Object Oriented Software Engineering David Kung Pdf

Eventually, you will no question discover a other experience and endowment by spending more cash. nevertheless when? reach you endure that you require to get those every needs taking into consideration having significantly cash? Why dont you attempt to get something basic in the beginning? Thats something that will guide you to comprehend even more not far off from the globe, experience, some places, in the manner of history, amusement, and a lot more?

It is your categorically own mature to feign reviewing habit. among guides you could enjoy now is **Object Oriented Software Engineering David Kung Pdf** below.

Introduction to Software Testing - Paul Ammann 2008-01-28
Extensively class-tested, this textbook takes an innovative approach to software testing: it defines testing as the process of applying a few well-defined, general-purpose test criteria to a structure or model of the software. It incorporates the latest innovations in testing, including techniques to test modern types of software such as OO, web applications, and embedded software. The book contains numerous examples throughout. An instructor's solution manual, PowerPoint slides, sample syllabi, additional examples and updates, testing tools for students, and example software programs in Java are available on an extensive website.

Software Testing and Quality Assurance - Kshirasagar Naik 2011-09-23
A superior primer on software testing and quality assurance, from integration to execution and automation This important new work fills the pressing need for a user-friendly text that aims to provide software engineers, software quality professionals, software developers, and students with the fundamental developments in testing theory and common testing practices. Software Testing and Quality Assurance: Theory and Practice equips readers with a solid understanding of: Practices that support the production of quality software Software testing techniques Life-cycle models for requirements, defects, test cases, and test results Process models for units, integration, system, and acceptance testing How to build test teams, including recruiting and retaining test engineers Quality Models, Capability Maturity Model, Testing Maturity Model, and Test Process Improvement Model Expertly balancing theory with practice, and complemented with an abundance of pedagogical tools, including test questions, examples, teaching suggestions, and chapter summaries, this book is a valuable, self-contained tool for professionals and an ideal introductory text for courses in software testing, quality assurance, and software engineering.

Operating Systems - Thomas Anderson 2014
Over the past two decades, there has been a huge amount of innovation in both the principles and practice of operating systems Over the same period, the core ideas in a modern operating system - protection, concurrency, virtualization, resource allocation, and reliable storage - have become widely applied throughout computer science. Whether you get a job at Facebook, Google, Microsoft, or any other leading-edge technology company, it is impossible to build resilient, secure, and flexible computer systems without the ability to apply operating systems concepts in a variety of settings. This book examines the both the principles and practice of modern operating systems, taking important, high-level concepts all the way down to the level of working code. Because operating systems concepts are among the most difficult in computer science, this top to bottom approach is the only way to really understand and master this important material.

RESTful Web Services - Leonard Richardson 2008-12-17
"Every developer working with the Web needs to read this book." -- David Heinemeier Hansson, creator of the Rails framework "RESTful Web Services finally provides a practical roadmap for constructing services that embrace the Web, instead of trying to route around it." -- Adam Trachtenberg, PHP author and EBay Web Services Evangelist You've built web sites that can be used by humans. But can you also build web sites that are usable by machines? That's where the future lies, and that's what RESTful Web Services shows you how to do. The World Wide Web is the most popular distributed application in history, and Web services and mashups have turned it into a powerful distributed computing platform. But today's web service technologies have lost sight of the simplicity that made the Web successful. They don't work like the Web, and they're missing out on its advantages. This book puts the "Web" back into web services. It shows how you can connect to the programmable web with the technologies you already use every day. The key is REST, the architectural style that drives the Web. This book: Emphasizes the

power of basic Web technologies -- the HTTP application protocol, the URI naming standard, and the XML markup language Introduces the Resource-Oriented Architecture (ROA), a common-sense set of rules for designing RESTful web services Shows how a RESTful design is simpler, more versatile, and more scalable than a design based on Remote Procedure Calls (RPC) Includes real-world examples of RESTful web services, like Amazon's Simple Storage Service and the Atom Publishing Protocol Discusses web service clients for popular programming languages Shows how to implement RESTful services in three popular frameworks -- Ruby on Rails, Restlet (for Java), and Django (for Python) Focuses on practical issues: how to design and implement RESTful web services and clients This is the first book that applies the REST design philosophy to real web services. It sets down the best practices you need to make your design a success, and the techniques you need to turn your design into working code. You can harness the power of the Web for programmable applications: you just have to work with the Web instead of against it. This book shows you how.
The Implementation of Functional Programming Languages - Simon L. Peyton Jones 1987

Collaborative Software Engineering - Ivan Mistrík 2010-03-10
Collaboration among individuals – from users to developers – is central to modern software engineering. It takes many forms: joint activity to solve common problems, negotiation to resolve conflicts, creation of shared definitions, and both social and technical perspectives impacting all software development activity. The difficulties of collaboration are also well documented. The grand challenge is not only to ensure that developers in a team deliver effectively as individuals, but that the whole team delivers more than just the sum of its parts. The editors of this book have assembled an impressive selection of authors, who have contributed to an authoritative body of work tackling a wide range of issues in the field of collaborative software engineering. The resulting volume is divided into four parts, preceded by a general editorial chapter providing a more detailed review of the domain of collaborative software engineering. Part 1 is on "Characterizing Collaborative Software Engineering", Part 2 examines various "Tools and Techniques", Part 3 addresses organizational issues, and finally Part 4 contains four examples of "Emerging Issues in Collaborative Software Engineering". As a result, this book delivers a comprehensive state-of-the-art overview and empirical results for researchers in academia and industry in areas like software process management, empirical software engineering, and global software development. Practitioners working in this area will also appreciate the detailed descriptions and reports which can often be used as guidelines to improve their daily work.
Encyclopedia of Information Science and Technology - Mehdi Khosrow-Pour 2009
"This set of books represents a detailed compendium of authoritative, research-based entries that define the contemporary state of knowledge on technology"--Provided by publisher.
Venture Deals - Jason Mendelson 2011-07-05
An engaging guide to excelling in today's venture capital arena Beginning in 2005, Brad Feld and Jason Mendelson, managing directors at Foundry Group, wrote a long series of blog posts describing all the parts of a typical venture capital Term Sheet: a document which outlines key financial and other terms of a proposed investment. Since this time, they've seen the series used as the basis for a number of college courses, and have been thanked by thousands of people who have used the information to gain a better understanding of the venture capital field. Drawn from the past work Feld and Mendelson have written about in their blog and augmented with newer material, Venture Capital Financings puts this discipline in perspective and lays out the strategies that allow entrepreneurs to excel in their start-up companies. Page by

page, this book discusses all facets of the venture capital fundraising process. Along the way, Feld and Mendelson touch on everything from how valuations are set to what externalities venture capitalists face that factor into entrepreneurs' businesses. Includes a breakdown analysis of the mechanics of a Term Sheet and the tactics needed to negotiate Details the different stages of the venture capital process, from starting a venture and seeing it through to the later stages Explores the entire venture capital ecosystem including those who invest in venture capitalist Contain standard documents that are used in these transactions Written by two highly regarded experts in the world of venture capital The venture capital arena is a complex and competitive place, but with this book as your guide, you'll discover what it takes to make your way through it.

Software Engineering - Hans van Vliet 2000-10-10
This work aims to provide the reader with sound engineering principles, whilst embracing relevant industry practices and technologies, such as object orientation and requirements engineering. It includes a chapter on software architectures, covering software design patterns.

Agile Processes in Software Engineering and Extreme Programming - Viktoria Stray 2020-01-01
This open access book constitutes the proceedings of the 21st International Conference on Agile Software Development, XP 2020, which was planned to be held during June 8-12, 2020, at the IT University of Copenhagen, Denmark. However, due to the COVID-19 pandemic the conference was postponed until an undetermined date. XP is the premier agile software development conference combining research and practice. It is a hybrid forum where agile researchers, academics, practitioners, thought leaders, coaches, and trainers get together to present and discuss their most recent innovations, research results, experiences, concerns, challenges, and trends. Following this history, for both researchers and seasoned practitioners XP 2020 provided an informal environment to network, share, and discover trends in Agile for the next 20 years. The 14 full and 2 short papers presented in this volume were carefully reviewed and selected from 37 submissions. They were organized in topical sections named: agile adoption; agile practices; large-scale agile; the business of agile; and agile and testing.

Software Testing Foundations - Andreas Spillner 2014-03-19
Professional testing of software is an essential task that requires a profound knowledge of testing techniques. The International Software Testing Qualifications Board (ISTQB) has developed a universally accepted, international qualification scheme aimed at software and system testing professionals, and has created the Syllabi and Tests for the "Certified Tester." Today about 300,000 people have taken the ISTQB certification exams. The authors of Software Testing Foundations, 4th Edition, are among the creators of the Certified Tester Syllabus and are currently active in the ISTQB. This thoroughly revised and updated fourth edition covers the "Foundations Level" (entry level) and teaches the most important methods of software testing. It is designed for self-study and provides the information necessary to pass the Certified Tester-Foundations Level exam, version 2011, as defined by the ISTQB. Also in this new edition, technical terms have been precisely stated according to the recently revised and updated ISTQB glossary. Topics covered: Fundamentals of Testing Testing and the Software Lifecycle Static and Dynamic Testing Techniques Test Management Test Tools Also mentioned are some updates to the syllabus that are due in 2015.

Algorithms and Data Structures for External Memory - Jeffrey Scott Vitter 2008
Describes several useful paradigms for the design and implementation of efficient external memory (EM) algorithms and data structures. The problem domains considered include sorting, permuting, FFT, scientific computing, computational geometry, graphs, databases, geographic information systems, and text and string processing.

Theories of the Information Society - Frank Webster 2002
Popular opinion suggests that information has become a distinguishing feature of the modern world. Where once economies were built on industry and conquest, we are now instead said to be part of a global information economy. In this new and thoroughly revised edition of his popular book, author Webster brings his work up-to-date both with new theoretical work and with social and technological changes - such as the rapid growth of the internet and accelerated globalization - and reassesses the work of key theorists in light of these changes. This book is essential reading for students of contemporary social theory and anybody interested in social and technological change in the post-war era.

Methods and Applications for Advancing Distance Education

Technologies: International Issues and Solutions - Syed, Mahbubur Rahman 2009-04-30
Provides communication technologies, intelligent technologies, and quality educational pedagogy for advancing distance education for both teaching and learning.

Object-Oriented Software Engineering: An Agile Unified Methodology - David Kung 2013-01-25
Object-Oriented Software Engineering: An Agile Unified Methodology by David Kung presents a step-by-step methodology that integrates modeling and design, UML, patterns, test-driven development, quality assurance, configuration management, and agile principles throughout the life cycle. The overall approach is casual and easy to follow, with many practical examples that show the theory at work. The author uses his experiences as well as real-world stories to help the reader understand software design principles, patterns, and other software engineering concepts. The book also provides stimulating exercises that go far beyond the type of question that can be answered by simply copying portions of the text.

The Future of Computing Performance - National Research Council 2011-04-21
The end of dramatic exponential growth in single-processor performance marks the end of the dominance of the single microprocessor in computing. The era of sequential computing must give way to a new era in which parallelism is at the forefront. Although important scientific and engineering challenges lie ahead, this is an opportune time for innovation in programming systems and computing architectures. We have already begun to see diversity in computer designs to optimize for such considerations as power and throughput. The next generation of discoveries is likely to require advances at both the hardware and software levels of computing systems. There is no guarantee that we can make parallel computing as common and easy to use as yesterday's sequential single-processor computer systems, but unless we aggressively pursue efforts suggested by the recommendations in this book, it will be "game over" for growth in computing performance. If parallel programming and related software efforts fail to become widespread, the development of exciting new applications that drive the computer industry will stall; if such innovation stalls, many other parts of the economy will follow suit. The Future of Computing Performance describes the factors that have led to the future limitations on growth for single processors that are based on complementary metal oxide semiconductor (CMOS) technology. It explores challenges inherent in parallel computing and architecture, including ever-increasing power consumption and the escalated requirements for heat dissipation. The book delineates a research, practice, and education agenda to help overcome these challenges. The Future of Computing Performance will guide researchers, manufacturers, and information technology professionals in the right direction for sustainable growth in computer performance, so that we may all enjoy the next level of benefits to society.

Eat Pray Love - Elizabeth Gilbert 2010-06-29
Traces the author's decision to quit her job and travel the world for a year after suffering a midlife crisis and divorce, an endeavor that took her to three places in her quest to explore her own nature, experience fulfillment and learn the art of spiritual balance. (Biography & autobiography). Reissue. A best-selling book. Movie tie-in.

Systems Analysis and Design in a Changing World - John W. Satzinger 2015-02-01
Refined and streamlined, SYSTEMS ANALYSIS AND DESIGN IN A CHANGING WORLD, 7E helps students develop the conceptual, technical, and managerial foundations for systems analysis design and implementation as well as project management principles for systems development. Using case driven techniques, the succinct 14-chapter text focuses on content that is key for success in today's market. The authors' highly effective presentation teaches both traditional (structured) and object-oriented (OO) approaches to systems analysis and design. The book highlights use cases, use diagrams, and use case descriptions required for a modeling approach, while demonstrating their application to traditional, web development, object-oriented, and service-oriented architecture approaches. The Seventh Edition's refined sequence of topics makes it easier to read and understand than ever. Regrouped analysis and design chapters provide more flexibility in course organization. Additionally, the text's running cases have been completely updated and now include a stronger focus on connectivity in applications. Important Notice: Media content referenced within the product description or the product text may not be available in the ebook

version.

Agile Software Development - Alistair Cockburn 2006-10-19
"Agile Software Development is a highly stimulating and rich book. The author has a deep background and gives us a tour de force of the emerging agile methods." —Tom Gilb The agile model of software development has taken the world by storm. Now, in Agile Software Development, Second Edition, one of agile's leading pioneers updates his Jolt Productivity award-winning book to reflect all that's been learned about agile development since its original introduction. Alistair Cockburn begins by updating his powerful model of software development as a "cooperative game of invention and communication." Among the new ideas he introduces: harnessing competition without damaging collaboration; learning lessons from lean manufacturing; and balancing strategies for communication. Cockburn also explains how the cooperative game is played in business and on engineering projects, not just software development Next, he systematically illuminates the agile model, shows how it has evolved, and answers the questions developers and project managers ask most often, including · Where does agile development fit in our organization? · How do we blend agile ideas with other ideas? · How do we extend agile ideas more broadly? Cockburn takes on crucial misconceptions that cause agile projects to fail. For example, you'll learn why encoding project management strategies into fixed processes can lead to ineffective strategy decisions and costly mistakes. You'll also find a thoughtful discussion of the controversial relationship between agile methods and user experience design. Cockburn turns to the practical challenges of constructing agile methodologies for your own teams. You'll learn how to tune and continuously reinvent your methodologies, and how to manage incomplete communication. This edition contains important new contributions on these and other topics: · Agile and CMMI · Introducing agile from the top down · Revisiting "custom contracts" · Creating change with "stickers" In addition, Cockburn updates his discussion of the Crystal methodologies, which utilize his "cooperative game" as their central metaphor. If you're new to agile development, this book will help you succeed the first time out. If you've used agile methods before, Cockburn's techniques will make you even more effective.

Testing Object-oriented Systems - Robert Binder 2000
More than ever, mission-critical and business-critical applications depend on object-oriented (OO) software. Testing techniques tailored to the unique challenges of OO technology are necessary to achieve high reliability and quality. "Testing Object-Oriented Systems: Models, Patterns, and Tools" is an authoritative guide to designing and automating test suites for OO applications. This comprehensive book explains why testing must be model-based and provides in-depth coverage of techniques to develop testable models from state machines, combinational logic, and the Unified Modeling Language (UML). It introduces the test design pattern and presents 37 patterns that explain how to design responsibility-based test suites, how to tailor integration and regression testing for OO code, how to test reusable components and frameworks, and how to develop highly effective test suites from use cases. Effective testing must be automated and must leverage object technology. The author describes how to design and code specification-based assertions to offset testability losses due to inheritance and polymorphism. Fifteen micro-patterns present oracle strategies--practical solutions for one of the hardest problems in test design. Seventeen design patterns explain how to automate your test suites with a coherent OO test harness framework. The author provides thorough coverage of testing issues such as: The bug hazards of OO programming and differences from testing procedural code How to design responsibility-based tests for classes, clusters, and subsystems using class invariants, interface data flow models, hierarchic state machines, class associations, and scenario analysis How to support reuse by effective testing of abstract classes, generic classes, components, and frameworks How to choose an integration strategy that supports iterative and incremental development How to achieve comprehensive system testing with testable use cases How to choose a regression test approach How to develop expected test results and evaluate the post-test state of an object How to automate testing with assertions, OO test drivers, stubs, and test frameworks Real-world experience, world-class best practices, and the latest research in object-oriented testing are included. Practical examples illustrate test design and test automation for Ada 95, C++, Eiffel, Java, Objective-C, and Smalltalk. The UML is used throughout, but the test design patterns apply to systems developed with any OO language or methodology. 0201809389B04062001

OBJECT-ORIENTED SOFTWARE ENGINEERING - YOGESH SINGH

2012-03-05
This comprehensive and well-written book presents the fundamentals of object-oriented software engineering and discusses the recent technological developments in the field. It focuses on object-oriented software engineering in the context of an overall effort to present object-oriented concepts, techniques and models that can be applied in software estimation, analysis, design, testing and quality improvement. It applies unified modelling language notations to a series of examples with a real-life case study. The example-oriented approach followed in this book will help the readers in understanding and applying the concepts of object-oriented software engineering quickly and easily in various application domains. This book is designed for the undergraduate and postgraduate students of computer science and engineering, computer applications, and information technology. KEY FEATURES : Provides the foundation and important concepts of object-oriented paradigm. Presents traditional and object-oriented software development life cycle models with a special focus on Rational Unified Process model. Addresses important issues of improving software quality and measuring various object-oriented constructs using object-oriented metrics. Presents numerous diagrams to illustrate object-oriented software engineering models and concepts. Includes a large number of solved examples, chapter-end review questions and multiple choice questions along with their answers.

Computational Science and Its Applications -- ICCSA 2013 - Beniamino Murgante 2013-06-22
The five-volume set LNCS 7971-7975 constitutes the refereed proceedings of the 13th International Conference on Computational Science and Its Applications, ICCSA 2013, held in Ho Chi Minh City, Vietnam in June 2013. The 248 revised papers presented in five tracks and 33 special sessions and workshops were carefully reviewed and selected. The 46 papers included in the five general tracks are organized in the following topical sections: computational methods, algorithms and scientific applications; high-performance computing and networks; geometric modeling, graphics and visualization; advanced and emerging applications; and information systems and technologies. The 202 papers presented in special sessions and workshops cover a wide range of topics in computational sciences ranging from computational science technologies to specific areas of computational sciences such as computer graphics and virtual reality.

Converging Technologies for Improving Human Performance - Mihail C. Roco 2013-04-17
M. C. Roco and W.S. Bainbridge In the early decades of the 21st century, concentrated efforts can unify science based on the unity of nature, thereby advancing the combination of nanotechnology, biotechnology, information technology, and new technologies based in cognitive science. With proper attention to ethical issues and societal needs, converging in human abilities, societal technologies could achieve a tremendous improvement outcomes, the nation's productivity, and the quality of life. This is a broad, cross cutting, emerging and timely opportunity of interest to individuals, society and humanity in the long term. The phrase "convergent technologies" refers to the synergistic combination of four major "NBIC" (nano-bio-info-cogno) provinces of science and technology, each of which is currently progressing at a rapid rate: (a) nanoscience and nanotechnology; (b) biotechnology and biomedicine, including genetic engineering; (c) information technology, including advanced computing and communications; (d) cognitive science, including cognitive neuroscience. Timely and Broad Opportunity. Convergence of diverse technologies is based on material unity at the nanoscale and on technology integration from that scale.

The Agile Samurai - Jonathan Rasmusson 2010-09-25
Printed in full color. Faced with a software project of epic proportions? Tired of over-committing and under-delivering? Enter the dojo of the agile samurai, where agile expert Jonathan Rasmusson shows you how to kick-start, execute, and deliver your agile projects. Combining cutting-edge tools with classic agile practices, The Agile Samurai gives you everything you need to deliver something of value every week and make rolling your software into production a non-event. Get ready to kick some software project butt. By learning the ways of the agile samurai you will discover: how to create plans and schedules your customer and your team can believe in what characteristics make a good agile team and how to form your own how to gather requirements in a fraction of the time using agile user stories what to do when you discover your schedule is wrong, and how to look like a pro correcting it how to execute fiercely by leveraging the power of agile software engineering practices By the end of this book you will know everything you need to set up, execute, and successfully deliver agile projects, and have fun along the way. If

you're a project lead, this book gives you the tools to set up and lead your agile project from start to finish. If you are an analyst, programmer, tester, usability designer, or project manager, this book gives you the insight and foundation necessary to become a valuable agile team member. The Agile Samurai slices away the fluff and theory that make other books less-than-agile. It's packed with best practices, war stories, plenty of humor and hands-on tutorial exercises that will get you doing the right things, the right way. This book will make a difference.

Integrated Design and Manufacturing in Mechanical Engineering - Patrick Chedmail 2012-12-06
This volume contains the selected papers of the first I.D.M.M.E. conference on 'Integrated Design and Manufacturing in Mechanical Engineering', held in Nantes from 15-17 April 1996. Its objective was to discuss the questions related to the definition of the optimal design and manufacturing processes and to their integration through coherent methodologies in adapted environments. The initiative of the Conference and the organization thereof, is mainly due to the efforts of the french PRIMECA group (Pool of Computer Resources for Mechanics) started eight years ago. We were able to attract the internationru community with the support of the International Institution for Production Engineering Research (C.I.R.P.). The conference brought together two hundred and fifty specialists from around the world. About ninety papers and twenty posters were presented covering three main topics : optimization and evaluation of the product design process, optimization and evaluation of the manufacturing systems and methodological aspects.

Conceptual Modeling of Information Systems - Antoni Olivé 2007-08-15
This brilliant textbook explains in detail the principles of conceptual modeling independently from particular methods and languages and shows how to apply them in real-world projects. The author covers all aspects of the engineering process from structural modeling over behavioral modeling to meta-modeling, and completes the presentation with an extensive case study based on the osCommerce system. Written for computer science students in classes on information systems modeling as well as for professionals feeling the need to formalize their experiences or to update their knowledge, Olivé delivers here a comprehensive treatment of all aspects of the modeling process. His book is complemented by lots of exercises and additional online teaching material.

Information Systems Engineering - Arne Soelvberg 2012-12-06
This book presents a selection of subjects which the authors deem to be important for information systems engineers. The book is intended for introductory teaching. We have tried to write the book in such a way that students with only fragmented knowledge of computers are able to read the book without too many difficulties. Students who have had only an intro ductory course in computer programming should be able to read most of the book. We have tried to achieve simplicity without compromising on depth in our discussions of the various aspects of information systems engineering. So it is our hope that also those who have deeper knowledge in computing may find pleasure in reading parts of the book. The writing of a textbook is a major undertaking for its authors. One is quite often forced to reexamine truisms in the subject area, and must be prepared to reevaluate one's opinions and priorities as one learns more. In particular this is so in new fields, where formalisms have been scarcely used, and where consensus has not yet emerged either on what constitutes the subject area or on how practical problems within the field shall be approached. Contemporary practice in computer applications is confronted with an increasingly complex world, both in a technical sense and in the complexity of problems that are solved by computer.

Object-Oriented and Classical Software Engineering - Stephen R. Schach 2001-11
Classical and Object-Oriented Software Engineering, 5/e is designed for an introductory software engineering course. This book provides an excellent introduction to software engineering fundamentals, covering both traditional and object-oriented techniques.Schach's unique organization and style makes it excellent for use in a classroom setting. It presents the underlying software engineering theory in Part I and follows it up with the more practical life-cycle material in Part II. Many software engineering books are more like reference books, which do not provide the appropriate fundamentals before inundating students with implementation details.In this edition, more practical material has been added to help students understand how to use what they are learning. This has been done through the use of "How To" boxes and greater implementation detail in the case study. Additionally, the new edition contains the references to the most current literature and includes an overview of extreme programmming.The website in this edition will be more extensive. It will include Solutions, PowerPoints that incorporate lecture notes, newly developed self-quiz questions, and source code for the term project and case study.

Object-oriented Software Engineering - David C. Kung 2013-02
Presents a step-by-step methodology that integrates modeling and design, UML, patterns, test-driven development, quality assurance, configuration management, and agile principles throughout the life cycle. This book provides stimulating exercises that go far beyond the type of question that can be answered by simply copying portions of the text.

Technological Slavery (Large Print 16pt) - Theodore J. Kaczynski 2011-02
Theodore Kaczynski saw violent collapse as the only way to bring down the techno-industrial system, and in more than a decade of mail bomb terror he killed three people and injured 23 others. One does not need to support the actions that landed Kaczynski in supermax prison to see the value of his essays disabusing the notion of heroic technology while revealing the manner in which it is destroying the planet. For the first time, readers will have an uncensored personal account of his anti-technology philosophy, including a corrected version of the notorious ''Unabomber Manifesto,''Kaczynski, s critique of anarcho-primitivism, and essays regarding ''the Coming Revolution.''

Game Architecture and Design - Andrew Rollings 2004
A guide to computer game design, architecture, and management explores the application of design principles, shares the experiences of game programmers, and offers an overview of game development software.

Testing Object-Oriented Software - David C. Kung 1998-11-10
Object-oriented programming increases software reusability, extensibility, interoperability, and reliability. Software testing is necessary to realize these benefits by uncovering as many programming errors as possible at a minimum cost. A major challenge to the software engineering community remains how to reduce the cost while improving the quality of software testing. The requirements for testing object-oriented programs differ from those for testing conventional programs. Testing Object-Oriented Software illustrates these differences and discusses object-oriented software testing problems, focusing on the difficulties and challenges testers face. The text contains of nineteen reprinted papers providing a general framework for class- and system-level testing and examines object-oriented design criteria and high testability metrics. It offers object-oriented testing techniques, ideas and methods for unit testing, and object-oriented program integration-testing strategy. Readers are shown how to drastically reduce regression test costs, presented with steps for object-oriented testing, and introduced to object-oriented test tools and systems. The book's intended audience includes object-oriented program testers, program developers, software project managers, and researchers working with object-oriented testing.

Software Testing Techniques - Boris Beizer 2003
Software Testing Techniques, 2nd Edition is the first book-length work that explicitly addresses the idea that design for testability is as important as testing itself not just by saying that testability is a desirable goal, but by showing the reader how it to do it. Every chapter has testability guidelines that illustrate how the technique discussed in the chapter can be used to make software more easily tested and therefore more reliable and maintainable. Application of all techniques to unit, integration, maintenance, and system testing are discussed throughout this book.As a self-study text, as a classroom text, as a working reference, it is a book that no programmer, independent software tester, software engineer, testing theorist, system designer, or software project manager can be without.

The Tipping Point - Malcolm Gladwell 2006-11-01
From the bestselling author of The Bomber Mafia: discover Malcolm Gladwell's breakthrough debut and explore the science behind viral trends in business, marketing, and human behavior. The tipping point is that magic moment when an idea, trend, or social behavior crosses a threshold, tips, and spreads like wildfire. Just as a single sick person can start an epidemic of the flu, so too can a small but precisely targeted push cause a fashion trend, the popularity of a new product, or a drop in the crime rate. This widely acclaimed bestseller, in which Malcolm Gladwell explores and brilliantly illuminates the tipping point phenomenon, is already changing the way people throughout the world think about selling products and disseminating ideas. "A wonderful page-turner about a fascinating idea that should affect the way every thinking person looks at the world." —Michael Lewis

Testing Object-Oriented Software - David C. Kung 1998-11-10

Object-oriented programming increases software reusability, extensibility, interoperability, and reliability. Software testing is necessary to realize these benefits. Software testing aims to uncover as many programming errors as possible at a minimum cost. A major challenge to the software engineering community remains how to reduce the cost and improve the quality of software testing. The requirements for testing object-oriented programs differ from those for testing conventional programs. Testing Object-Oriented Software illustrates these differences and discusses object-oriented software testing problems, focusing on the difficulties and challenges testers face. The book provides a general framework for class- and system-level testing and examines object-oriented design criteria and high testability metrics. It offers object-oriented testing techniques, ideas and methods for unit testing, and object-oriented program integration-testing strategy. Readers are shown how they can drastically reduce regression test costs, presented with steps for object-oriented testing, and introduced to object-oriented test tools and systems. In addition to software testing problems, the text covers various test methods developers can use during the design phase to generate programs with good testability. The book's intended audience includes object-oriented program testers, program developers, software project managers, and researchers working with object-oriented testing.

Research Anthology on Recent Trends, Tools, and Implications of Computer Programming - Management Association, Information Resources 2020-08-03
Programming has become a significant part of connecting theoretical development and scientific application computation. Computer programs and processes that take into account the goals and needs of the user meet with the greatest success, so it behooves software engineers to consider the human element inherent in every line of code they write. Research Anthology on Recent Trends, Tools, and Implications of Computer Programming is a vital reference source that examines the latest scholarly material on trends, techniques, and uses of various programming applications and examines the benefits and challenges of these computational developments. Highlighting a range of topics such as coding standards, software engineering, and computer systems development, this multi-volume book is ideally designed for programmers, computer scientists, software developers, analysts, security experts, IoT software programmers, computer and software engineers, students, professionals, and researchers.

Ikigai - Héctor García 2017-08-29
INTERNATIONAL BESTSELLER • 1.5 MILLION+ COPIES SOLD WORLDWIDE "Workers looking for more fulfilling positions should start by identifying their ikigai." —Business Insider "One of the unintended—yet positive—consequences of the [pandemic] is that it is forcing people to reevaluate their jobs, careers, and lives. Use this time wisely, find your personal ikigai, and live your best life." —Forbes Find your ikigai (pronounced ee-key-guy) to live longer and bring more meaning and joy to all your days. "Only staying active will make you want to live a hundred years." —Japanese proverb According to the Japanese, everyone has an ikigai—a reason for living. And according to the residents of the Japanese village with the world's longest-living people, finding it is the key to a happier and longer life. Having a strong sense of ikigai—where what you love, what you're good at, what you can get paid for, and what the world needs all overlap—means that each day is infused with meaning. It's the reason we get up in the morning. It's also the reason many Japanese never really retire (in fact there's no word in Japanese that means retire in the sense it does in English): They remain active and work at what they enjoy, because they've found a real purpose in life—the happiness of always being busy. In researching this book, the authors interviewed the residents of the Japanese village with the highest percentage of 100-year-olds—one of the world's Blue Zones. Ikigai reveals the secrets to their longevity and happiness: how they eat, how they move, how they work, how they foster collaboration and community, and—their best-kept secret—how they find the ikigai that brings satisfaction to their lives. And it provides practical tools to help you discover your own ikigai. Because who doesn't want to find happiness in every day?

Advances in Computing and Communications, Part II - Ajith Abraham 2011-07-08
This volume is the second part of a four-volume set (CCIS 190, CCIS 191, CCIS 192, CCIS 193), which constitutes the refereed proceedings of the First International Conference on Computing and Communications, ACC 2011, held in Kochi, India, in July 2011. The 72 revised full papers presented in this volume were carefully reviewed and selected from a large number of submissions. The papers are organized in topical sections on database and information systems; distributed software development; human computer interaction and interface; ICT; internet and Web computing; mobile computing; multi agent systems; multimedia and video systems; parallel and distributed algorithms; security, trust and privacy.

Object - Oriented Modeling And Design With Uml, 2/E - Blaha 2007-09
The revision offers a crisp, clear explanation of the basics of object-oriented thinking via UML models, then presents a process for applying these principles to software development, including C++, Java, and relational databases. An integrated case study threads throughout the book, illustrating key ideas as well as their application.

Facts and Fallacies of Software Engineering - Robert L. Glass 2003
Regarding the controversial and thought-provoking assessments in this handbook, many software professionals might disagree with the authors, but all will embrace the debate. Glass identifies many of the key problems hampering success in this field. Each fact is supported by insightful discussion and detailed references.