edition 3, Section 10.6 Hashing, although the exposition has been expanded and clarified
- The Variations section discusses a map-based hybrid data structure plus Java's support for hashing

Chapter 9: The Priority Queue ADT

- This chapter contains what was previously (in Edition 3):
 - the first part of Section 9.1 Priority Queues which is now Section 9.1 The Priority Queue Interface
 - the second part of Section 9.1 Priority Queues which is now Section 9.2 Priority Queue Implementations and includes an actual array-based implementation
 - the first part of Section 9.2 Heaps which is now Section 9.3 The Heap

- the second part of Section 9.2 Heaps which is now Section 9.4 The Heap Implementation – Section 9.4 also includes, as mentioned previously, Edition 3 Section 8.9 A Nonlinked Representation of Binary Trees

Chapter 10: The Graph ADT

- This chapter contains the material from Edition 3, Sections 9.3 through 9.6 (with the material from Edition 3, Section 9.6 split between two Sections (10.4 and 10.5))

Chapter 11: Sorting and Searching Algorithms

- This chapter contains Edition 3 Chapter 10, except the material from section 6 of that chapter (Hashing) was moved to the new Chapter 8

Table of Contents Comparison Outlines Chapter Reorganization

Object-Oriented Data Structures Using Java, Third Edition	Object-Oriented Data Structures Using Java, Fourth Edition
Chapter 1 Getting Organized	Chapter 1 Getting Organized
Chapter 2 Abstract Data Types	Chapter 2 The Stack ADT
Chapter 3 The Stack ADT	Chapter 3 Recursion
Chapter 4 Recursion	Chapter 4 The Queue ADT
Chapter 5 The Queue ADT	Chapter 5 The Collection ADT
Chapter 6 The List ADT	Chapter 6 The List ADT
Chapter 7 More Lists	Chapter 7 The Binary Search Tree ADT
Chapter 8 Binary Search Trees	Chapter 8 The Map ADT
Chapter 9 Priority Queues, Heaps, and Graphs	Chapter 9 The Priority Queue ADT
Chapter 10 Sorting and Searching Algorithms	Chapter 10 The Graph ADT
Appendix A Java Reserved Words	Chapter 11 Sorting and Searching Algorithms
Appendix B Operator Precedence	Appendix A Java Reserved Words
Appendix C Primitive Data Types	Appendix B Operator Precedence
Appendix D ASCII Subset of Unicode	Appendix C Primitive Data Types
Appendix E Application of Programmer Interfaces for the Java Classes and Interfaces Used in This Book	Appendix D ASCII Subset of Unicode