**OA: Principles of Service Design,** Thomas Erl. The key to succeeding with service-oriented architecture (SOA) lies in comprehending the meaning and significance of its most fundamental building block: the service. Only by understanding service design can truly service-oriented solution logic be created to support achieving the strategic goals associated with SOA and service-oriented computing.

The author guides readers through an exploration of service-orientation design, revealing exactly how services should and should not be designed for real-world SOA. The book includes a concise introduction to SOA and service-oriented computing concepts and benefits. It also provides a thorough exploration of the service-orientation design paradigm as represented by eight specific design principles. The author then presents an exhaustive examination of service contracts, emphasizing standardization, abstraction, and the utilization of WS-Policy, XML Schema, and WSDL definitions.

Prentice Hall PTR; www.phptr.com; 0-13-234482-3; 608 pp.

**Chases and Escapes: The Mathematics of Pursuit and Evasion,** Paul J. Nahin. The simple chase game of tag and its variants represent an application of pursuit theory that uses the same principles found in military strategy, high-seas chases by the Coast Guard, and even romantic pursuits. This volume relates the first complete history of this fascinating mathematics area.

Writing in an accessible style, the author traces the development of modern pursuit theory from its classical analytical beginnings to the present day. Along the way, he informs his mathematical discussions with fun facts and captivating stories, inviting readers to explore the different approaches to solving various chase-and-escape problems, drawing upon game theory, geometry, linear algebra, target-tracking algorithms, and much more.


This second volume continues the story of music engineering, focusing on the digital and computational domain. The author goes deeper into the mathematics of music and sound, beginning with digital audio, sampling, and binary numbers, as well as complex numbers and how they simplify representation of musical signals. The book devotes a chapter each to the Fourier transform, convolution, filtering, resonance, the wave equation, acoustical systems, sound synthesis, the short-time Fourier transform, and the wavelet transform. These subjects provide the theoretical underpinnings of today’s music technology.

Cross-references for concepts introduced in the first volume are included, as well as additional mathematical orientation where necessary. The topics are all subjects that contemporary composers, musicians, and music engineers have found important, while the examples explore practical problems in music and audio.


**Continuous Integration: Improving Software Quality and Reducing Risk,** Paul Duvall, Steve Matyas, and Andrew Glover. For any software developer who has spent days in “integration hell,” cobbling together myriad software components, this book illustrates how to transform integration from a necessary evil into an everyday part of the development process. The key, the authors show, is to integrate regularly and often using continuous integration (CI) practices and techniques.

The authors first examine the concept of CI and its practices from the ground up, then explore other effective processes that CI systems perform, such as database integration, testing, inspection, deployment, and feedback. Readers learn that CI can lead to more rapid software development, produce deployable software at every step in the development life cycle, and reduce the time between defect introduction and detection, lowering costs.

Addison-Wesley; www.awprofessional.com; 0-321-33638-0; 336 pp.

**Java Methods for Financial Engineering: Applications in Finance and Investment,** Phil Barker. This book explores the main theories and models that practitioners use to engineer finance and investment tools. Each chapter is largely self-contained, letting readers conveniently focus on a defined tool and providing immediate access to an implemented solution. Those engaged in the design and evaluation of new products will find the quick access to a wide range of robust core methods valuable in constructing bespoke implementations.

All of the methods in this book are written in Java and use the fully optimized Java Collections for data manipulation. For practitioners and students alike who still work with legacy C++ or Visual Basic implementations, this book can serve as a reference for translating or porting their applications into a Web-centric environment.

Springer; www.springer.com; 978-1-85233-832-9; 568 pp.

Send book announcements to newbooks@computer.org.