
Object Oriented Software Engineering

Getting the books Object Oriented Software Engineering now is not type of inspiring means. You could not single-handedly going subsequent to ebook accretion or library or borrowing from your links to right of entry them. This is an enormously simple means to specifically acquire guide by on-line. This online message Object Oriented Software Engineering can be one of the options to accompany you in the same way as having supplementary time.

It will not waste your time. recognize me, the e-book will enormously appearance you supplementary thing to read. Just invest little time to way in this on-line broadcast Object Oriented Software Engineering as with ease as review them wherever you are now.



Object-oriented Software Engineering with Eiffel Addison-Wesley Professional
This comprehensive and well-written book presents the fundamentals of object-oriented software engineering and discusses the recent technological developments in the field. It focuses on object-oriented software

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

first book to provide hands-on process and practice in software engineering essentials for the beginner. The book presents steps through the software development life cycle and two running case studies that develop as the steps are presented. Running parallel to the process presentation and case studies, the book supports a semester-long software development project. This book focuses on object-oriented software development, and supports the conceptualization, analysis, design and implementation of an object-oriented project. It is mostly language-independent, with necessary code examples in Java. A subset of UML is used, with the notation explained as needed to support the readers' work. Two running case studies a video game and a library check out system show the development of a software project. Both have sample deliverables and thus provide the reader with examples of the type of work readers are to create. This book is appropriate for readers looking to gain experience in project analysis, design implementation, and testing.

Emerging Research and Opportunities
Prentice Hall

In today's modernized environment, a growing number of software companies are changing their traditional engineering approaches in response to the rapid development of computing technologies. As these businesses

adopt modern software engineering practices, they face various challenges including the integration of current methodologies and contemporary design models and the refactoring of existing systems using advanced approaches. Applications and Approaches to Object-Oriented Software Design: Emerging Research and Opportunities is a pivotal reference source that provides vital research on the development of modern software practices that impact maintenance, design, and developer productivity. While highlighting topics such as augmented reality, distributed computing, and big data processing, this publication explores the current infrastructure of software systems as well as future advancements. This book is ideally designed for software engineers, IT specialists, data scientists, business professionals, developers, researchers, students, and academicians seeking current research on contemporary software engineering methods.

A Use Case Driven Approach

Pearson Higher Ed

Integrating case studies to show the object oriented approach to software engineering, Object-Oriented and Classical Software Engineering, 7/e presents an excellent introduction to

software engineering fundamentals, covering both traditional and object-oriented techniques. The coverage of both Agile processes and Open Source Software has been considerably expanded. In addition, the Osbert Oglesby running case study has been replaced with a new case study on the Martha Stockton Greengage Foundation. The new study highlights even more aspects of the Unified Process. The book's unique organization remains in place, with Part I covering underlying software engineering theory, and Part II presenting the more practical life cycle. Complementing this well-balanced approach is the straightforward, student-friendly writing style, through which difficult concepts are presented in a clear, understandable manner. The new seventh edition provides an extensive updating of this classic software engineering text!

Object-Oriented Software Engineering Using UML, Patterns, and Java: Pearson New International Edition Addison-Wesley

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).

Object-Oriented Software: Design and Maintenance
Prentice Hall

This book covers the essential knowledge and skills needed by a student who is specializing in software engineering. Readers will learn principles of object orientation, software development, software modeling, software design, requirements analysis, and testing. The use of the Unified Modelling Language to develop software is taught in depth.

Many concepts are illustrated using complete examples, with code written in Java.

Object-Oriented Software Engineering with UML McGraw-Hill Science, Engineering & Mathematics

Aiming to provide a comprehensive introduction to object-orientation, this book places an emphasis on analysis and design and presents a coherent methodology. It includes a chapter on software engineering and uses a running example to illustrate the concepts of object-orientation.

Classical and Object-oriented Software Engineering with UML and Java Addison-Wesley

Object-Oriented Software Engineering: An Agile Unified Methodology by David Kung presents a step-by-step methodology that integrates modeling and design, UML, patterns, test-driven development, quality assurance, configuration management, and agile principles throughout the life cycle. The overall approach is casual and easy to follow, with many practical examples that show the theory at work. The author uses his experiences as well as real-world stories to help the reader understand software design principles, patterns, and other software

engineering concepts. The book also provides stimulating exercises that go far beyond the type of question that can be answered by simply copying portions of the text.

Object-Oriented Software Engineering: An Agile Unified Methodology Prentice Hall

This book covers the essential knowledge and skills needed by a student who is specializing in software engineering. Readers will learn principles of object orientation, software development, software modeling, software design, requirements analysis, and testing. The use of the Unified Modelling Language to develop software is taught in depth. Many concepts are illustrated using complete examples, with code written in Java.

Using UML, Patterns, and Java Tata McGraw-Hill Education

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. Object-Oriented Software Engineering Using UML, Patterns, and Java, 3e, shows readers 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:

readers 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).

Essays on Object-oriented Software Engineering World Scientific Publishing Company

The object-oriented paradigm supplements traditional software engineering by providing solutions to common problems such as modularity and reusability. Objects can be written for a specific purpose acting as an encapsulated black-box API that can work with other components by forming a complex system. This book provides a comprehensive overview of the many facets of the object-oriented paradigm and how it applies to software engineering. Starting with an in-depth look at objects, the book naturally progresses through the software engineering life cycle and shows how object-oriented concepts enhance each step.

Furthermore, it is designed as a roadmap with each chapter, preparing the reader with the skills necessary to advance the project. This book should be used by anyone interested in

learning about object-oriented software engineering, including students and seasoned developers. Without overwhelming the reader, this book hopes to provide enough information for the reader to understand the concepts and apply them in their everyday work. After learning about the fundamentals of the object-oriented paradigm and the software engineering life cycle, the reader is introduced to more advanced topics such as web engineering, cloud computing, agile development, and big data. In recent years, these fields have been rapidly growing as many are beginning to realize the benefits of developing on a highly scalable, automated deployment system. Combined with the speed and effectiveness of agile development, legacy systems are beginning to make the transition to a more adaptive environment. Core Features: 1. Provides a thorough exploration of the object-oriented paradigm. 2. Provides a detailed look at each step of the software engineering life cycle. 3. Provides supporting examples and documents. 4. Provides a detailed look at emerging technology and standards in object-oriented software engineering.

Object-oriented Software Engineering McGraw-Hill

Education

Object-Oriented Software Engineering is written for both the traditional one-semester and the newer two-semester software engineering curriculum. Part I covers the underlying software engineering theory, while Part II presents the more practical life cycle, workflow by workflow. The text is intended for the substantial object-oriented segment of the software engineering market. It focuses exclusively on object-oriented approaches to the development of large software systems that are the most widely used. Text includes 2 running case studies, expanded coverage of agile processes and open-source development. A Hands-On Approach McGraw-Hill Book Company Limited

Classical and Object-Oriented Software Engineering is designed for an introductory software engineering course. This book provides an excellent introduction to software engineering fundamentals, covering both traditional and object-oriented techniques. Schach's unique organization and style makes it excellent for use in a classroom setting. It presents the underlying software engineering theory in Part I and follows it up with the more practical life-cycle material in Part II. Many software engineering books are more like reference books, which do not provide the appropriate fundamentals before inundating students with implementation details. In this edition, more practical material has been added to help students understand how to use what

they are learning. This has been done through the use of "How To" boxes and greater implementation detail in the case study. Additionally, the new edition contains the references to the most current literature and includes an overview of extreme programming. The website in this edition will be more extensive. It will include Solutions, PowerPoints that incorporate lecture notes, newly developed self-quizz questions, and source code for the term project and case study.

Object-oriented Software Engineering McGraw-Hill Higher Education

This book describes how object-oriented language and object-oriented ideas can be employed throughout the software project. It describes the software engineering process from requirements analysis up to acceptance testing and contains such topics as unit testing, and system design. The book uses the C++ programming language and is intended for both the undergraduate student and the industrial developer. Material on the relationship between object-oriented techniques and prototyping is also included.

Object-oriented Software Engineering Cambridge University Press

Software -- Software Engineering.

Managing Object-oriented Software Engineering

Object-oriented Software Engineering A Use Case Driven Approach

An indispensable resource for anyone working with Eiffel, this up-to-date guide provides full coverage of the most recent version of the language, focusing on Eiffel's practical use in the development of large, mission-critical software systems. In addition to a comprehensive description of Eiffel's syntax and semantics, you will find in-depth information on style guides, analysis and design, design patterns, and validation and testing. Descriptions and comparisons of available compilers and libraries will help you decide which Eiffel tools best fit your development needs. The book even includes an Eiffel resource guide. The book's most notable feature is its three large-scale case studies that demonstrate Eiffel in action, illustrating implementation techniques and showcasing Eiffel's power and effectiveness in three different realms: the MIS world, the embedded systems/telecommunications world, and the numeric world. By reading this book, you will not only obtain a knowledge of the mechanics of Eiffel programming, but you will also come away with an understanding of Eiffel's role in the field of object-oriented technology and a sense of the language's strong potential in large software development.

0201633817B04062001

Object-oriented Software Construction Pearson College Division

The authors analyze how the structure of a package determines its developmental complexity according to such measures as bug

search times and documentation information content. The work presents arguments for why these issues impact solution cost and time more than does scalable performance. The final chapter explores the question of scalable execution and shows how scalable design relates to scalable execution. The book's focus is on program organization, which has received considerable attention in the broader software engineering community, where graphical description standards for modeling software structure and behavior have been developed by computer scientists. These discussions might be enriched by engineers who write scientific codes. This book aims to bring such scientific programmers into discussion with computer scientists. The authors do so by introducing object-oriented software design patterns in the context of scientific simulation.

A Hands-on Approach IGI Global

The object-oriented methodology OOram is new and different from all others on the market, and has been in use and development in Norway for over 12 years. This book is the authoritative account of the OOram methodology for software analysis, design, development, maintenance, and reuse.

Classical and Object-oriented Software Engineering McGraw-Hill Companies
Examines object-oriented methods, practices, terminology, and concepts

Software Engineering with Ada McGraw-Hill
Science, Engineering & Mathematics
Software -- Software Engineering.