Read Returns

The highly praised book in communications networking from IEEE Press, now available in the Eastern Economy Edition.This is a non-mathematical introduction to Distributed Operating Systems explaining the fundamental concepts and design principles of this emerging technology. As a textbook for students and as a self-study text for systems managers and software engineers, this book provides a concise and an informal introduction to the subject.
CD-ROM contains: Examples for text -- Toon3DCreator 1.7 with full source code.
"Get the Java skills you will need to start developing Android apps apps"--Cover.
Provides information on writing a driver in Linux, covering such topics as character devices, network interfaces, driver debugging, concurrency, and interrupts.
Both authors have taught the course of “Distributed Systems” for many years in the respective schools. During the teaching, we feel strongly that “Distributed systems” have evolved from traditional “LAN” based distributed systems towards “Internet based” systems. Although there exist many excellent textbooks on this topic, because of the fast development of distributed systems and network programming/protocols, we have difficulty in finding an appropriate textbook for the course of “distributed systems” with orientation to the requirement of the undergraduate level study for today’s distributed technology. Specifically, from - to-date concepts, algorithms, and models to implementations for both distributed system designs and application programming. Thus the philosophy behind this book is to integrate the concepts, algorithm designs and implementations of distributed systems based on network programming. After using several materials of other textbooks and research books, we found that many texts

treat the distributed systems with separation of concepts, algorithm design and network programming and it is very difficult for students to map the concepts of distributed systems to the algorithm design, prototyping and implementations. This book intends to enable readers, especially postgraduates and senior undergraduate level, to study up-to-date concepts, algorithms and network programming skills for building modern distributed systems. It enables students not only to master the concepts of distributed network system but also to readily use the material introduced into implementation practices.

”Tiny Python Projects is a gentle and amusing introduction to Python that will firm up key programming concepts while also making you giggle.”—Amanda Debler, Schaeffler Key Features Learn new programming concepts through 21-bitesize programs Build an insult generator, a Tic-Tac-Toe AI, a talk-like-a-pirate program, and more Discover testing techniques that will make you a better programmer Code-along with free accompanying videos on YouTube Purchase of the print book includes a free eBook in PDF, Kindle, and ePub formats from Manning Publications. About The Book The 21 fun-but-powerful activities in Tiny Python Projects teach Python fundamentals through puzzles and games. You’ll be engaged and entertained with every exercise, as you learn about text manipulation, basic algorithms, and lists and dictionaries, and other foundational programming skills. Gain confidence and experience while you create each satisfying project. Instead of going quickly through a wide range of concepts, this book concentrates on the most useful skills, like text manipulation, data structures, collections, and program logic with projects that include a password creator, a word

rhymer, and a Shakespearean insult generator. Author Ken Youens-Clark also teaches you good programming practice, including writing tests for your code as you go. What You Will Learn Write command-line Python programs Manipulate Python data structures Use and control randomness Write and run tests for programs and functions Download testing suites for each project This Book Is Written For For readers familiar with the basics of Python programming. About The Author Ken Youens-Clark is a Senior Scientific Programmer at the University of Arizona. He has an MS in Biosystems Engineering and has been programming for over 20 years. Table of Contents 1 How to write and test a Python program 2 The crow's nest: Working with strings 3 Going on a picnic: Working with lists 4 Jump the Five: Working with dictionaries 5 Howler: Working with files and STDOUT 6 Words count: Reading files and STDIN, iterating lists, formatting strings 7 Gashlycrumb: Looking items up in a dictionary 8 Apples and Bananas: Find and replace 9 Dial-a-Curse: Generating random insults from lists of words 10 Telephone: Randomly mutating strings 11 Bottles of Beer Song: Writing and testing functions 12 Ransom: Randomly capitalizing text 13 Twelve Days of Christmas: Algorithm design 14 Rhymer: Using regular expressions to create rhyming words 15 The Kentucky Friar: More regular expressions 16 The Scrambler: Randomly reordering the middles of words 17 Mad Libs: Using regular expressions 18 Gematria: Numeric encoding of text using ASCII values 19 Workout of the Day: Parsing CSV files, creating text table output 20 Password strength: Generating a secure and memorable password

21 Tic-Tac-Toe: Exploring state 22 Tic-Tac-Toe redux: An interactive version with type hints

An interactive way to introduce the world of Python Programming KEY FEATURES Detailed comparisons and differentiation of python language from other most popular languages C/C++/Java. Authentic and extensive set of programming illustrations in every chapter of the book. Broad study on all the programming constructs of the python programming language such as native data types, looping, decision making, exception handling, file handling etc. Broad study of Python Object Oriented Programming features with illustrations. Numerous review questions and exercises at the end of every chapter. DESCRIPTION This Book is meant for wide range of readers who wish to learn the basics of Python programming language. It can be helpful for students, programmers, researchers, and software developers. The basic concepts of python programming are dealt in detail. The various concepts of python language such as object-oriented features, operators, native data types, control structures, functions, exception handling, file handling, etc are discussed in detail with the authentic programming illustration of each. presently, python programming is a hot topic among academician's researchers, and program developers. As a result, the book is designed to give an in-depth knowledge of programming in python. This book can be used as handbook as well as a guide for students of all computer science stream at any grade beginning from 10+1 to Research in PhD. To conclude, we hope that the readers will

find this book a helpful guide and valuable source of information about python programming. WHAT WILL YOU LEARN Python Data Types, Input Output Operators and Expressions Control Structures Python Functions, Modules Exception Handling File Management, Classes and Objects Inheritance, Python Operator Overloading WHO THIS BOOK IS FOR Students, Programmers, researchers, and software developers who wish to learn the basics of Python programming language. Table of Contents 1. Introduction to Python Language 2. Python Data Types and Input Output 3. Operators and Expressions 4. Control Structures 5. Python Native Data Types 6. Python Functions 7. Python Modules 8. Exception Handling 9. File Management in Python 10. Classes and Objects 11. Inheritance 12. Python Operator Overloading
The Material Point Method: A Continuum-Based Particle Method for Extreme Loading Cases systematically introduces the theory, code design, and application of the material point method, covering subjects such as the spatial and temporal discretization of MPM, frequently-used strength models and equations of state of materials, contact algorithms in MPM, adaptive MPM, the hybrid/coupled material point finite element method, object-oriented programming of MPM, and the application of MPM in impact, explosion, and metal forming. Recent progresses are also stated in this monograph, including improvement of efficiency, memory storage, coupling/combination with the finite element method, the contact algorithm, and their application to problems. Provides a user’s guide and several numerical examples of the MPM3D-F90 code that

can be downloaded from a website Presents models that describe different types of material behaviors, with a focus on extreme events. Includes applications of MPM and its extensions in extreme events, such as transient crack propagation, impact/penetration, blast, fluid-structure interaction, and biomechanical responses to extreme loading

A smart, back-to-the-basics approach for generating abnormally high returns Turn the TV on and you'll hear a chorus of voices telling you where, when, why, and how to invest your money. Founder and editor of the popular investing blog Abnormal Returns Tadas Viskanta has some advice: Don't listen to them. The truth is, all that noise will just confuse you. In Abnormal Returns, Viskanta reveals the simple truths about fixed income investing, risk management, portfolio management, global investing, ETFs, and active investing. In no time, you'll have the knowledge you need to address your portfolio issues with skill and confidence. Prices are low and access to quality information is more abundant than ever. Now is the time to kick your investing into high gear with Abnormal Returns.

Louis L'Amour's long-lost first novel, faithfully completed by his son, takes readers on a voyage into danger and violence on the high seas. Fate is a ship. As the shadows of World War II gather, the SS Lichenfield is westbound across the Pacific carrying eighty thousand barrels of highly explosive naphtha. The cargo alone makes the journey perilous, with the entire crew aware that one careless moment could lead to disaster.

But another sort of peril haunts the Lichenfield. Even beyond their day-to-day existence, the lives of the crew are mysteriously intertwined. Though each has his own history, dreams and hardships, all are connected by a deadly web of chance and circumstance. Some are desperately fleeing the past; others chase an unknown destiny. But in their hearts, these men carry the seeds of salvation or destruction. And all of them--kind or cruel, strong or broken--are bound to the fate of the vessel that carries them toward an ever-darkening horizon. Inspired by Louis L'Amour's own experiences as a merchant seaman, No Traveller Returns is a revelatory work by a world-renowned author--and a brilliant illustration of a writer discovering his literary voice.

This book constitutes the refereed proceedings of the 17th International Conference on Principles of Distributed Systems, OPODIS 2013, held in Nice, France, in December 2013. The 19 papers presented together with two invited talks were carefully reviewed and selected from 41 submissions. The conference is an international forum for the exchange of state-of-the-art knowledge on distributed computing and systems. Papers were sought soliciting original research contributions to the theory, specification, design and implementation of distributed systems.

Appendices accompany vols. 64, 67-71.

This book contains letters from Mary Shelley's life from 1797 to 1822.

Some may say that this book is long overdue; others, including myself, will state that the book appears atjust the right time. The latter is likely more true, for it is doubtful that many in the professions would, until now, link issues of learning disabilities with those of neurophysiological

dysfunction in the manner in which ultimately must be the case. As a matter of fact, there are those who deny the relationship completely. Lee Wiederholt (1974)1 in his short, but excellent, review of the historical perspectives of learning disabilities, traces the early interest in this problem to the work of Gall (1802), and to his successors Broca (1861), Jackson (1864), Bastian (1869), and a few others. Each of these men would, at the time of this writing, be considered to have interests in the field of neurology, although at the time of their investigations, neurology per se was but a gleam in the eye of the anatomical beholder. A relative detour then took place. Cerebral palsy, in the decades of the 1940s and 1950s, caught the attention of researchers through the work of Winthrop Morgan Phelps (orthopedist) and George Deaver (physiatrist) and one or two other medically oriented individuals. This was related to the writings of W. J. Little (1810-1894). It was, however, Kurt Goldstein, Heinz Wemer, both eminent German scientists, and Alfred A.

Hacking is the art of creative problem solving, whether that means finding an unconventional solution to a difficult problem or exploiting holes in sloppy programming. Many people call themselves hackers, but few have the strong technical foundation needed to really push the envelope. Rather than merely showing how to run existing exploits, author Jon Erickson explains how arcane hacking techniques actually work. To share the art and science of hacking in a way that is accessible to everyone, Hacking: The Art of Exploitation, 2nd Edition introduces the fundamentals of C programming from a hacker's perspective. The included LiveCD provides a complete Linux programming and debugging environment—all without modifying your current operating system. Use it to follow along with the book's examples as you fill gaps in your knowledge and explore hacking techniques on your own. Get your hands

dirty debugging code, overflowing buffers, hijacking network communications, bypassing protections, exploiting cryptographic weaknesses, and perhaps even inventing new exploits. This book will teach you how to: – Program computers using C, assembly language, and shell scripts – Corrupt system memory to run arbitrary code using buffer overflows and format strings – Inspect processor registers and system memory with a debugger to gain a real understanding of what is happening – Outsmart common security measures like nonexecutable stacks and intrusion detection systems – Gain access to a remote server using port-binding or connect-back shellcode, and alter a server's logging behavior to hide your presence – Redirect network traffic, conceal open ports, and hijack TCP connections – Crack encrypted wireless traffic using the FMS attack, and speed up brute-force attacks using a password probability matrix Hackers are always pushing the boundaries, investigating the unknown, and evolving their art. Even if you don't already know how to program, Hacking: The Art of Exploitation, 2nd Edition will give you a complete picture of programming, machine architecture, network communications, and existing hacking techniques. Combine this knowledge with the included Linux environment, and all you need is your own creativity.

For more than twenty years, serious C programmers have relied on one book for practical, in-depth knowledge of the programming interfaces that drive the UNIX and Linux kernels: W. Richard Stevens' Advanced Programming in the UNIX® Environment . Now, once again, Rich's colleague Steve Rago has thoroughly updated this classic work. The new third edition supports today's leading platforms, reflects new technical advances and best practices, and aligns with Version 4 of the Single UNIX

Specification. Steve carefully retains the spirit and approach that have made this book so valuable. Building on Rich's pioneering work, he begins with files, directories, and processes, carefully laying the groundwork for more advanced techniques, such as signal handling and terminal I/O. He also thoroughly covers threads and multithreaded programming, and socket-based IPC. This edition covers more than seventy new interfaces, including POSIX asynchronous I/O, spin locks, barriers, and POSIX semaphores. Most obsolete interfaces have been removed, except for a few that are ubiquitous. Nearly all examples have been tested on four modern platforms: Solaris 10, Mac OS X version 10.6.8 (Darwin 10.8.0), FreeBSD 8.0, and Ubuntu version 12.04 (based on Linux 3.2). As in previous editions, you'll learn through examples, including more than ten thousand lines of downloadable, ISO C source code. More than four hundred system calls and functions are demonstrated with concise, complete programs that clearly illustrate their usage, arguments, and return values. To tie together what you've learned, the book presents several chapter-length case studies, each reflecting contemporary environments. Advanced Programming in the UNIX® Environment has helped generations of programmers write code with exceptional power, performance, and reliability. Now updated for today's systems, this third edition will be even more valuable.

Concurrent systems are generally understood in terms of behavioral notions. Models for Concurrency analyzes the subject in terms of events and their temporal relationship

rather than on global states. It presents a comprehensive analysis of model theory applied to concurrent protocols, and seeks to provide a theory of concurrency that is both intuitively appealing and rigorously based on mathematical foundations. The book is divided into three main sections. The first introduces the required concepts from model theory, details the structures that are used to model concurrency, gives an in-depth description and explanation of the semantics of a simple language that allows concurrent execution of sequential programs, and deals with the question of resolving executions into higher-level and lower-level granularities. The second and third sections apply the theory developed to practical examples, and an exposition of the producer/consumer problem with details of two solutions is given. The author also deals with message passing, as opposed to shared memory.

Programming knowledge is often necessary for finding a solution to a biological problem. Based on the author’s experience working for an agricultural biotechnology company, Python for Bioinformatics helps scientists solve their biological problems by helping them understand the basics of programming. Requiring no prior knowledge of programming-related concepts, the book focuses on the easy-to-use, yet powerful, Python computer language. The book begins with a very basic introduction that teaches the principles of programming. It then introduces the Biopython package, which can be useful in solving life science problems. The next section covers sophisticated tools for bioinformatics, including relational database management systems and XML. The last

part illustrates applications with source code, such as sequence manipulation, filtering vector contamination, calculating DNA melting temperature, parsing a genbank file, inferring splicing sites, and more. The appendices provide a wealth of supplementary information, including instructions for installing Python and Biopython and a Python language and style guide. By incorporating examples in biology as well as code fragments throughout, the author places a special emphasis on practice, encouraging readers to experiment with the code. He shows how to use Python and the Biopython package for building web applications, genomic annotation, data manipulation, and countless other applications.

Essential Java Skills--Made Easy! What Special – In this book I covered and explained several topics of latest Java 8 Features in detail for Developers & Fresher's, Topics Like– Lambdas. || Java 8 Functional interface, || Stream and Time API in Java 8. This Java book doesn't require previous programming experience. However, if you come from a C or C++ programming background, then you will be able to learn faster. Learn the all basics and advanced features of Java programming in no time from Bestseller Java Programming Author Harry. H. Chaudhary (More than 1,67,000 Books Sold !). This Java Guide, starts with the basics and Leads to Advance features of Java in detail with thousands of Java Codes and new features of Java 8 like Lambdas. Java 8 Functional interface, || Stream and Time API in Java 8. , I promise this book will make you expert level champion of java. Anyone can learn java through this book at expert

level. The main objective of this java book is not to give you just Java Programming Knowledge, I have followed a pattern of improving the question solution of thousands of Codes with clear theory explanations with different Java complexities for each java topic problem, and you will find multiple solutions for complex java problems. Engineering Students and fresh developers can also use this book. This book covers common core syllabus for all Computer Science Professional Degrees If you are really serious then go ahead and make your day with this ultimate java book. First Part- Teach you how to compile and run a Java program, shows you everything you need to develop, compile, debug, and run Java programs. And then discusses the keywords, syntax, and constructs that form the core of the Java language. After that it leads you to advanced features of java, including multithreaded programming and Applets. Learning a new language is no easy task especially when it’s an oop’s programming language like Java. You might think the problem is your brain. It seems to have a mind of its own, a mind that doesn't always want to take in the dry, technical stuff you're forced to study. The fact is your brain craves novelty. This Java Book is very serious java stuff: A complete introduction to Java. You'll learn everything from the fundamentals to advanced topics, if you've read this book, you know what to expect--a visually rich format designed for the way your brain works. To use this book does not require any previous programming experience. However, if you come from a C/C++ background, then you will be able to advance a bit more rapidly. As most readers will know, Java is

similar, in form and spirit, to C/C++. Thus, knowledge of those languages helps, but is not necessary. Even if you have never programmed before, you can learn to program in Java using this book. Inside Contents (Chapters): 1. (Overview of Java) 2.(Java Language) 3.(Control Statements) 4.(Scanner class, Arrays & Command Line Args) 5.(Class & Objects in Java) 6.(Inheritance in Java) 7.(Object oriented programming) 8.(Packages in Java) 9.(Interface in Java) 10.(String and StringBuffer) 11.(Exception Handling) 12.(Multi-Threaded Programming) 13.(Modifiers/Visibility modes) 14.(Wrapper Class) 15.(Input/Output in Java) 16.(Applet Fundamentals) 17.(Abstract Windows Toolkit)(AWT) 18.(Introduction To AWT Events) 19.(Painting in AWT) 20.(java.lang.Object Class) 21.(Collection Framework) PART - II (Java 8 Features for Developers) 22. Java 8 Features for Developers – Lambdas. 23. Java 8 Functional interface,Stream & Time API. 24. Key Features that Make Java More Secure than Other Languages.

Abstract: "Formal proofs generated mechanically by theorem provers are often very large and shallow, and the theorem provers are themselves very complex. Therefore, in certain application areas, such as in safety-critical systems, it is necessary to have an independent means for ensuring the consistency of such formal proofs. This report describes an efficient proof checker for the HOL theorem prover. This proof checker has been tested with practical proofs consisting of thousands of inference steps. It was implemented in Standard ML of New Jersey. The first part of the report gives an

overview of the program. It describes the rationale of developing a proof checker, how to use the checker, and how the checker works. The second part of the report describes the program in detail. The complete source code is included in the description."

Of all the Ajax-specific frameworks that have popped up in recent years, one clearly stands out as the industrial strength solution. Dojo is not just another JavaScript toolkit—it's the JavaScript toolkit—and Dojo: The Definitive Guide demonstrates how to tame Dojo's extensive library of utilities so that you can build rich and responsive web applications like never before. Dojo founder Alex Russell gives a foreword that explains the "why" of Dojo and of this book. Dojo provides an end-to-end solution for development in the browser, including everything from the core JavaScript library and turnkey widgets to build tools and a testing framework. Its vibrant open source community keeps adding to Dojo's arsenal, and this book provides an ideal companion to Dojo's official documentation. Dojo: the Definitive Guide gives you the most thorough overview of this toolkit available, showing you everything from how to create complex layouts and form controls closely resembling those found in the most advanced desktop applications with stock widgets, to advanced JavaScript idioms to AJAX and advanced communication transports. With this definitive reference you get:

Get a concise introduction to Dojo that's good for all 1.x versions Well-explained examples, with scores of tested code samples, that let you see Dojo in action A comprehensive reference to Dojo's standard JavaScript library (including fundamental utilities in Base, Dojo's tiny but powerful kernel) that you'll wonder how you ever lived without An extensive look at additional Core features, such as animations, drag-and-drop, back-button handling, animations like wipe and slide, and more Exhaustive coverage of out-of-the-box Dijits (Dojo widgets) as well as definitive coverage on how to create your own, either from scratch or building on existing ones An itemized inventory of DojoX subprojects, the build tools, and the DOH, Dojo's unit-testing framework that you can use with Dojo—or anywhere else If you're a DHTML-toting web developer, you need to read this book—whether you're a one-person operation or part of an organization employing scores of developers. Dojo packs the standard JavaScript library you've always wanted, and Dojo: The Definitive Guide helps you transform your ideas into working applications quickly by leveraging design concepts you already know.