

Software Engineering Diploma Notes

The challenges in implementing intelligent and autonomous software systems remain the development of self-adapting systems, self-healing applications, corporate global creation, and collaborated robotic teams. With software agent technology widely recognized as a key approach in implementing such global infrastructure, the importance of the role of quality assurance of agent-based systems and system development is growing daily. Based on the authors' more than fifteen years of experience in software agent technology, *Quality of Agent-Based and Self-Managed Systems* presents the basic principles and structures of agent technology. It covers the main quality issues of software system development and provides examples of agent measurement and evaluation. The authors focus on software agent systems and multi-agent systems (MAS) and discuss the determination of quality properties. They also explain different techniques and approaches to evaluate the development of MAS. The final chapter summarizes quality assurance approaches for agent-based systems and discusses some open problems and future directions. Although often complex and difficult to manage, the applications for software agent systems in essential life systems increase every day. Since the quality of the agent-based self-managing systems is a central point of software risks, analyzing, evaluating, and improving the quality measurement situation will always be a concern when developing these systems. With more than sixty illustrations and twenty tables, this book builds a foundation in quality and quality control for agent-based technology.

This book constitutes the refereed proceedings of the First European Symposium on Principles of Data Mining and Knowledge Discovery, PKDD '97, held in Trondheim, Norway, in June 1997. The volume presents a total of 38 revised full papers together with abstracts of one invited talk and four tutorials. Among the topics covered are data and knowledge representation, statistical and probabilistic methods, logic-based approaches, man-machine interaction aspects, AI contributions, high performance computing support, machine learning, automated scientific discovery, quality assessment, and applications.

The book is designed to help the first year engineering students in building their concepts in the course on Programming for Problem Solving. It introduces the subject in a simple and lucid manner for a better understanding. It adopts a student friendly approach to the subject matter with many solved examples and unsolved questions, illustrations and well-structured C programs.

Empirical verification of knowledge is one of the foundations for developing any discipline. As far as software construction is concerned, the empirically verified knowledge is not only sparse but also not very widely disseminated among developers and researchers. This book aims to spread the idea of the importance of empirical knowledge in software development from a highly practical viewpoint. It has two goals: (1) Define the body of empirically validated knowledge in software development so as to advise practitioners on what methods or techniques have been empirically analysed and what the results were; (2) as empirical tests have traditionally been carried out by universities or research centres, propose techniques applicable by industry to check on the software development technologies they use. Contents:

Limitations of Empirical Testing Technique Knowledge (N Juristo et al.); Replicated Studies: Building a Body of Knowledge about Software Reading Techniques (F Shull et al.); Combining Data from Reading Experiments in Software Inspections OCo A Feasibility Study (C Wholin et al.); External Experiments OCo A Workable Paradigm for Collaboration Between Industry and Academia (F Houdek); (Quasi-)Experimental Studies in Industrial Settings (O Laitenberger & D Rombach); Experimental Validation of New Software Technology (M V Zelkowitz et al.).

Readership: Researchers, academics and professionals in software engineering."

This is the eBook of the printed book and may not include any media, website access codes, or print supplements that may come packaged with the bound book. Intended for introductory and advanced courses in software engineering. The ninth edition of *Software Engineering* presents a broad perspective of software engineering, focusing on the processes and techniques fundamental to the creation of reliable, software systems. Increased coverage of agile methods and software reuse, along with coverage of 'traditional' plan-driven software engineering, gives readers the most up-to-date view of the field currently available. Practical case studies, a full set of easy-to-access supplements, and extensive web resources make teaching the course easier than ever. The book is now structured into four parts: 1: Introduction to Software Engineering 2: Dependability and Security 3: Advanced Software Engineering 4: Software Engineering Management Statistics and Probability for Engineering Applications provides a complete discussion of all the major topics typically covered in a college engineering statistics course. This textbook minimizes the derivations and mathematical theory, focusing instead on the information and techniques most needed and used in engineering applications. It is filled with practical techniques directly applicable on the job. Written by an experienced industry engineer and statistics professor, this book makes learning statistical methods easier for today's student. This book can be read sequentially like a normal textbook, but it is designed to be used as a handbook, pointing the reader to the topics and sections pertinent to a particular type of statistical problem. Each new concept is clearly and briefly described, whenever possible by relating it to previous topics. Then the student is given carefully chosen examples to deepen understanding of the basic ideas and how they are applied in engineering. The examples and case studies are taken from real-world engineering problems and use real data. A number of practice problems are provided for each section, with answers in the back for selected problems. This book will appeal to engineers in the entire engineering spectrum (electronics/electrical, mechanical, chemical, and civil engineering); engineering students and students taking computer science/computer engineering graduate courses; scientists needing to use applied statistical methods; and engineering technicians and technologists. * Filled with practical techniques directly applicable on the job * Contains hundreds of solved problems and case studies, using real data sets * Avoids unnecessary theory

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.

"All aspects pertaining to algorithm design and algorithm analysis have been discussed over the chapters in this book-- Design and Analysis of Algorithms"--Resource description page.

The goal of this book is to introduce to the students a limited number of concepts and practices which will achieve the following two objectives: Teach the student the skills needed to execute a smallish commercial project. Provide the students necessary conceptual background for undertaking advanced studies in software engineering, through organized courses or on their own. This book focuses on key tasks in two dimensions - engineering and project management - and discusses concepts and techniques that can be applied to effectively execute these tasks. The book is organized in a simple manner, with one chapter for each of the key tasks in a project. For engineering, these tasks are requirements analysis and specification, architecture design, module level design, coding and unit testing, and testing. For project management, the key tasks are project planning and project monitoring and control, but both are discussed together in one chapter on project planning as even monitoring has to be planned. In addition, one chapter clearly defines the problem domain of Software Engineering, and another Chapter discusses the central concept of software process which integrates the different tasks executed in a project. Each chapter opens with some introduction and clearly lists the chapter goals, or what the reader can expect to learn from the chapter. For the task covered in the chapter, the important concepts are first discussed, followed by a discussion of the output of the task, the desired quality properties of the output, and

some practical methods and notations for performing the task. The explanations are supported by examples, and the key learnings are summarized in the end for the reader. The chapter ends with some self-assessment exercises. Finally, the book contains a question bank at the end which lists out questions with answers from major universities.

This Three-Volume-Set constitutes the refereed proceedings of the Second International Conference on Software Engineering and Computer Systems, ICSECS 2011, held in Kuantan, Malaysia, in June 2011. The 190 revised full papers presented together with invited papers in the three volumes were carefully reviewed and selected from numerous submissions. The papers are organized in topical sections on software engineering; network; bioinformatics and e-health; biometrics technologies; Web engineering; neural network; parallel and distributed; e-learning; ontology; image processing; information and data management; engineering; software security; graphics and multimedia; databases; algorithms; signal processing; software design/testing; e- technology; ad hoc networks; social networks; software process modeling; miscellaneous topics in software engineering and computer systems. 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.

Topological UML Modeling: An Improved Approach for Domain Modeling and Software Development presents a specification for Topological UML® that combines the formalism of the Topological Functioning Model (TFM) mathematical topology with a specified software analysis and design method. The analysis of problem domain and design of desired solutions within software development processes has a major impact on the achieved result – developed software. While there are many tools and different techniques to create detailed specifications of the solution, the proper analysis of problem domain functioning is ignored or covered insufficiently. The design of object-oriented software has been led for many years by the Unified Modeling Language (UML®), an approved industry standard modeling notation for visualizing, specifying, constructing, and documenting the artifacts of a software-intensive system, and this comprehensive book shines new light on the many advances in the field. Presents an approach to formally define, analyze, and verify functionality of existing processes and desired processes to track incomplete or incorrect functional requirements Describes the path from functional and nonfunctional requirements specification to software design with step-by-step creation and transformation of diagrams and models with very early capturing of security requirements for software systems. Defines all modeling constructs as extensions to UML®, thus creating a new UML® profile which can be implemented in existing UML® modeling tools and toolsets

Fundamentals of Software Engineering PHI Learning Pvt. Ltd. British Vocational Qualifications Kogan Page Publishers

For almost three decades, Roger Pressman's Software Engineering: A Practitioner's Approach has been the world's leading textbook in software engineering. The new eighth edition represents a major restructuring and update of previous editions, solidifying the book's position as the most comprehensive guide to this important subject. The eighth edition of Software Engineering: A Practitioner's Approach has been designed to consolidate and restructure the content introduced over the past two editions of the book. The chapter structure will return to a more linear presentation of software engineering topics with a direct emphasis on the major activities that are part of a generic software process. Content will focus on widely used software engineering methods and will de-emphasize or completely eliminate discussion of secondary methods, tools and techniques. The intent is to provide a more targeted, prescriptive, and focused approach, while attempting to maintain SEPA's reputation as a comprehensive guide to software engineering. The 39 chapters of the eighth edition are organized into five parts - Process, Modeling, Quality Management, Managing Software Projects, and Advanced Topics. The book has been revised and restructured to improve pedagogical flow and emphasize new and important software engineering processes and practices.

Mobile agents are software nomads that act as your personal representative, working autonomously through networks. They are able to visit network nodes directly using available computing power and are not limited by platform. This emerging field is now poised to become a cornerstone for new Web-based ubiquitous computing environments. Mobile Agents provides a practical introduction to mobile agent technology and surveys the state of the art in mobile agent research. Students and researchers can use the book as an introduction to the concepts and possibilities of this field and as an overview of ongoing research. Developers can use it to identify the capabilities of the technology to decide if mobile agents are the right solution for them. Practitioners can also gain hands-on experience in programming mobile agents through exploration of the source code for a complete mobile agent environment available through the companion website. *Summarizes the state of the art in mobile agent research *Identifies the benefits and limitations of current mobile agent technology to help developers understand the possibilities of this new field *Extensive mobile agents web portal (www.mobile-agents.org) with the Java source code for a complete industrial-quality environment for mobile agents, with significant parts of the system open source

The increasing requirement for Junior Engineers/Technicians in PSUs has created a large job opportunities for the diploma holders all over India. Every PSU conducts its own qualifying exam based on the vacancies available for various positions such as Junior Engineer and Technician. This series has been thoroughly updated to equip the diploma engineers appearing for the exams of BHEL, BEL, GAIL, IOCL, HPCL, ONGC, DMRC, DRDO, Railway, Staff Selection Commission and other diploma engineering competitive examinations. It aids in fast revision through key notes such as terms, definitions and formulae. The series also provides conceptual clarity to ease in attempting questions. A vast collection of questions has been categorized under two levels? questions for practice and previous years? questions of various PSU examinations to give you a feel of the actual exam. Features ? Theory and key concepts in a systematical manner ? Ample number of MCQs for practice in each chapter ? Previous years? questions to familiarize you with the pattern and level of the examination

Growing demands for the quality, safety, and security of software can only be satisfied by the rigorous application of formal methods during software design. This book methodically investigates the potential of first-order logic automated theorem provers for applications in software engineering. Illustrated by complete case studies on protocol verification, verification of security protocols, and logic-based software reuse, this book provides techniques for assessing the

prover's capabilities and for selecting and developing an appropriate interface architecture.

This custom edition is published for the University of Southern Queensland.

This book constitutes the thoroughly refereed post-proceedings of the Second International Workshop on Rapid Integration of Software Engineering Techniques, RISE 2005. The book presents 19 revised full papers together with the abstract of a keynote paper. Among the topics addressed are modelling safety case evolution, practical approaches in model mapping, context-aware service composition, techniques for representing product line core assets for automation, formal development of reactive fault-tolerant systems, and more.

Over the last decade as the importance of vocational qualifications has been firmly established, the system has become increasingly complex and hard to grasp. Now in its sixth edition, this popular and accessible reference book provides up-to-date information on over 3500 vocational qualifications in the UK. Divided into five parts, the first clarifies the role of the accrediting and major awarding bodies and explains the main types of vocational qualifications available. A directory then lists over 3500 vocational qualifications, classified by professional and career area, giving details of type of qualification, title, level, awarding body and, where possible, the course code and content. The third section comprises a glossary of acronyms used, together with a comprehensive list of awarding bodies, industry lead bodies, professional institutes and associations, with their contact details. Section four is a directory of colleges offering vocational qualifications in the UK, arranged alphabetically by area. Finally, section five is an index of all qualifications, listed alphabetically by title.

It is, indeed, widely acceptable today that nowhere is it more important to focus on the improvement of software quality than in the case of systems with requirements in the areas of safety and reliability - especially for distributed, real-time and embedded systems. Thus, much research work is under progress in these fields, since software process improvement impinges directly on achieved levels of quality, and many application experiments aim to show quantitative results demonstrating the efficacy of particular approaches. Requirements for safety and reliability - like other so-called non-functional requirements for computer-based systems - are often stated in imprecise and ambiguous terms, or not at all. Specifications focus on functional and technical aspects, with issues like safety covered only implicitly, or not addressed directly because they are felt to be obvious; unfortunately what is obvious to an end user or system user is progressively less so to others, to the extent that a software developer may not even be aware that safety is an issue. Therefore, there is a growing evidence for encouraging greater understanding of safety and reliability requirements issues, right across the spectrum from end user to software developer; not just in traditional safety-critical areas (e.g. nuclear, aerospace) but also acknowledging the need for such things as heart pacemakers and other medical and robotic systems to be highly dependable.

This book aims at providing the necessary knowledge in understanding the concepts of software testing and software quality assurance so that you can take any internationally recognized software testing / quality assurance certification examination and come out with flying colors. Also, equipped with this knowledge, you can do a great job as a testing and quality assurance professional in your career and contribute in developing reliable software for different applications, which in turn improves the quality of life of everyone on this earth. · Introduction· Software Development Life Cycle and Quality Assurance· Fundamentals of Testing· Testing Levels and Types· Static Testing Techniques· Dynamic Testing and Test Case Design Techniques· Managing the Testing Process· Software Testing Tools· Code of Ethics for Software Professionals

This book discusses a comprehensive spectrum of software engineering techniques and shows how they can be applied in practical software projects. This edition features updated chapters on critical systems, project management and software requirements.

Praise for the first edition: "This excellent text will be useful to every system engineer (SE) regardless of the domain. It covers ALL relevant SE material and does so in a very clear, methodical fashion. The breadth and depth of the author's presentation of SE principles and practices is outstanding." –Philip Allen This textbook presents a comprehensive, step-by-step guide to System Engineering analysis, design, and development via an integrated set of concepts, principles, practices, and methodologies. The methods presented in this text apply to any type of human system -- small, medium, and large organizational systems and system development projects delivering engineered systems or services across multiple business sectors such as medical, transportation, financial, educational, governmental, aerospace and defense, utilities, political, and charity, among others. Provides a common focal point for "bridging the gap" between and unifying System Users, System Acquirers, multi-discipline System Engineering, and Project, Functional, and Executive Management education, knowledge, and decision-making for developing systems, products, or services Each chapter provides definitions of key terms, guiding principles, examples, author's notes, real-world examples, and exercises, which highlight and reinforce key SE&D concepts and practices Addresses concepts employed in Model-Based Systems Engineering (MBSE), Model-Driven Design (MDD), Unified Modeling Language (UML) / Systems Modeling Language (SysML), and Agile/Spiral/V-Model Development such as user needs, stories, and use cases analysis; specification development; system architecture development; User-Centric System Design (UCSD); interface definition & control; system integration & test; and Verification & Validation (V&V) Highlights/introduces a new 21st Century Systems Engineering & Development (SE&D) paradigm that is easy to understand and implement. Provides practices that are critical staging points for technical decision making such as Technical Strategy Development; Life Cycle requirements; Phases, Modes, & States; SE Process; Requirements Derivation; System Architecture Development, User-Centric System Design (UCSD); Engineering Standards, Coordinate Systems, and Conventions; et al. Thoroughly illustrated, with end-of-chapter exercises and numerous case studies and examples, Systems Engineering Analysis, Design, and Development, Second Edition is a primary textbook for multi-discipline, engineering, system analysis, and project management undergraduate/graduate level students and a valuable reference for professionals.

This volume contains selected papers from the 4th International Workshop on Graph Grammars and Their Application to Computer Science. The topics range from foundations through algorithmic and implementational aspects to various issues that arise in application areas.

The explosive growth of application areas such as electronic commerce, enterprise resource planning and mobile computing has profoundly and irreversibly changed our views on software systems. Nowadays, software is to be based on open architectures that continuously change and evolve to accommodate new components and meet new requirements. Software must also operate on different platforms, without recompilation, and with minimal assumptions about its operating environment and its users. Furthermore, software must be robust and ? autonomous, capable of serving a naive user with a minimum of overhead and interference. Agent concepts hold great promise for responding to the new realities of software systems. They offer higher-level abstractions and mechanisms which address issues such as knowledge representation and reasoning, communication, coordination, cooperation among heterogeneous and autonomous parties, p-

ception, commitments, goals, beliefs, and intentions, all of which need conceptual modelling. On the one hand, the concrete implementation of these concepts can lead to advanced functionalities, e.g., in inference-based query answering, transaction control, adaptive workflows, brokering and integration of disparate information sources, and automated communication processes. On the other hand, their rich representational capabilities allow more faithful and flexible treatments of complex organizational processes, leading to more effective requirements analysis and architectural/detailed design.

"This book offers information on the latest advancements and research for Enterprise Interoperability knowledge as well as core concepts, theories, and future directions"--

Introduces the features of the C programming language, discusses data types, variables, operators, control flow, functions, pointers, arrays, and structures, and looks at the UNIX system interface

For courses in Software Engineering, Software Development, or Object-Oriented Design and Analysis at the Junior/Senior or Graduate level. This text can also be utilized in short technical courses or in short, intensive management courses. Shows students how to use both the principles of software engineering and the practices of various object-oriented tools, processes, and products. Using a step-by-step case study to illustrate the concepts and topics in each chapter, Bruegge and Dutoit emphasize learning object-oriented software engineer through practical experience: students can apply the techniques learned in class by implementing a real-world software project. The third edition addresses new trends, in particular agile project management (Chapter 14 Project Management) and agile methodologies (Chapter 16 Methodologies).

This book constitutes the joint refereed proceedings of the 17th International Workshop on Computer Science Logic, CSL 2003, held as the 12th Annual Conference of the EACSL and of the 8th Kurt Gödel Colloquium, KGC 2003 in Vienna, Austria, in August 2003. The 30 revised full papers presented together with abstracts of 9 invited presentations were carefully reviewed and selected from a total of 112 submissions. All current aspects of computer science logic are addressed ranging from mathematical logic and logical foundations to the application of logics in various computing aspects.

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.

This volume is based on the research papers presented in the 4th Computer Science On-line Conference. The volume Software Engineering in Intelligent Systems presents new approaches and methods to real-world problems, and in particular, exploratory research that describes novel approaches in the field of Software Engineering. Particular emphasis is laid on modern trends in selected fields of interest. New algorithms or methods in a variety of fields are also presented. The Computer Science On-line Conference (CSOC 2015) is intended to provide an international forum for discussions on the latest high-quality research results in all areas related to Computer Science. The addressed topics are the theoretical aspects and applications of Computer Science, Artificial Intelligences, Cybernetics, Automation Control Theory and Software Engineering.

In November 1989 we organised a workshop on software re-use, inviting members of the leading research teams across Europe. In retrospect, we realise that we missed a few research teams out, but nevertheless we did have a very fruitful workshop. This book is the outcome of that meeting. Prior to the workshop, teams submitted short position papers, and at the workshop made very short presentations of these. Most of the time was spent in four parallel sessions, and the reports of these sessions are given in Chapter 2. After the workshop we invited the attendees to revise and resubmit their papers in the light of the workshop, and it is these updated papers that appear in Chapter 4 onwards. The papers are in alphabetical order of first author. To complete this text we have added an introduction to software re-use as a first chapter-this was prepared by Liesbeth Dusink. We have added a comprehensive bibliography as Chapter 3, merging the bibliographies accumulated at Delft and at Brunei. To be able to organise the workshop we were sponsored by SERC, the Software Engineering Research Centre in Utrecht, Netherlands. November 1990
Liesbeth Dusink Pat Hall Contents Ust of Contributors xi

The free book "Fundamentals of Computer Programming with C#" is a comprehensive computer programming tutorial that teaches programming, logical thinking, data structures and algorithms, problem solving and high quality code with lots of examples in C#. It starts with the first steps in programming and software development like variables, data types, conditional statements, loops and arrays and continues with other basic topics like methods, numeral systems, strings and string processing, exceptions, classes and objects. After the basics this fundamental programming book enters into more advanced programming topics like recursion, data structures (lists, trees, hash-tables and graphs), high-quality code, unit testing and refactoring, object-oriented principles (inheritance, abstraction, encapsulation and polymorphism) and their implementation the C# language. It also covers fundamental topics that each good developer should know like algorithm design, complexity of algorithms and problem solving. The book uses C# language and Visual Studio to illustrate the programming concepts and explains some C# / .NET specific technologies like lambda expressions, extension methods and LINQ. The book is written by a team of developers lead by Svetlin Nakov who has 20+ years practical software development experience. It teaches the major programming concepts and way of thinking needed to become a good software engineer and the C# language in the meantime. It is a great start for anyone who wants to become a skillful software engineer. The books does not teach technologies like databases, mobile and web development, but shows the true way to master the basics of programming regardless of the languages, technologies and tools. It is good for beginners and intermediate developers who want to put a solid base for a successful career in the software engineering industry. The book is accompanied by free video lessons, presentation slides and mind maps, as well as hundreds of exercises and live examples. Download the free C# programming book, videos, presentations and other resources from <http://introprogramming.info>. Title: Fundamentals of Computer Programming with C# (The Bulgarian C# Programming Book) ISBN: 9789544007737 ISBN-13: 978-954-400-773-7 (9789544007737) ISBN-10: 954-400-773-3 (9544007733) Author: Svetlin Nakov & Co. Pages: 1132 Language: English Published: Sofia, 2013 Publisher: Faber Publishing, Bulgaria Web site: <http://www.introprogramming.info> License: CC-Attribution-Share-Alike Tags: free, programming, book, computer programming, programming fundamentals, ebook, book programming, C#, CSharp, C# book, tutorial, C# tutorial; programming concepts, programming fundamentals, compiler, Visual Studio, .NET, .NET Framework, data types, variables, expressions, statements, console, conditional statements, control-flow logic, loops, arrays, numeral systems, methods, strings, text processing, StringBuilder, exceptions, exception handling, stack trace, streams, files, text files, linear data structures, list, linked list, stack, queue, tree, balanced tree, graph, depth-first search,

DFS, breadth-first search, BFS, dictionaries, hash tables, associative arrays, sets, algorithms, sorting algorithm, searching algorithms, recursion, combinatorial algorithms, algorithm complexity, OOP, object-oriented programming, classes, objects, constructors, fields, properties, static members, abstraction, interfaces, encapsulation, inheritance, virtual methods, polymorphism, cohesion, coupling, enumerations, generics, namespaces, UML, design patterns, extension methods, anonymous types, lambda expressions, LINQ, code quality, high-quality code, high-quality classes, high-quality methods, code formatting, self-documenting code, code refactoring, problem solving, problem solving methodology, 9789544007737, 9544007733

This proceedings book includes a selection of refereed papers presented at the International Conference on Modern Mechanics and Applications (ICOMMA) 2020, which took place in Ho Chi Minh City, Vietnam, on December 2–4, 2020. The contributions highlight recent trends and applications in modern mechanics. Subjects covered include biological systems; damage, fracture, and failure; flow problems; multiscale multi-physics problems; composites and hybrid structures; optimization and inverse problems; lightweight structures; mechatronics; dynamics; numerical methods and intelligent computing; additive manufacturing; natural hazards modeling. The book is intended for academics, including graduate students and experienced researchers interested in recent trends in modern mechanics and application.

[Copyright: bccd2c82e39da1498ed3eabdc5fdb816](#)