

Computer Science An Overview J Glenn Brookshear

The user-friendly, object-oriented programming language Python is quickly becoming the most popular introductory programming language for both students and instructors. This updated Second Edition of Python Programming in Context provides a comprehensive, accessible introduction to Python fundamentals. An ideal first language for learners entering the rapidly expanding field of computer science, Python gives students a solid platform of key problem-solving skills that translate easily across programming languages. Building on essential concepts of computer science, and offering a plenitude of real-world examples, Python Programming in Context, Second Edition offers a thorough overview of multiple applied areas, including image processing, cryptography, astronomy, the Internet, and bioinformatics. The text's emphasis on problem-solving, extrapolation, and development of independent exploration and solution-building provides students with a unique and innovative approach to learning programming. Python Programming in Context, Second Edition is the ideal introductory text for those delving into computer programming. Key Features - Utilizes Python 3 - Provides a clear, accessible, and skill-focused approach to programming with Python - Contains problem sets based on real-world examples and problem-solving rather than language features - Offers a variety of exercises that develop independent skill-building and exploration - Every new copy of the text is packaged with full student access to Turing's Craft Custom CodeLab. Customized to match the organization of the text, CodeLab offers students hands-on Python programming experience with immediate feedback. - Accompanied by a full suite of instructor support material, including solutions to the exercises in the text, downloadable source code, PowerPoint Lecture Outlines, and a complete Test Bank.

When digital content and technologies are designed in a way that is inaccessible for persons with disabilities, they are locked out of commerce, education, employment, and access to government information. In developing areas of the world, as new technical infrastructures are being built, it is especially important to ensure that accessibility is a key design goal. Unfortunately, nearly all research on Information and Communication Technology (ICT) accessibility and innovation for persons with disabilities-whether from the legal, technical, or development fields-has focused on developed countries, with very little being written about developing world initiatives. Accessible Technology and the Developing World aims to change this, by bringing increased attention to ICT accessibility in developing areas. This book brings together a unique combination of contributors with diverse disciplinary backgrounds, including authors from well-known non-governmental organizations, significant United Nations entities, and universities in both the developing and developed world. Together, they present a unique and much needed review of this critical and growing area of work, and primarily address three core themes - the lack of attention given to innovations taking place in the developing world, the need to ensure that infrastructures in the Global South do not present barriers to people with disabilities, and the need to exercise caution when applying techniques from the Global North to the Global South that won't transfer effectively. This book will be of use to researchers in the fields of civil rights, development studies, disability rights, disability studies, human-computer interaction and accessibility, human rights, international law, political science, and universal design.

This book is appropriate for both majors of computer science and students of other disciplines."--BOOK JACKET.

For Introduction to Computer Science courses. Surveys the breadth of computer science--with the depth needed to explore concepts Computer Science: An Overview is written for students of computer science as well as students from other disciplines. Its broad coverage and clear exposition are accessible to students from all backgrounds, encouraging a practical and realistic understanding of the subject. Written to provide students with a bottom-up, concrete-to-abstract foundation, this broad background exposes beginning computer science students to the breadth of the subject in which they are planning to major, and students from other disciplines to what they need to relate to the technical society in which they live. Individual chapters are independent, and can be covered in an order that suits instructor course needs with selected content marked as optional for the introductory course. With a new full-color design, each chapter in the 13th Edition has seen revisions, updates, and corrections from the previous editions. The text also continues to use Python to provide programming tools for exploration and experimentation. More than 1,000 questions and exercises, Chapter Review Problems, and Social Issues questions reinforce core concepts. The text's Companion Website extends resources to enhance the course.

Over the past sixty years, the spectacular growth of the technologies associated with the computer is visible for all to see and experience. Yet, the science underpinning this technology is less visible and little understood outside the professional computer science community. As a scientific discipline, computer science stands alongside the likes of molecular biology and cognitive science as one of the most significant new sciences of the post Second World War era. In this Very Short Introduction, Subrata Dasgupta sheds light on these lesser known areas and considers the conceptual basis of computer science. Discussing algorithms, programming, and sequential and parallel processing, he considers emerging modern ideas such as biological computing and cognitive modelling, challenging the idea of computer science as a science of the artificial. ABOUT THE SERIES: The Very Short Introductions series from Oxford University Press contains hundreds of titles in almost every subject area. These pocket-sized books are the perfect way to get ahead in a new subject quickly. Our expert authors combine facts, analysis, perspective, new ideas, and enthusiasm to make interesting and challenging topics highly readable.

The first edition won the award for Best 1990 Professional and Scholarly Book in Computer Science and Data Processing by the Association of American Publishers. There are books on algorithms that are rigorous but incomplete and others that cover masses of material but lack rigor. Introduction to Algorithms combines rigor and comprehensiveness. The book covers a broad range of algorithms in depth, yet makes their design and analysis accessible to all levels of readers. Each chapter is relatively self-contained and can be used as a unit of study. The algorithms are described in English and in a pseudocode designed to be readable by anyone who has done a little programming. The explanations have been kept elementary without sacrificing depth of coverage or mathematical rigor. The first edition became the standard reference for professionals and a widely used text in universities worldwide. The second edition features new chapters on the role of algorithms, probabilistic analysis and randomized algorithms, and linear programming, as well as extensive revisions to virtually every section of the book. In a subtle but important change, loop invariants are introduced early and used throughout the text to prove algorithm correctness. Without changing the mathematical and analytic focus, the authors have moved much of the mathematical foundations material from Part I to an appendix and have included additional motivational material at the beginning.

First-ever comprehensive introduction to the major new subject of quantum computing and quantum information.

Computational science is an exciting new field at the intersection of the sciences, computer science, and mathematics because much scientific investigation now involves computing as well as theory and experiment. This textbook provides students with a versatile and accessible introduction to the subject. It assumes only a background in high school algebra, enables instructors to follow tailored pathways through the material, and is the only textbook of its kind designed specifically for an introductory course in the computational science and engineering curriculum. While the text itself is generic, an accompanying website offers tutorials and files in a variety of software packages. This fully updated and expanded edition features two new chapters on agent-based simulations and modeling with matrices, ten new project modules, and an additional module on diffusion. Besides increased treatment of high-performance computing and its applications, the book also includes additional quick review questions with answers, exercises, and individual and team projects. The only introductory textbook of its kind—now fully updated and expanded Features two new chapters on agent-based simulations and modeling with matrices Increased coverage of high-performance computing and its applications Includes additional modules, review questions, exercises, and projects An online instructor's manual with exercise answers, selected project solutions, and a test bank and solutions (available only to professors) An online illustration package is available to professors

The new edition of an introductory text that teaches students the art of computational problem solving, covering topics ranging from simple algorithms to information visualization. This book introduces students with little or no prior programming experience to the art of computational problem solving using Python and various Python libraries, including PyLab. It provides students with skills that will enable them to make productive use of computational techniques, including some of the tools and techniques of data science for using computation to model and interpret data. The book is based on an MIT course (which became the most popular course offered through MIT's OpenCourseWare) and was developed for use not only in a conventional classroom but in in a massive open online course (MOOC). This new edition has been updated for Python 3, reorganized to make it easier to use for courses that cover only a subset of the material, and offers additional material including five new chapters. Students are introduced to Python and the basics of programming in the context of such computational concepts and techniques as exhaustive enumeration, bisection search, and efficient approximation algorithms. Although it covers such traditional topics as computational complexity and simple algorithms, the book focuses on a wide range of topics not found in most introductory texts, including information visualization, simulations to model randomness, computational techniques to understand data, and statistical techniques that inform (and misinform) as well as two related but relatively advanced topics: optimization problems and dynamic programming. This edition offers expanded material on statistics and machine learning and new chapters on Frequentist and Bayesian statistics.

Completely revised and updated, Computer Systems, Fourth Edition offers a clear, detailed, step-by-step introduction to the central concepts in computer organization, assembly language, and computer architecture. Important Notice: The digital edition of this book is missing some of the images or content found in the physical edition.

This book provides an overview of how to approach computer science education research from a pragmatic perspective. It represents the diversity of traditions and approaches inherent in this interdisciplinary area, while also providing a structure within which to make sense of that diversity. It provides multiple 'entry points'- to literature, to methods, to topics Part One, 'The Field and the Endeavor', frames the nature and conduct of research in computer science education. Part Two, 'Perspectives and Approaches', provides a number of grounded chapters on particular topics or themes, written by experts in each domain. These chapters cover the following topics: * design * novice misconceptions * programming environments for novices * algorithm visualisation * a schema theory view on learning to program * critical theory as a theoretical approach to computer science education research Juxtaposed and taken together, these chapters indicate just how varied the perspectives and research approaches can be. These chapters, too, act as entry points, with illustrations drawn from published work.

A new framework for understanding computing: a coherent set of principles spanning technologies, domains, algorithms, architectures, and designs. Computing is usually viewed as a technology field that advances at the breakneck speed of Moore's Law. If we turn away even for a moment, we might miss a game-changing technological breakthrough or an earthshaking theoretical development. This book takes a different perspective, presenting computing as a science governed by fundamental principles that span all technologies. Computer science is a science of information processes. We need a new language to describe the science, and in this book Peter Denning and Craig Martell offer the great principles framework as just such a language. This is a book about the whole of computing—its algorithms, architectures, and designs. Denning and Martell divide the great principles of computing into six categories: communication, computation, coordination, recollection, evaluation, and design. They begin with an introduction to computing, its history, its many interactions with other fields, its domains of practice, and the structure of the great principles framework. They go on to examine the great principles in different areas: information, machines, programming, computation, memory, parallelism, queueing, and design. Finally, they apply the great principles to networking, the Internet in particular. Great Principles of Computing will be essential reading for professionals in science and engineering fields with a “computational” branch, for practitioners in computing who want overviews of less familiar areas of computer science, and for non-computer science majors who want an accessible entry way to the field.

Never HIGHLIGHT a Book Again Virtually all testable terms, concepts, persons, places, and events are included.

Cram101 Textbook Outlines gives all of the outlines, highlights, notes for your textbook with optional online practice tests. Only Cram101 Outlines are Textbook Specific. Cram101 is NOT the Textbook. Accompanys: 9780521673761

Now in its eighth edition, this book continues to provide a comprehensive, accessible, and up-to-date introduction to the dynamic field of computer science using a breadth-first approach. The table of contents and the text itself have been revised and expanded to reflect changes in the field, including the trend toward using Web and Internet Technology, the evolution of Objects, and the important growth in the field of databases. Specifically, chapter three from the previous

edition has been expanded into two chapters. Chapter three will now only cover Operating Systems and the new chapter four will focus on Networks and the Internet. Anyone interested in gaining a thorough introduction to Computer Science. Build real-world Artificial Intelligence applications with Python to intelligently interact with the world around you About This Book Step into the amazing world of intelligent apps using this comprehensive guide Enter the world of Artificial Intelligence, explore it, and create your own applications Work through simple yet insightful examples that will get you up and running with Artificial Intelligence in no time Who This Book Is For This book is for Python developers who want to build real-world Artificial Intelligence applications. This book is friendly to Python beginners, but being familiar with Python would be useful to play around with the code. It will also be useful for experienced Python programmers who are looking to use Artificial Intelligence techniques in their existing technology stacks. What You Will Learn Realize different classification and regression techniques Understand the concept of clustering and how to use it to automatically segment data See how to build an intelligent recommender system Understand logic programming and how to use it Build automatic speech recognition systems Understand the basics of heuristic search and genetic programming Develop games using Artificial Intelligence Learn how reinforcement learning works Discover how to build intelligent applications centered on images, text, and time series data See how to use deep learning algorithms and build applications based on it In Detail Artificial Intelligence is becoming increasingly relevant in the modern world where everything is driven by technology and data. It is used extensively across many fields such as search engines, image recognition, robotics, finance, and so on. We will explore various real-world scenarios in this book and you'll learn about various algorithms that can be used to build Artificial Intelligence applications. During the course of this book, you will find out how to make informed decisions about what algorithms to use in a given context. Starting from the basics of Artificial Intelligence, you will learn how to develop various building blocks using different data mining techniques. You will see how to implement different algorithms to get the best possible results, and will understand how to apply them to real-world scenarios. If you want to add an intelligence layer to any application that's based on images, text, stock market, or some other form of data, this exciting book on Artificial Intelligence will definitely be your guide! Style and approach This highly practical book will show you how to implement Artificial Intelligence. The book provides multiple examples enabling you to create smart applications to meet the needs of your organization. In every chapter, we explain an algorithm, implement it, and then build a smart application.

Discover how graph algorithms can help you leverage the relationships within your data to develop more intelligent solutions and enhance your machine learning models. You'll learn how graph analytics are uniquely suited to unfold complex structures and reveal difficult-to-find patterns lurking in your data. Whether you are trying to build dynamic network models or forecast real-world behavior, this book illustrates how graph algorithms deliver value—from finding vulnerabilities and bottlenecks to detecting communities and improving machine learning predictions. This practical book walks you through hands-on examples of how to use graph algorithms in Apache Spark and Neo4j—two of the most common choices for graph analytics. Also included: sample code and tips for over 20 practical graph algorithms that cover optimal pathfinding, importance through centrality, and community detection. Learn how graph analytics vary from conventional statistical analysis Understand how classic graph algorithms work, and how they are applied Get guidance on which algorithms to use for different types of questions Explore algorithm examples with working code and sample datasets from Spark and Neo4j See how connected feature extraction can increase machine learning accuracy and precision Walk through creating an ML workflow for link prediction combining Neo4j and Spark

A surprisingly simple way for students to master any subject--based on one of the world's most popular online courses and the bestselling book *A Mind for Numbers* *A Mind for Numbers* and its wildly popular online companion course "Learning How to Learn" have empowered more than two million learners of all ages from around the world to master subjects that they once struggled with. Fans often wish they'd discovered these learning strategies earlier and ask how they can help their kids master these skills as well. Now in this new book for kids and teens, the authors reveal how to make the most of time spent studying. We all have the tools to learn what might not seem to come naturally to us at first--the secret is to understand how the brain works so we can unlock its power. This book explains:

- Why sometimes letting your mind wander is an important part of the learning process
- How to avoid "rut think" in order to think outside the box
- Why having a poor memory can be a good thing
- The value of metaphors in developing understanding
- A simple, yet powerful, way to stop procrastinating

Filled with illustrations, application questions, and exercises, this book makes learning easy and fun.

Completely revised and updated with the latest version of C++, the new Fifth Edition of *Programming and Problem Solving with C++* provides the clearest introduction to C++, object-oriented programming, and software development available. Renowned author team Nell Dale and Chip Weems are careful to include all topics and guidelines put forth by the ACM/IEEE. A new chapter on Data Structures makes this text ideal for the one- or two-term course. New Software Maintenance Case Studies teach students how to read code in order to debug, alter, or enhance existing class or code segments. Important Notice: The digital edition of this book is missing some of the images or content found in the physical edition

A guide to the C# programming language covers such topics as object-oriented programming, creating database applications, and Windows 8 programming.

A concise introduction to the emerging field of data science, explaining its evolution, relation to machine learning, current uses, data infrastructure issues, and ethical challenges. The goal of data science is to improve decision making through the analysis of data. Today data science determines the ads we see online, the books and movies that are recommended to us online, which emails are filtered into our spam folders, and even how much we pay for health insurance. This volume in the MIT Press Essential Knowledge series offers a concise introduction to the emerging field of data science, explaining its evolution, current uses, data infrastructure issues, and ethical challenges. It has never been easier for organizations to gather, store, and process data. Use of data science is driven by the rise of big data and social media, the development of high-performance computing, and the emergence of such powerful methods for data analysis and modeling as deep learning. Data science encompasses a set of principles, problem definitions, algorithms, and processes for extracting non-obvious and useful patterns from large datasets. It is

closely related to the fields of data mining and machine learning, but broader in scope. This book offers a brief history of the field, introduces fundamental data concepts, and describes the stages in a data science project. It considers data infrastructure and the challenges posed by integrating data from multiple sources, introduces the basics of machine learning, and discusses how to link machine learning expertise with real-world problems. The book also reviews ethical and legal issues, developments in data regulation, and computational approaches to preserving privacy. Finally, it considers the future impact of data science and offers principles for success in data science projects.

An investigation into why so few African American and Latino high school students are studying computer science reveals the dynamics of inequality in American schools. The number of African Americans and Latino/as receiving undergraduate and advanced degrees in computer science is disproportionately low, according to recent surveys. And relatively few African American and Latino/a high school students receive the kind of institutional encouragement, educational opportunities, and preparation needed for them to choose computer science as a field of study and profession. In *Stuck in the Shallow End*, Jane Margolis looks at the daily experiences of students and teachers in three Los Angeles public high schools: an overcrowded urban high school, a math and science magnet school, and a well-funded school in an affluent neighborhood. She finds an insidious “virtual segregation” that maintains inequality. Two of the three schools studied offer only low-level, how-to (keyboarding, cutting and pasting) introductory computing classes. The third and wealthiest school offers advanced courses, but very few students of color enroll in them. The race gap in computer science, Margolis finds, is one example of the way students of color are denied a wide range of occupational and educational futures. Margolis traces the interplay of school structures (such factors as course offerings and student-to-counselor ratios) and belief systems—including teachers' assumptions about their students and students' assumptions about themselves. *Stuck in the Shallow End* is a story of how inequality is reproduced in America—and how students and teachers, given the necessary tools, can change the system.

This title gives students an integrated and rigorous picture of applied computer science, as it comes to play in the construction of a simple yet powerful computer system.

For the Introduction to Computer Science course *Computer Science: An Overview* uses broad coverage and clear exposition to present a complete picture of the dynamic computer science field. Accessible to students from all backgrounds, Glenn Brookshear uses a language-independent context to encourage the development of a practical, realistic understanding of the field. An overview of each of the important areas of Computer Science provides students with a general level of proficiency for future courses. Teaching and Learning Experience This program will provide a better teaching and learning experience—for you and your students. It will help: Develop a Practical, Realistic Understanding of Computer Science: A language-independent overview of each of the important areas of Computer Science prepares students for future courses. Fit your Course Preferences: Individual chapters are independent and can be covered in an order that suits your course. Reinforce Core Concepts: More than 1000 Questions and Exercises, Chapter Review Problems, and Social Issues questions give students the opportunity to apply concepts. The full text downloaded to your computer. With eBooks you can: search for key concepts, words and phrases make highlights and notes as you study share your notes with friends Print 5 pages at a time Compatible for PCs and MACs No expiry (offline access will remain whilst the Bookshelf software is installed. eBooks are downloaded to your computer and accessible either offline through the VitalSource Bookshelf (available as a free download), available online and also via the iPad/Android app. When the eBook is purchased, you will receive an email with your access code. Simply go to <http://bookshelf.vitalsource.com/> to download the FREE Bookshelf software. After installation, enter your access code for your eBook. Time limit The VitalSource products do not have an expiry date. You will continue to access your VitalSource products whilst you have your VitalSource Bookshelf installed.

Introduction to Computing is a comprehensive text designed for the CS0 (Intro to CS) course at the college level. It may also be used as a primary text for the Advanced Placement Computer Science course at the high school level.

Previously named *A Dictionary of Computing*, this bestselling dictionary has been renamed *A Dictionary of Computer Science*, and fully revised by a team of computer specialists, making it the most up-to-date and authoritative guide to computing available. Containing over 6,500 entries and with expanded coverage of multimedia, computer applications, networking, and personal computer science, it is a comprehensive reference work encompassing all aspects of the subject and is as valuable for home and office users as it is indispensable for students of computer science. Terms are defined in a jargon-free and concise manner with helpful examples where relevant. The dictionary contains approximately 150 new entries including cloud computing, cross-site scripting, iPad, semantic attack, smartphone, and virtual learning environment. Recommended web links for many entries, accessible via the *Dictionary of Computer Science* companion website, provide valuable further information and the appendices include useful resources such as generic domain names, file extensions, and the Greek alphabet. This dictionary is suitable for anyone who uses computers, and is ideal for students of computer science and the related fields of IT, maths, physics, media communications, electronic engineering, and natural sciences.

This book covers elementary discrete mathematics for computer science and engineering. It emphasizes mathematical definitions and proofs as well as applicable methods. Topics include formal logic notation, proof methods; induction, well-ordering; sets, relations; elementary graph theory; integer congruences; asymptotic notation and growth of functions; permutations and combinations, counting principles; discrete probability. Further selected topics may also be covered, such as recursive definition and structural induction; state machines and invariants; recurrences; generating functions.

Written by the members of the IFIP Working Group 2.3 (Programming Methodology) this text constitutes an exciting reference on the front-line of research activity in programming methodology. The range of subjects reflects the current interests of the members, and will offer insightful and controversial opinions on modern programming methods and practice. The material is arranged in thematic sections, each one introduced by a problem which epitomizes the spirit of that topic. The exemplary problem will encourage vigorous discussion and will form the basis for an introduction/tutorial for its section.

You know how to code..but is it enough? Do you feel left out when other programmers talk about asymptotic bounds? Have you failed a job interview because you don't know computer science? The author, a senior developer at a major software company with a PhD in computer science, takes you through what you would have learned while earning a four-year computer science degree. Volume one covers the most frequently referenced topics, including algorithms and data structures, graphs, problem-solving techniques, and complexity theory. When you finish this book, you'll have the tools you need to hold your own with people who have - or expect you to have - a computer science degree. This guide offers students an overview of computer science principles, and provides a solid foundation for those continuing their study in this dynamic and exciting discipline. New features of this edition include: a chapter on computer security providing readers with the latest information on preventing unauthorized access; types of malware and anti-virus software; protecting online information, including data collection issues with Facebook, Google, etc.; security issues with mobile and portable devices; a new section on cloud computing offering

readers an overview of the latest way in which businesses and users interact with computers and mobile devices; a rewritten section on social networks including new data on Google+ and Facebook; updates to include HTML5; revised and updated Did You Know callouts are included in the chapter margins; revisions of recommendations by the ACM dealing with computer ethic issues. --

"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.

Understanding and overcoming the gender gap in computer science education. The information technology revolution is transforming almost every aspect of society, but girls and women are largely out of the loop. Although women surf the Web in equal numbers to men and make a majority of online purchases, few are involved in the design and creation of new technology. It is mostly men whose perspectives and priorities inform the development of computing innovations and who reap the lion's share of the financial rewards. As only a small fraction of high school and college computer science students are female, the field is likely to remain a "male clubhouse," absent major changes. In *Unlocking the Clubhouse*, social scientist Jane Margolis and computer scientist and educator Allan Fisher examine the many influences contributing to the gender gap in computing. The book is based on interviews with more than 100 computer science students of both sexes from Carnegie Mellon University, a major center of computer science research, over a period of four years, as well as classroom observations and conversations with hundreds of college and high school faculty. The interviews capture the dynamic details of the female computing experience, from the family computer kept in a brother's bedroom to women's feelings of alienation in college computing classes. The authors investigate the familial, educational, and institutional origins of the computing gender gap. They also describe educational reforms that have made a dramatic difference at Carnegie Mellon—where the percentage of women entering the School of Computer Science rose from 7% in 1995 to 42% in 2000—and at high schools around the country.

This book is suitable for use in a university-level first course in computing (CS1), as well as the increasingly popular course known as CS0. It is difficult for many students to master basic concepts in computer science and programming. A large portion of the confusion can be blamed on the complexity of the tools and materials that are traditionally used to teach CS1 and CS2. This textbook was written with a single overarching goal: to present the core concepts of computer science as simply as possible without being simplistic.

Mathematical Modelling sets out the general principles of mathematical modelling as a means comprehending the world. Within the book, the problems of physics, engineering, chemistry, biology, medicine, economics, ecology, sociology, psychology, political science, etc. are all considered through this uniform lens. The author describes different classes of models, including lumped and distributed parameter systems, deterministic and stochastic models, continuous and discrete models, static and dynamical systems, and more. From a mathematical point of view, the considered models can be understood as equations and systems of equations of different nature and variational principles. In addition to this, mathematical features of mathematical models, applied control and optimization problems based on mathematical models, and identification of mathematical models are also presented. Features Each chapter includes four levels: a lecture (main chapter material), an appendix (additional information), notes (explanations, technical calculations, literature review) and tasks for independent work; this is suitable for undergraduates and graduate students and does not require the reader to take any prerequisite course, but may be useful for researchers as well. Described mathematical models are grouped both by areas of application and by the types of obtained mathematical problems, which contributes to both the breadth of coverage of the material and the depth of its understanding. Can be used as the main textbook on a mathematical modelling course, and is also recommended for special courses on mathematical models for physics, chemistry, biology, economics, etc.

Deontic logic originated from moral philosophy and the philosophy of law as an attempt to formalise normative and legal reasoning. This book draws on the experience of researchers - working in fields as diverse as Artificial Intelligence, linguistics, computer system specification and law - who have discovered the benefits of deontic logic when applied to solving computer science and AI problems. A useful logic in which to specify normative system behaviour, deontic logic has a broad spectrum of possible applications within the field: from legal expert systems to natural language processing, database integrity to electronic contracting and the specification of fault-tolerant software. This book provides a unique and timely assessment of the practical value of deontic logic for computer scientists in AI and law and, more particularly, in such areas as distributed AI and intelligent cooperative information systems.

Essential Computational Thinking: Computer Science from Scratch helps students build a theoretical and practical foundation for learning computer science. Rooted in fundamental science, this text defines elementary ideas including data and information, quantifies these ideas mathematically, and, through key concepts in physics and computation, demonstrates the relationship between computer science and the universe itself. In Part I, students explore the theoretical underpinnings of computer science in a wide-ranging manner. Readers receive a robust overview of essential computational theories and programming ideas, as well as topics that examine the mathematical and physical foundations of computer science. Part 2 presents the basics of computation and underscores programming as an invaluable tool in the discipline. Students can apply their newfound knowledge and begin writing substantial programs immediately. Finally, Part 3 explores more sophisticated computational ideas, including object-oriented programming, databases, data science, and some of the underlying principles of machine learning. *Essential Computational Thinking* is an ideal text for a firmly technical CS0 course in computer science. It is also a valuable resource for highly-motivated non-computer science majors at the undergraduate or graduate level who are interested in learning more about the discipline for either professional or personal development.

Computer Science: An Overview uses broad coverage and clear exposition to present a complete picture of the dynamic computer science field. Accessible to students from all backgrounds, Glenn Brookshear uses a language-independent context to encourage the development of a practical, realistic understanding of the field. An overview of each of the important areas of Computer Science (e.g. Networking, OS, Computer Architecture, Algorithms) provides students with a general level of proficiency for future courses. The Eleventh Edition features two new contributing authors (David Smith — Indiana University of PA; Dennis Brylow — Marquette University), new, modern examples, and updated coverage based on current technology.

Describes the LISP programming language, and covers basic procedures, data, and modularity

[Copyright: 8673d085f833b11aaa05230d44918ae4](#)