

2004

# Data Structures with Java

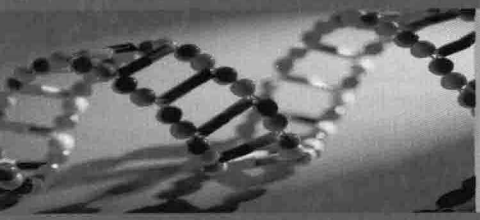
John R. Hubbard
*University of Richmond*, jhubbard@richmond.edu

Anita Huray
*University of Richmond*, ahubbard@richmond.edu

# Data Structures with Java™

*John R. Hubbard*
*Anita Huray*
**University of Richmond**


Upper Saddle River, NJ 07458



*Algorithms + Data Structures = Programs*

— Niklaus Wirth, 1976

# *Preface*

This book is intended to be used as a textbook for a university course in data structures, the standard CS2 course offered in American universities. It assumes that the student has completed an elementary course in computer programming with Java. A summary of Java fundamentals is provided in Appendix B for students who need to review the language.

The book covers all the classical data structures topics: basic concepts in Chapters 1–3, linear data structures (stacks, queues, lists, and tables) in Chapters 4–9, nonlinear data structures (trees and graphs) in Chapters 10–14 and 16, and a substantial treatment of ten sorting algorithms in Chapter 15. It is flexible enough to be used either for a one-semester course in data structures or for a two-semester course in data structures and algorithms.

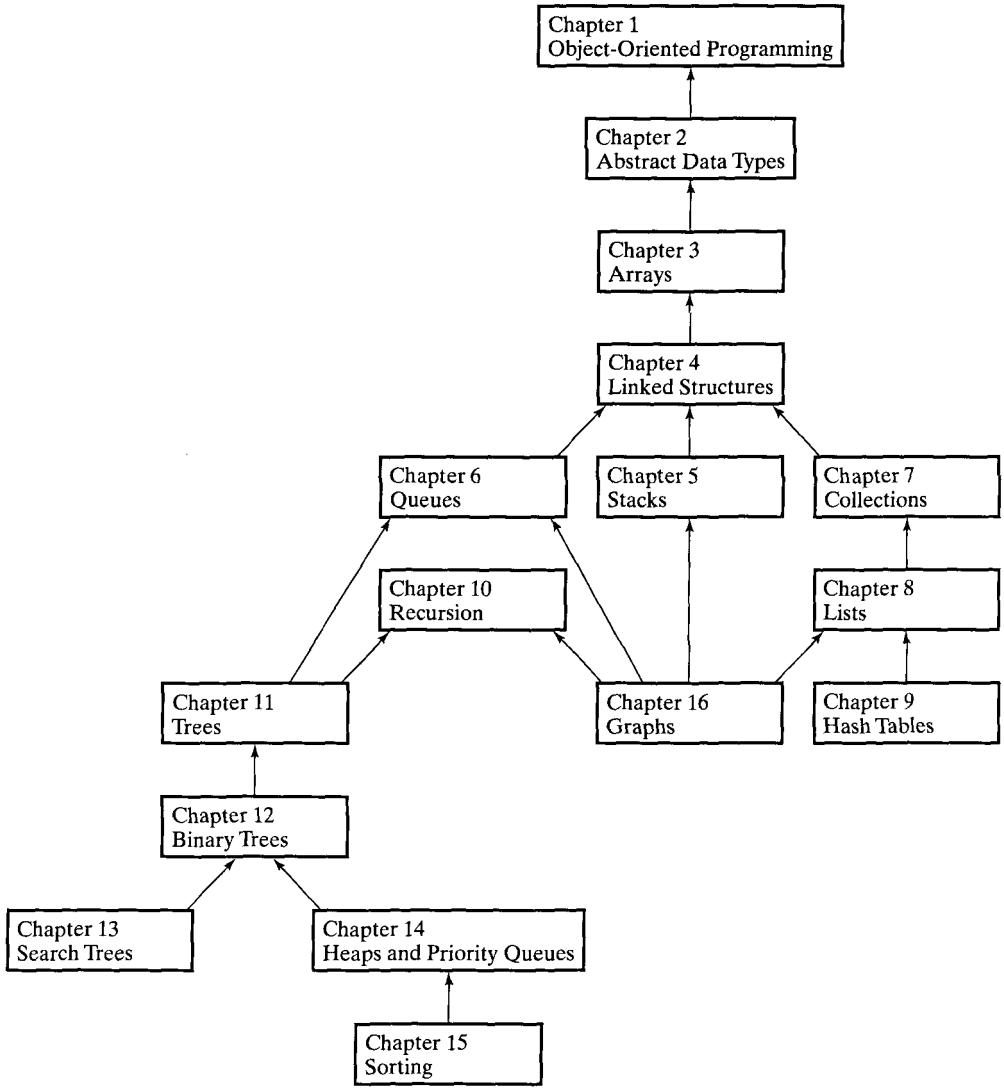
Algorithms are presented explicitly throughout the book. They are introduced in Section 1.5 at a simple level and then studied at progressively deeper levels in later chapters. Complexity analysis is introduced in Section 3.6 and then applied later in chapters 9–16. Loop invariants are finally introduced in Chapter 15 to establish the correctness of the sorting algorithms. Some of these advanced topics can easily be omitted in a one-semester CS2 course.

The chart on the next page shows the chapter dependencies. It shows that the book is flexible enough to support a reordering of some topics. For example, recursion could be done near the beginning of the course, and hash tables could be done at the end of the course.

We use the spiral approach on difficult topics. For example, linked structures are introduced in Chapter 4 as a variant of indirect arrays, and then gradually developed through Chapters 5–8.

The book emphasizes the important distinction between abstract data types (ADTs) and their underlying data structures. ADTs are represented by UML diagrams and realized as Java interfaces. Specific data structures, such as arrays and linked lists, are seen as backing structures for alternative implementations of a given ADT. This important separation of design from implementation is embodied throughout the Java Collections Framework (JCF), which is thoroughly covered in Chapter 7 and Appendix D. Not only does the JCF clarify the value of abstraction, but it also provides the student with many exemplary programming strategies.

Although this is a textbook about data structures, the authors are well aware that students in this course are still learning fundamental programming concepts and Java techniques. Accordingly, we have included many examples and explanations of topics such as polymorphism, simulation, `abstract` classes, inner classes, and reflection.




**Chapter Dependencies**

We are firm believers in the old adage, "A picture is worth 1024 words." So you will find extensive illustrations throughout the book. It contains over 350 figures. It could be subtitled "Visual Data Structures."

We have also tried to make the book as current as possible. It uses Java 1.4, including new features such as the `assert` statement in Chapter 2 and the new `LinkedHash` classes in Appendix D. It also uses Unified Modeling Language (UML) diagrams to summarize class definitions and their relationships.

The companion website for this book is:

`www.prenhall.com/hubbard/`

It contains an on-line study guide and Java source code for all the examples in the book.

An Instructor's Resource CD is also available. It contains Power Point slides, a test bank for each chapter, and solutions to all the exercises and programming problems in the book.

We are grateful for the assistance we have received on this work from our friends at Prentice-Hall and at the University of Richmond. In particular, we wish to thank Matthew Albin, Rom Chan, Natalie Goldberg, Dan Katz, Andrew Lobo, and Molly McCann at Richmond, and Lakshmi Balasubramanian, Petra Recter, and Camille Trentacoste at Prentice-Hall.We also appreciate the suggestions and insights provided by reviewers Benjamin Shults at Western Carolina University, Ted Pawlicki at the University of Rochester, Simon Gray at Ashland University, Frank Coyle at Southern Methodist University, and Bina Ramamurthy at SUNY Buffalo. Finally, we thank our students who used the pre-publication versions in our Data Structures course.

JOHN R. HUBBARD
ANITA HURAY