Object Oriented Data Structures Using Java 4th Edition

Data structures play a key role in any serious development project, determining how the program acquires, stores, updates, and processes its in-memory data. Many of the basic techniques for constructing and governing access to data structures are well-documented, but most are structured programming techniques that do not translate well in an object-oriented environment. Object-Oriented C++ Data Structures for Real Programmers corrects this imbalance, teaching experienced C++ and Java developers the most effective methods for designing and implementing highly functional data structures in any type of object-oriented programming effort. The first part of the book introduces the various approaches, focusing on the purposes for which each is most suited. From there, the author examines advanced functionality that can be achieved in a number of ways, helping readers choose and apply the optimal technique. Key Features * Advanced coverage from an accomplished developer and programming author * Written explicitly for experienced object-oriented programmers * Helps you choose the best way to build the desired functionality, then provides the instruction you need to do it * Covers all major data

structure approaches, including arrays, vectors, lists, stacks, and queues * Explains how to achieve a wide range of functionality, including data sorting, searching, hashing, dictionaries, and indexes
The design and analysis of efficient data structures has long been recognized as a key component of the Computer Science curriculum. Goodrich, Tomassia and Goldwasser's approach to this classic topic is based on the object-oriented paradigm as the framework of choice for the design of data structures. For each ADT presented in the text, the authors provide an associated Java interface. Concrete data structures realizing the ADTs are provided as Java classes implementing the interfaces. The Java code implementing fundamental data structures in this book is organized in a single Java package, net.datastructures. This package forms a coherent library of data structures and algorithms in Java specifically designed for educational purposes in a way that is complimentary with the Java Collections Framework.
Data Structures and Other Objects Using Java is a gradual, "just-in-time" introduction to Data Structures for a CS2 course. Each chapter provides a review of the key aspects of object-oriented programming and a syntax review, giving students the foundation for understanding significant programming concepts. With this framework they are able to accomplish writing functional data structures by using a five-step

method for working with data types; understanding the data type abstractly, writing a specification, using the data type, designing and implementing the data type, and analyzing the implementation. Students learn to think analytically about the efficiency and efficacy of design while gaining exposure to useful Java classes libraries.

Praise for the first edition: "The well-written, comprehensive book...[is] aiming to become a de facto reference for the language and its features and capabilities. The pace is appropriate for beginners; programming concepts are introduced progressively through a range of examples and then used as tools for building applications in various domains, including sophisticated data structures and algorithms...Highly recommended. Students of all levels, faculty, and professionals/practitioners. —D. Papamichail, University of Miami in CHOICE Magazine Mark Lewis' Introduction to the Art of Programming Using Scala was the first textbook to use Scala for introductory CS courses. Fully revised and expanded, the new edition of this popular text has been divided into two books. Object-Orientation, Abstraction, and Data Structures Using Scala, Second Edition is intended to be used as a textbook for a second or third semester course in Computer Science. The Scala programming language provides powerful constructs for expressing both object orientation and abstraction. This book provides

students with these tools of object orientation to help them structure solutions to larger, more complex problems, and to expand on their knowledge of abstraction so that they can make their code more powerful and flexible. The book also illustrates key concepts through the creation of data structures, showing how data structures can be written, and the strengths and weaknesses of each one. Libraries that provide the functionality needed to do real programming are also explored in the text, including GUIs, multithreading, and networking. The book is filled with end-of-chapter projects and exercises, and the authors have also posted a number of different supplements on the book website. Video lectures for each chapter in the book are also available on YouTube. The videos show construction of code from the ground up and this type of "live coding" is invaluable for learning to program, as it allows students into the mind of a more experienced programmer, where they can see the thought processes associated with the development of the code. About the Authors Mark Lewis is an Associate Professor at Trinity University. He teaches a number of different courses, spanning from first semester introductory courses to advanced seminars. His research interests included simulations and modeling, programming languages, and numerical modeling of rings around planets with nearby moons. Lisa Lacher is an Assistant Professor at the

University of Houston, Clear Lake with over 25 years of professional software development experience. She teaches a number of different courses spanning from first semester introductory courses to graduate level courses. Her research interests include Computer Science Education, Agile Software Development, Human Computer Interaction and Usability Engineering, as well as Measurement and Empirical Software Engineering.

This is a complete introduction to the critical topic of data structures, written from the object-oriented perspective most students and practitioners are adopting.The book introduces data structures using C++, a language whose classes and object-oriented constructs are specifically designed to efficiently implement data structures. The opening chapters introduce the ideas behind object-oriented programming and C++; once these ideas are explained, the book introduces data structures and algorithms from an O-O point of view. All standard data structures are described, including stacks, queues, sets, linked lists, trees and graphs. Searching and sorting algorithms are also studied.This book is for students and others working with data structures, especially object-oriented developers interested in ways data structures can enhance their effectiveness.

Data Structures in Java is a continuation of Nell Dale's best-selling Introduction to Java and Software

Design text. Data Structures is designed for students who have already taken one semester of computer science and are able to take a problem of medium complexity, write an algorithm to solve the problem, code the algorithm in a programming language, and demonstrate the correctness of their solution. The focus is on teaching computer science principles with chapter concepts being reinforced by case studies. The object-oriented concepts of encapsulation, inheritance, and polymorphism are covered, while the book remains centered on abstract data types.

This book provides a broad coverage of fundamental and advanced con cepts of data structures and algorithms. The material presented includes a treatment of elementary data structures such as arrays, lists, stacks, and trees, as well as newer structures that have emerged to support the process ing of multidimensional or spatial data files. These newer structures and algorithms have received increasing attention in recent years in conjunc tion with the rapid growth in computer-aided design, computer graphics, and related fields in which multidimensional data structures are of great interest. Our main objective is to mesh the underlying concepts with application examples that are of practical use and are timely in their implementations. To this end, we have used mainly the Abstract Data Structure (or Abstract Data Type

(ADT)) approach to define structures for data and operations. Object-oriented programming (OOP) methodologies are employed to im plement these ADT concepts. In OOP, data and operations for an ADT are combined into a single entity (object). ADTs are used to specifiy the objects-arrays, stacks, queues, trees, and graphs. OOP allows the pro grammer to more closely mimic the real-world applications. This OOP is more structured and modular than previous attempts. OOP has become de facto state-of-the-art in the 1990s.

This compact and comprehensive book provides an introduction to data structures from an object-oriented perspective using the powerful language C++ as the programming vehicle. It is designed as an ideal text for the students before they start designing algorithms in C++. The book begins with an overview of C++, then it goes on to analyze the basic concepts of data structures, and finally focusses the reader's attention on abstract data structures. In so doing, the text uses simple examples to explain the meaning of each data type. Throughout, an attempt has been made to enable students to progress gradually from simple object-oriented abstract data structures to more advanced data structures. A large number of worked examples and the end-of-chapter exercises help the students reinforce the knowledge gained.Intended as a one-semester course for undergraduate students in

computer science and for those who offer this course in engineering and management, the book should also prove highly useful to those IT professionals who have a keen interest in the subject.
About The Book: Bruno Preiss presents readers with a modern, object-oriented perspective for looking at data structures and algorithms, clearly showing how to use polymorphism and inheritance, and including fragments from working and tested programs. The book uses a single class hierarchy as a framework to present all of the data structures. This framework clearly shows the relationships between data structures and illustrates how polymorphism and inheritance can be used effectively.
"It is a practical book with emphasis on real problems the programmers encounter daily."
--Dr.Tim H. Lin, California State Polytechnic University, Pomona "My overall impressions of this book are excellent. This book emphasizes the three areas I want: advanced C++, data structures and the STL and is much stronger in these areas than other competing books." --Al Verbanec, Pennsylvania State University Think, Then Code When it comes to writing code, preparation is crucial to success. Before you can begin writing successful code, you need to first work through your options and analyze the expected performance of your design. That's why Elliot Koffman and Paul Wolfgang's Objects, Abstraction, Data Structures, and Design: Using C++

encourages you to Think, Then Code, to help you make good decisions in those critical first steps in the software design process. The text helps you thoroughly understand basic data structures and algorithms, as well as essential design skills and principles. Approximately 20 case studies show you how to apply those skills and principles to real-world problems. Along the way, you'll gain an understanding of why different data structures are needed, the applications they are suited for, and the advantages and disadvantages of their possible implementations. Key Features * Object-oriented approach. * Data structures are presented in the context of software design principles. * 20 case studies reinforce good programming practice. * Problem-solving methodology used throughout... "Think, then code!" * Emphasis on the C++ Standard Library. * Effective pedagogy.

Object-Oriented Data Structures Using Java, Fourth Edition presents traditional data structures and object-oriented topics with an emphasis on problem-solving, theory, and software engineering principles.

Data Structures & Theory of Computation

Now in its second edition, D.S. Malik brings his proven approach to C++ programming to the CS2 course. Clearly written with the student in mind, this text focuses on Data Structures and includes advanced topics in C++ such as Linked Lists and the Standard Template Library (STL). The text features abundant visual diagrams, examples, and extended Programming Examples, all of which serve to

illuminate difficult concepts. Complete programming code and clear display of syntax, explanation, and example are used throughout the text, and each chapter concludes with a robust exercise set. Important Notice: Media content referenced within the product description or the product text may not be available in the ebook version.

Harness the power of Python objects and data structures to implement algorithms for analyzing your data and efficiently extracting information Key Features Turn your designs into working software by learning the Python syntax Write robust code with a solid understanding of Python data structures Understand when to use the functional or the OOP approach Book Description This Learning Path helps you get comfortable with the world of Python. It starts with a thorough and practical introduction to Python. You'll quickly start writing programs, building websites, and working with data by harnessing Python's renowned data science libraries. With the power of linked lists, binary searches, and sorting algorithms, you'll easily create complex data structures, such as graphs, stacks, and queues. After understanding cooperative inheritance, you'll expertly raise, handle, and manipulate exceptions. You will effortlessly integrate the object-oriented and not-so-object-oriented aspects of Python, and create maintainable applications using higher level design patterns. Once you've covered core topics, you'll understand the joy of unit testing and just how easy it is to create unit tests. By the end of this Learning Path, you will have built components that are easy to understand, debug, and can be used across different applications. This Learning Path includes content from the following Packt products: Learn Python Programming - Second Edition by Fabrizio Romano Python Data Structures and Algorithms by Benjamin Baka Python 3 Object-Oriented Programming by Dusty Phillips What you will learn Use data structures and control

flow to write code Use functions to bundle together a sequence of instructions Implement objects in Python by creating classes and defining methods Design public interfaces using abstraction, encapsulation and information hiding Raise, define, and manipulate exceptions using special error objects Create bulletproof and reliable software by writing unit tests Learn the common programming patterns and algorithms used in Python Who this book is for If you are relatively new to coding and want to write scripts or programs to accomplish tasks using Python, or if you are an object-oriented programmer for other languages and seeking a leg up in the world of Python, then this Learning Path is for you. Though not essential, it will help you to have basic knowledge of programming and OOP.

A book for an undergraduate course on data structures which integrates the concepts of object-oriented programming and GUI programming.

Data Structures and Object-Oriented Programming with C++ has been specifically designed and written to meet the requirements of the engineering students. This is a core subject in the curriculum of all Computer Science programs. The aim of this book is to help the students develop programming and analytical skills simultaneously such that they are able to design programs with maximum efficiency.C language has been used in the book to permit the execution of basic data structures in a variety of ways. This book also provides an in-depth coverage of object-oriented concepts, such as encapsulation, abstraction, inheritance, polymorphism, message passing and dynamic binding, templates, exception handling, streams and standard template library (STL) in C++.

THIS TEXTBOOK is about computer science. It is also about Python. However, there is much more. The study of algorithms and data structures is central to understanding

what computer science is all about. Learning computer science is not unlike learning any other type of difficult subject matter. The only way to be successful is through deliberate and incremental exposure to the fundamental ideas. A beginning computer scientist needs practice so that there is a thorough understanding before continuing on to the more complex parts of the curriculum. In addition, a beginner needs to be given the opportunity to be successful and gain confidence. This textbook is designed to serve as a text for a first course on data structures and algorithms, typically taught as the second course in the computer science curriculum. Even though the second course is considered more advanced than the first course, this book assumes you are beginners at this level. You may still be struggling with some of the basic ideas and skills from a first computer science course and yet be ready to further explore the discipline and continue to practice problem solving. We cover abstract data types and data structures, writing algorithms, and solving problems. We look at a number of data structures and solve classic problems that arise. The tools and techniques that you learn here will be applied over and over as you continue your study of computer science.

The Object of Data Abstraction and Structures Using Java is the perfect book for your data structures course. It presents traditional data structures topics with a distinct object-oriented flavor that offers students useful approaches for data structure design and implementation.

Guide to the object-oriented programming language Object-oriented database management systems (OODBMS) are used to imple ment and maintain large object databases on persistent storage. Regardless whether the underlying database model follows the object-oriented, the rela tional or the object-relational paradigm, a key feature of any DBMS product is content based access to data sets. On the one

hand this feature provides user-friendly query interfaces based on predicates to describe the desired data. On the other hand it poses challenging questions regarding DBMS design and implementation as well as the application development process on top of the DBMS. The reason for the latter is that the actual query performance depends on a technically meaningful use of access support mechanisms. In particular, if chosen and applied properly, such a mechanism speeds up the execution of predicate based queries. In the object-oriented world, such queries may involve arbitrarily complex terms referring to inheritance hierarchies and aggregation paths. These features are attractive at the application level, however, they increase the complexity of appropriate access support mechanisms which are known to be technically non-trivial in the relational world. Provides a comprehensive coverage of the subject, Includes numerous illustrative example, Demonstrate the development of algorithms in a lucid manner, Demonstrate the implementation of algorithms in a good programming style, provides challenging programming exercise to test you knowledge gained about the subject, Glossary of terms for ready reference Takes a gentle approach to learning data structures using the Java programming language. Providing an early, self-contained review of object-oriented programming and Java, this text gives readers a firm grasp of key concepts and allows those experienced in another language to adjust easily. It has a solid foundation in building and using abstract data types, along with an assortment of advanced topics such as B-trees for project building and graph. It incorporates Java 5.0 including the use of scanner class and generic data types (generics). MARKET: This book is if for anyone interested in learning how to write effective data structures using the Java language.

Bringing together the fundamental topics of a traditional introductory data structures course and the current world of C++ and object-oriented programming, Data Structures via C++: Objects by Evolution offers an evolutionary approach to the subject. It combines a sound pedagogy for teaching data structures at the introductory (CS2) level with modern ideas in software engineering and object-oriented programming. The book introduces students (and instructors) to C++ and object-oriented programming using a "just-in-time" approach which leads readers from traditional techniques to more current ideas. This text emphasizes abstraction by introducing each new data structure first as an abstract data type (ADT), then discussing the external interface, and following with implementation. The primary data structures included are lists, stacks, queues, tables, trees, and graphs. All examples are developed using C++, and advanced features are introduced as needed or just-in-time. Berman's real-world examples, such as simulation of an Ethernet, robot navigation, and expression processing, help to illustrate use of data structures in concrete terms. C++ language features and object-oriented concepts, both very useful in solving problems encountered in the course, are also covered. Techniques of object-oriented programming are introduced, with a strong emphasis on encapsulation and detailed coverage of inheritance. An overview of software engineering is presented, including discussion of the software life-cycle, design, testing, assertions and loop invariants, and abstract data types. All supporting materials will be available to faculty and students via the World Wide Web at: http: //www.rowan.edu/evolve.
Object-Oriented Data Structures Using JavaJones & Bartlett Publishers
This introduction to the Java language integrates a

discussion of object-oriented programming with the design and implementation of data structures. It covers the most important topics, including algorithm analysis; time and space complexities; Java built-in data structure classes; input and output, data, and access streams; and the persistency of data.

Based on the authors' market leading data structures books in Java and C++, this textbook offers a comprehensive, definitive introduction to data structures in Python by authoritative authors. Data Structures and Algorithms in Python is the first authoritative object-oriented book available for the Python data structures course. Designed to provide a comprehensive introduction to data structures and algorithms, including their design, analysis, and implementation, the text will maintain the same general structure as Data Structures and Algorithms in Java and Data Structures and Algorithms in C++.

This textbook provides an in depth course on data structures in the context of object oriented development. Its main themes are abstraction, implementation, encapsulation, and measurement: that is, that the software process begins with abstraction of data types, which then lead to alternate representations and encapsulation, and finally to resource measurement. A clear object oriented approach, making use of Booch components, will provide readers with a useful library of data structure components and experience in software reuse. Students using this book are expected to have a reasonable understanding of the basic logical structures such as stacks and queues. Throughout, Ada 95 is used

and the author takes full advantage of Ada's encapsulation features and the ability to present specifications without implementational details. Ada code is supported by two suites available over the World Wide Web.
Computer Science
This new book provides a concise and engaging introduction to Java and object-oriented programming with an abundance of original examples, use of Unified Modeling Language throughout, and coverage of the new Java 1.5. Addressing critical concepts up front, the book's five-part structure covers object-oriented programming, linear structures, algorithms, trees and collections, and advanced topics. KEY FEATURES: "Data Structures and Algorithms in Java" takes a practical approach to real-world programming and introduces readers to the process of crafting programs by working through the development of projects, often providing multiple versions of the code and consideration for alternate designs. The book features the extensive use of games as examples; a gradual development of classes analogous to the Java Collections Framework; complete, working code in the book and online; and strong pedagogy including extended examples in most chapters along with exercises, problems and projects. For readers and professionals with a familiarity with the basic control structures of Java or C and a precalculus level of mathematics who want to expand their knowledge to Java data structures and algorithms. Ideal for a second undergraduate course in computer science.
Copyright: 438b617a80722499e079249917610576