

## 15 313 Foundations Of Software Engineering

This book constitutes the proceedings of the 15th Asian Symposium on Programming Languages and Systems, APLAS 2017, held in Suzhou, China, in November 2017. The 24 papers presented in this volume were carefully reviewed and selected from 56 submissions. They were organized in topical sections named: security; heap and equivalence reasoning; concurrency and verification; domain-specific languages; semantics; and numerical reasoning. The volume also contains two invited talks in full-paper length.

Theoretical Foundations of Learning Environments provides students, faculty, and instructional designers with a clear, concise introduction to the major pedagogical and psychological theories and their implications for the design of new learning environments for schools, universities, or corporations. Leading experts describe the most important contemporary theories that form the foundation of the conception and design of student-centered learning environments and new applications of educational technologies. This book is well suited as a textbook for courses in instructional design, educational psychology, learning theory, curriculum theory and design, and related areas. The rise of constructivism and its associated theories represented a paradigm shift for educators and instructional designers to a view of learning as necessarily more social, conversational, and constructive than traditional transmissive views of learning. This bestselling book was the first to provide a manageable overview of the altered field, and the second edition has been fully updated to include expert introductions to Metacognition, Argumentation, and other key contemporary theories.

The second instance of the international summer school on Generative and Transformational Techniques in Software Engineering (GTTSE 2007) was held in Braga, Portugal, during July 2–7, 2007. This volume contains an augmented selection of the material presented at the school, including full tutorials, short tutorials, and contributions to the participants workshop. The GTTSE summer school series brings together PhD students, lecturers, technology presenters, as well as other researchers and practitioners who are interested in the generation and the transformation of programs, data, models, metamodels, documentation, and entire software systems. This concerns many areas of software engineering: software reverse and re-engineering, model-driven engineering, automated software engineering, generic language technology, to name a few. These areas differ with regard to the specific sorts of metamodels (or grammars, schemas, formats etc.) that underlie the involved artifacts, and with regard to the specific techniques that are employed for the generation and the transformation of the artifacts. The first instance of the school was held in 2005 and its proceedings appeared as volume 4143 in the LNCS series.

The two-volume set LNCS 9779 and LNCS 9780 constitutes the refereed proceedings of the 28th International Conference on Computer Aided Verification, CAV 2016, held in Toronto, ON, USA, in July 2016. The total of 46 full and 12 short papers presented in the proceedings was carefully reviewed and selected from 195 submissions. The papers were organized in topical sections named: probabilistic systems; synthesis; constraint solving; model checking; program analysis; timed and hybrid systems; verification in practice; concurrency; and automata and games.

In recent years, searching for source code on the web has become increasingly common among professional software developers and is emerging as an area of academic research. This volume surveys past research and presents the state of the art in the area of "code retrieval on the web." This work is concerned with the algorithms, systems, and tools to allow programmers to search for source code on the web and the empirical studies of these inventions and practices. It is a label that we apply to a set of related research from software engineering, information retrieval, human-computer interaction, management, as well as commercial products. The division of code retrieval on the web into snippet remixing and component reuse is driven both by empirical data, and analysis of existing search engines and tools. Contributors include leading researchers from human-computer interaction, software engineering, programming languages, and management. "Finding Source Code on the Web for Remix and Reuse" consists of five parts. Part I is titled "Programmers and Practices," and consists of a retrospective chapter and two empirical studies on how programmers search the web for source code. Part II is titled "From Data Structures to Infrastructures," and covers the creation of ground-breaking search engines for code retrieval required ingenuity in the adaptation of existing technology and in the creation of new algorithms and data structures. Part III focuses on "Reuse: Components and Projects," which are reused with minimal modification. Part IV is on "Remix: Snippets and Answers," which examines how source code from the web can also be used as solutions to problems and answers to questions. The book concludes with Part V, "Looking Ahead," that looks at future programming and the legalities of software reuse and remix and the implications of current intellectual property law on the future of software development. The story, "Richie Boss: Private Investigator Manager," was selected as the winner of a crowdfunded short story contest.

This volume contains the proceedings of ICALP 88, held at Tampere University of Technology, Finland, July 11-15, 1988. ICALP 88 is the 15th International Colloquium on Automata, Languages and Programming in a series of meetings sponsored by the European Association for Theoretical Computer Science (EATCS). It is a broadly based conference covering all aspects of theoretical computer science including topics such as computability, automata, formal languages, analysis of algorithms, computational complexity, data types and data structures, theory of data bases and knowledge bases, semantics of programming languages, program specification, transformation and verification, foundations of logic programming, theory of logical design and layout, parallel and distributed computation, theory of concurrency, symbolic and algebraic computation, term rewriting systems, cryptography, and theory of robotics.

The papers of this volume focus on the foundational aspects of computer science, the thematic origin and stronghold of LNCS, under the title "Computing and Software Science:

State of the Art and Perspectives”. They are organized in two parts: The first part, Computation and Complexity, presents a collection of expository papers on fashionable themes in algorithmics, optimization, and complexity. The second part, Methods, Languages and Tools for Future System Development, aims at sketching the methodological evolution that helps guaranteeing that future systems meet their increasingly critical requirements. Chapter 3 is available open access under a Creative Commons Attribution 4.0 International License via [link.springer.com](http://link.springer.com).

This book constitutes the proceedings of the 7th Joint International Conference on Service-Oriented Computing, ICSOC-ServiceWave 2009, held in Stockholm, Sweden, in November 2009. The 54 contributions to this volume, consisting of 37 full papers, 8 short papers and 9 demonstration papers, were carefully reviewed and selected from 228 submissions. The papers are arranged in topical sections on composition, discovery, design principles, customization and adaptation, negotiation, agreements and compliance, selection, platforms and infrastructures, security, modeling and design, validation and verification, reputation and ranking, and service management. This volume launches the new subline of Lecture Notes in Computer Science, entitled LNCS Services Science.

This book constitutes the refereed proceedings of the 12th International Conference on Concurrency Theory, CONCUR 2001, held in Aalborg, Denmark in August 2001. The 32 revised full papers presented together with six invited contributions were carefully reviewed and selected from 78 submissions. The papers are organized in topical sections on mobility, probabilistic systems, model checking, process algebra, unfoldings and prefixes, logic and compositionality, and games.

This book constitutes the refereed proceedings of the 16th International Conference on Software Reuse, ICSR 2017, held in Salvador, Brazil, in May 2017. The 8 revised full papers presented together with 3 short papers and 2 keynote presentations were carefully reviewed and selected from 34 submissions. The papers are grouped in topical sections on documentation reuse and repositories; software product lines; variability management and model variants; verification and refactoring for reuse; tools demonstrations; doctoral symposium; tutorials; and workshop.

The Internet is a remarkable catalyst for creativity, collaboration and innovation providing us with amazing possibilities that just two decades ago would have been impossible to imagine. This work includes a peer-reviewed collection of scientific papers addressing some of the challenges that shape the Internet of the future.

**CIO BEST PRACTICES Enabling Strategic Value with Information Technology SECOND EDITION** For anyone who wants to achieve better returns on their IT investments, CIO Best Practices, Second Edition presents the leadership skills and competencies required of a CIO addressing comprehensive enterprise strategic frameworks to fully leverage IT resources. Filled with real-world examples of CIO success stories, the Second Edition explores: CIO leadership responsibilities and opportunities The business impacts of both business and social networking, as well as ways the CIO can leverage the new reality of human connectivity on the Internet The increasingly inextricable relationships between customers, employees, and their use of personal information technologies Emerging cultural expectations and standards outside the workplace Current CRM best practices in terms of the relationship between customer preferences and shareholder wealth Enterprise energy utilization and sustainability practices—otherwise known as Green IT—with all the best practices collected here, in one place Best practices for one of the Internet's newest and most revolutionary technologies: cloud computing and ways it is shaping the new economics of business

**CIO Best Practices Enabling Strategic Value With Information Technology** John Wiley & Sons

This two-volume set (CCIS 1045 and CCIS 1046) constitutes the refereed proceedings of the Third International Conference on Advances in Computing and Data Sciences, ICACDS 2019, held in Ghaziabad, India, in April 2019. The 112 full papers were carefully reviewed and selected from 621 submissions. The papers are centered around topics like advanced computing, data sciences, distributed systems organizing principles, development frameworks and environments, software verification and validation, computational complexity and cryptography, machine learning theory, database theory, probabilistic representations.

This book constitutes the refereed proceedings of the 15th International Conference on Verification, Model Checking and Abstract Interpretation, VMCAI 2014, held in San Diego, CA, USA, in January 2013. The 25 revised full papers presented were carefully reviewed and selected from 64 submissions. The papers cover a wide range of topics including program verification, model checking, abstract interpretation and abstract domains, program synthesis, static analysis, type systems, deductive methods, program certification, debugging techniques, program transformation, optimization, hybrid and cyber-physical systems.

This book constitutes the refereed proceedings of the 15th International Conference on Product-Focused Software Process Improvement, PROFES 2014, held in Helsinki, Finland, in December 2014. The 18 revised full papers presented together with 14 short papers were carefully reviewed and selected from 45 initial submissions. The papers are organized in topical sections on agile development, decision-making, development practices and issues, product planning, and project management.

Smart cards or IC cards offer a huge potential for information processing purposes. The portability and processing power of IC cards allow for highly secure conditional access and reliable distributed information processing. IC cards that can perform highly sophisticated cryptographic computations are already available. Their application in the financial services and telecom industries are well known. But the potential of IC cards go well beyond that. Their applicability in mainstream Information Technology and the Networked Economy is limited mainly by our imagination; the information processing power that can be gained by using IC cards remains as yet mostly untapped and is not well understood. Here lies a vast uncovered research area which we are only beginning to assess, and which will have a great impact on the eventual success of the technology. The research challenges range from electrical engineering on the hardware side to tailor-made cryptographic applications on the software side, and their synergies. This volume comprises the proceedings of the Fourth Working Conference on Smart Card Research and Advanced Applications (CARDIS 2000), which was sponsored by the International

Federation for Information Processing (IFIP) and held at the Hewlett-Packard Labs in the United Kingdom in September 2000. CARDIS conferences are unique in that they bring together researchers who are active in all aspects of design of IC cards and related devices and environments, thus stimulating synergy between different research communities from both academia and industry. This volume presents the latest advances in smart card research and applications, and will be essential reading for smart card developers, smart card application developers, and computer science researchers involved in computer architecture, computer security, and cryptography.

Laboratory Imaging and Photography: Best Practices for Photomicrography and More is the definitive guide to the production of scientific images. Inside, the reader will find an overview of the theory and practice of laboratory photography, along with useful approaches to choosing equipment, handling samples, and working with microscopic subjects. Drawing from over 150 years of combined experience in the field, the authors outline methods of properly capturing, processing and archiving the images that are essential to scientific research. Also included are chapters on applied close-up photography, artificial light photography and the optics used in today's laboratory environment, with detailed entries on light, confocal and scanning electron microscopy. A lab manual for the digital era, this peerless reference book explains how to record visual data accurately in an industry where a photograph can serve to establish a scientific fact. Key features include: Over 200 full-color photographs and illustrations A condensed history of scientific photography Tips on using the Adobe Creative Suite for scientific applications A cheat sheet of best practices Methods used in computational photography

The LNCS Journal on Data Semantics is devoted to the presentation of notable work that, in one way or another, addresses research and development on issues related to data semantics. The scope of the journal ranges from theories supporting the formal definition of semantic content to innovative domain-specific applications of semantic knowledge. The journal addresses researchers and advanced practitioners working on the semantic web, interoperability, mobile information services, data warehousing, knowledge representation and reasoning, conceptual database modeling, ontologies, and artificial intelligence. Volume XV results from a rigorous selection among 25 full papers received in response to two calls for contributions issued in 2009 and 2010. In addition, this volume contains a special report on the Ontology Alignment Evaluation Initiative, an event that has been held once a year in the last five years and has attracted considerable attention from the ontology community. This is the last LNCS transactions volume of the Journal on Data Semantics; the next issue will appear as a regular Springer Journal, published quarterly starting from 2012.

The best way to learn software engineering is by understanding its core and peripheral areas. Foundations of Software Engineering provides in-depth coverage of the areas of software engineering that are essential for becoming proficient in the field. The book devotes a complete chapter to each of the core areas. Several peripheral areas are also explained by assigning a separate chapter to each of them. Rather than using UML or other formal notations, the content in this book is explained in easy-to-understand language. Basic programming knowledge using an object-oriented language is helpful to understand the material in this book. The knowledge gained from this book can be readily used in other relevant courses or in real-world software development environments. This textbook educates students in software engineering principles. It covers almost all facets of software engineering, including requirement engineering, system specifications, system modeling, system architecture, system implementation, and system testing. Emphasizing practical issues, such as feasibility studies, this book explains how to add and develop software requirements to evolve software systems. This book was written after receiving feedback from several professors and software engineers. What resulted is a textbook on software engineering that not only covers the theory of software engineering but also presents real-world insights to aid students in proper implementation. Students learn key concepts through carefully explained and illustrated theories, as well as concrete examples and a complete case study using Java. Source code is also available on the book's website. The examples and case studies increase in complexity as the book progresses to help students build a practical understanding of the required theories and applications.

This book offers an overview of global alternative media activity, before moving on to provide information about alternative media production and how to get involved in it.

For Computer Systems, Computer Organization and Architecture courses in CS, EE, and ECE departments. Few students studying computer science or computer engineering will ever have the opportunity to build a computer system. On the other hand, most students will be required to use and program computers on a near daily basis. Computer Systems: A Programmer's Perspective introduces the important and enduring concepts that underlie computer systems by showing how these ideas affect the correctness, performance, and utility of application programs. The text's hands-on approach (including a comprehensive set of labs) helps students understand the under-the-hood operation of a modern computer system and prepares them for future courses in systems topics such as compilers, computer architecture, operating systems, and networking.

This book constitutes the thoroughly refereed post-proceedings of the International Workshop on Graph Transformation with Industrial Relevance, AGTIVE'99, held in Kerkrade, The Netherlands, in June 1999. The 28 revised full papers presented went through an iterated process of reviewing and revision. Also included are three invited papers, 10 tool demonstrations, a summary of a panel discussion, and lists of graph transformation systems and books on graph transformations. The papers are organized in sections on modularization concepts, distributed systems modeling, software architecture: evolution and reengineering, visual graph transformation languages, visual language modeling and tool development, knowledge modeling, image recognition and constraint solving, process modeling and view integration, and visualization and animation tools.

This book constitutes the refereed proceedings of the 14th International Conference on the Quality of Information and Communications Technology, QUATIC 2021, held in Algarve, Portugal\*, in September 2021. The 30 full papers and 9 short papers were carefully reviewed and selected from 98 submissions. The papers are organized in topical sections: ICT verification and validation; software evolution; process modeling, improvement and assessment; quality aspects in quantum computing; safety, security, and privacy; quality aspects in machine learning, AI and data analytics; evidence-based software quality engineering; quality in cyber-physical systems; software quality education and training. \*The conference was held virtually due to the COVID-19 pandemic.

ETAPS2000wasthethirdinstanceoftheEuropeanJointConferencesonTheory and Practice of Software. ETAPS is an annual federated conference that was established in 1998 by combining a number of existing and new conferences. This year it comprised five conferences (FOSSACS, FASE, ESOP, CC, TACAS), five satellite workshops (CBS, CMCS, CoFI, GRATRA, INT), seven invited lectures, a panel discussion, and ten tutorials. The events that comprise ETAPS address various aspects of the system - velopmentprocess,includingspeci cation,design,implementation,analysis,and improvement. The languages, methodologies, and tools which support these - tivities are all well within its scope. Die rent blends of theory and practice are represented, with an inclination towards theory with a practical motivation on one hand and soundly-based practice on the other. Many of the issues involved in software design apply to systems in general, including hardware systems, and the emphasis on software is not intended to be exclusive. ETAPS is a loose confederation in which each event retains its own identity, with a separate program committee and independent proceedings. Its format is open-ended, allowing it to grow and evolve as time goes by. Contributed talks and system demonstrations are in synchronized parallel sessions, with invited lectures in plenary sessions. Two of the invited lectures are reserved for "u- fying" talks on topics of interest to the whole range of ETAPS attendees.

A complete introduction to building robust and reliable software Beginning Software Engineering demystifies the software engineering methodologies and techniques that professional developers use to

design and build robust, efficient, and consistently reliable software. Free of jargon and assuming no previous programming, development, or management experience, this accessible guide explains important concepts and techniques that can be applied to any programming language. Each chapter ends with exercises that let you test your understanding and help you elaborate on the chapter's main concepts. Everything you need to understand waterfall, Sashimi, agile, RAD, Scrum, Kanban, Extreme Programming, and many other development models is inside! Describes in plain English what software engineering is Explains the roles and responsibilities of team members working on a software engineering project Outlines key phases that any software engineering effort must handle to produce applications that are powerful and dependable Details the most popular software development methodologies and explains the different ways they handle critical development tasks Incorporates exercises that expand upon each chapter's main ideas Includes an extensive glossary of software engineering terms

This book and its companion volumes, LNCS volumes 9140, 9141 and 9142, constitute the proceedings of the 6th International Conference on Swarm Intelligence, ICSI 2015 held in conjunction with the Second BRICS Congress on Computational Intelligence, CCI 2015, held in Beijing, China in June 2015. The 161 revised full papers presented were carefully reviewed and selected from 294 submissions. The papers are organized in 28 cohesive sections covering all major topics of swarm intelligence and computational intelligence research and development, such as novel swarm-based optimization algorithms and applications; particle swarm optimization; ant colony optimization; artificial bee colony algorithms; evolutionary and genetic algorithms; differential evolution; brain storm optimization algorithm; biogeography based optimization; cuckoo search; hybrid methods; multi-objective optimization; multi-agent systems and swarm robotics; Neural networks and fuzzy methods; data mining approaches; information security; automation control; combinatorial optimization algorithms; scheduling and path planning; machine learning; blind sources separation; swarm interaction behavior; parameters and system optimization; neural networks; evolutionary and genetic algorithms; fuzzy systems; forecasting algorithms; classification; tracking analysis; simulation; image and texture analysis; dimension reduction; system optimization; segmentation and detection system; machine translation; virtual management and disaster analysis.

A summary of the grants made by the Charles Stewart Mott Foundation.

This text develops a comprehensive theory of programming languages based on type systems and structural operational semantics. Language concepts are precisely defined by their static and dynamic semantics, presenting the essential tools both intuitively and rigorously while relying on only elementary mathematics. These tools are used to analyze and prove properties of languages and provide the framework for combining and comparing language features. The broad range of concepts includes fundamental data types such as sums and products, polymorphic and abstract types, dynamic typing, dynamic dispatch, subtyping and refinement types, symbols and dynamic classification, parallelism and cost semantics, and concurrency and distribution. The methods are directly applicable to language implementation, to the development of logics for reasoning about programs, and to the formal verification language properties such as type safety. This thoroughly revised second edition includes exercises at the end of nearly every chapter and a new chapter on type refinements.

This is the eagerly-anticipated revision to one of the seminal books in the field of software architecture which clearly defines and explains the topic.

Putting capability management into practice requires both a solid theoretical foundation and realistic approaches. This book introduces a development methodology that integrates business and information system development and run-time adjustment based on the concept of capability by presenting the main findings of the CaaS project – the Capability-Driven Development (CDD) methodology, the architecture and components of the CDD environment, examples of real-world applications of CDD, and aspects of CDD usage for creating business value and new opportunities. Capability thinking characterizes an organizational mindset, putting capabilities at the center of the business model and information systems development. It is expected to help organizations and in particular digital enterprises to increase flexibility and agility in adapting to changes in their economic and regulatory environments. Capability management denotes the principles of how capability thinking should be implemented in an organization and the organizational means. This book is intended for anyone who wants to explore the opportunities for developing and managing context-dependent business capabilities and the supporting business services. It does not require a detailed understanding of specific development methods and tools, although some background knowledge and experience in information system development is advisable. The individual chapters have been written by leading researchers in the field of information systems development, enterprise modeling and capability management, as well as practitioners and industrial experts from these fields.

Challenges in unpredictable markets, changing customer requirements, and advancing information technologies have lead to progression towards service oriented engineering and agile and lean software development. These prevailing approaches to software systems provide solutions to challenges in demanding business environments. Agile and Lean Service-Oriented Development: Foundations, Theory and Practice explores the groundwork of service-oriented and agile and lean development and the conceptual basis and experimental evidences for the combination of the two approaches. Highlighting the best tools and guidelines for these developments in practice, this book is essential for researchers and practitioners in the software development and service computing fields.

The integration of AI with software is an essential enabler for science and the new economy, creating new markets and opportunities for a more reliable, flexible and robust society. Current software methodologies, tools and techniques often fall short of expectations, however, and much software remains insufficiently robust and reliable for a constantly changing and evolving market. This book presents 54 papers delivered at the 20th edition of the International Conference on New Trends in Intelligent Software Methodology Tools, and Techniques (SoMeT\_21), held in Cancun, Mexico, from 21–23 September 2021. The aim of the conference was to capture the essence of a new state-of-the-art in software science and its supporting technology and to identify the challenges that such a technology will need to master, and this book explores the new trends and theories illuminating the direction of development in this field as it heads towards a transformation in the role of software and science integration in tomorrow's global information society. The 54 revised papers were selected for publication by means of a rigorous review process involving 3 or 4 reviewers for each paper, followed by selection by the SoMeT\_21 international reviewing committee. The book is divided into 9 chapters, classified by paper topic and relevance to the chapter theme. Covering topics ranging from research practices, techniques and methodologies to proposing and reporting on the solutions required by global business, the book offers an opportunity for the software science community to consider where they are today and where they are headed in the future.

This book constitutes the thoroughly refereed post-workshop proceedings of the 7th International Workshop on Web Services and Formal Methods, WS-FM 2010, held in Hoboken, NJ, USA, in September 2010. The 11 revised full papers presented together with one invited paper were carefully reviewed and selected from 26 submissions. The papers feature topics such as web services; service oriented computing; cloud computing; formal methods; verification specification; testing; and business process management.

This book identifies, defines and illustrates the fundamental concepts and engineering techniques relevant to applications of software languages in software development. It presents software languages primarily from a software engineering perspective, i.e., it addresses how to parse, analyze, transform, generate, format, and otherwise process software artifacts in different software languages, as they appear in software development. To this end, it covers a wide range of software languages – most notably programming languages, domain-specific languages, modeling languages, exchange formats, and specifically also language definition languages. Further, different languages are leveraged to illustrate software language engineering concepts and techniques. The functional programming language Haskell dominates the book, while the mainstream programming languages Python and Java are additionally used for illustration. By doing this, the book collects and organizes scattered knowledge from software language engineering, focusing on application areas such as software analysis (software reverse engineering), software transformation (software re-engineering), software composition (modularity), and domain-specific languages. It is designed as a textbook for independent study as well as for bachelor's (advanced level) or master's university courses in Computer Science. An additional website provides complementary material, for example, lecture slides and videos. This book is a valuable resource for anyone wanting to understand the fundamental concepts and important engineering principles underlying software languages, allowing them to acquire much of the operational intelligence needed for dealing with software languages in software development practice. This is an important skill set for software engineers, as languages are increasingly permeating software development. Today, software engineers need to know not only how to program effectively but also how to develop proper engineering practices to make their codebase sustainable and healthy. This book emphasizes this difference between programming and software engineering. How can software engineers manage a living codebase that evolves and responds to changing requirements and demands over the length of its life? Based on their experience at Google, software engineers Titus Winters and Hyrum Wright, along with technical writer Tom Manshreck, present a candid and insightful look at how some of the world's leading practitioners construct and maintain software. This book covers Google's unique engineering culture, processes, and tools and how these aspects contribute to the effectiveness of an engineering organization. You'll explore three fundamental principles that software organizations should keep in mind when designing, architecting, writing, and maintaining code: How time affects the sustainability of software and how to make your code resilient over time How scale affects the viability of software practices within an engineering organization What trade-offs a typical engineer needs to make when evaluating design and development decisions

This book constitutes the proceedings of the 15th International Conference on Foundations of Software Science and Computational Structures, FOSSACS 2012, held as part of the joint European Conference on Theory and Practice of Software, ETAPS 2012, which took place in Tallinn, Estonia, in March/April 2012. The 29 papers presented in this book together with two invited talks in full paper length were carefully reviewed and selected from 100 full paper submissions. The papers deal with theories and methods to support analysis, synthesis, transformation and verification of programs and software systems.

[Copyright: f2e198cae9d52f0d06ce1b13a0b71876](#)