# Object Oriented Software Engineering A Use Case Driven Approach A Use Case Approach Acm Press

Understanding Object-oriented Software EngineeringObject-oriented Software EngineeringClassical and Object-oriented Software EngineeringReliable Object-Oriented SoftwareExam Prep for: Object-Oriented Software Engineering Using Object-oriented Software Engineering with C++Object-oriented Software Development Using JavaObject-oriented Software EngineeringObject Oriented Software EngineeringObject Oriented Software EngineeringObject-oriented Software ConstructionObject-Oriented Software Engineering Using UML, Patterns, and Java: Pearson New International EditionObject-Oriented Software EngineeringClassical and Object-Oriented Software Engineering with UML and Java + Code WarriorObject-oriented Software EngineeringEssays on Object-oriented Software EngineeringObject-oriented Software EngineeringPrinciples of Object-oriented Software DevelopmentObject-oriented Construction HandbookObject-Oriented Software Engineering: An Agile Unified MethodologyObject-Oriented Software Engineering: Practical Software Development using UML and JavaGrowing Object-Oriented Software, Guided by TestsOBJECT-ORIENTED SOFTWARE ENGINEERINGObject-oriented Software ConstructionObject-oriented Software EngineeringAdvances in Object-oriented Software EngineeringClassical and Object-oriented Software Engineering with UML and C++Object-oriented Software Engineering with EiffelObject-Oriented SoftwareSoftware Engineering (Sie) 7EObject-Oriented Software EngineeringObject-oriented Software and EngineeringObject-oriented Software EngineeringA First Course in Object-oriented Software EngineeringProject-based Software EngineeringObject-Oriented Software Engineering: An Agile Unified MethodologyDesign Patterns for Object-oriented Software DevelopmentObject-Oriented Software Engineering: Practical Software Development Using Uml And JavaObject Oriented Software EngineeringDesigning Object-oriented Software

## Understanding Object-oriented Software Engineering

This text provides an introduction to the process of software engineering. The revision concentrates on updating the book to reflect the most current trends and innovations in the field. The Universal Modeling Language (UML) has become an industry standard and now permeates this new edition. In this text, it is used for object-oriented analysis and design as well as when diagrams depict objects and their interrelationships. Design patterns, frameworks and software architecture have also become a popular topic in the field of software engineering and are part of a new chapter on reuse, portability, and inoperability. The inoperabilty material includes sections on such hot topics as OLE, COM, and CORBA. Some material from the 3rd edition has been reorganized into a new chapter on planning and estimating, including feature points and COCOMO II. While the text has been updated, the traditonal features which have defined the previous three editions of Schach's book have been retained. These include a balanced coverage of the object-oriented model along with the classical model (as reflected in the title) and an emphasis on metrics. The special considerations of object-oriented life-cycle models, object-oriented analysis, and object-oriented design are also retained in this edition.

## Object-oriented Software Engineering

Addresses critical software engineering issues, showing how an object - oriented approach can provide much improved solutions over other methods. Designed as a technology tool.

## Classical and Object-oriented Software Engineering

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

## Reliable Object-Oriented Software

## Exam Prep for: Object-Oriented Software Engineering Using

This volume aims to study how practicing software developers, in industrial as well as academic environments, can use object technology to improve the quality of the software they produce. It includes topics on concurrency and Internet programming.

## Object-oriented Software Engineering with C++

This is a textbook for a course in object-oriented software engineering at advanced undergraduate and graduate levels, as well as for software engineers. It contains more than 120 exercises of diverse complexity.The book discusses fundamental concepts and terminology on object-oriented software development, assuming little background on software engineering, and emphasizes design and maintenance rather than programming.It also presents up-to-date and easily understood methodologies and puts forward a software life cycle model which explicitly encourages reusability during software development and maintenance.

## Object-oriented Software Development Using Java

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.

## Object-oriented Software Engineering

## Object Oriented Software Engineering

Jia (software engineering, DePaul University) helps readers develop skills in designing software, and especially in writing object- oriented programs using Java. The text provides broad coverage of object-oriented technology, including object-oriented modeling using the Unified Modeling Language (UML), object-oriented design using design patterns, and object-oriented programming using Java. This second edition offers expanded coverage of design patterns, enhanced material on UML, and a new introduction to the iterative software development process made popular by extreme programming. Learning features include chapter summaries, exercises, and projects.

## Object Oriented Software Engineering

## Object-oriented Software Construction

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

A practical introduction to the important new topic of object-oriented design patterns. This book describes pure abstraction-based object-oriented software development - the design and usage of semi-finished reusable components and sybsystems which are based on abstractions of the real world. It starts with an introduction to abstraction based on object-oriented software development. The current state of the art in design patterns is discussed in detail, with an emphasis on the meta patterns approach which describes patterns on a high abstraction level, ignoring language-specific and domain dependent details. Examples demonstrate how the meta pattern approach can be applied in the realm of the GUI application framework E++, which supports GUI development in C++ on several UNIX platforms.

## Object-Oriented Software Engineering

## Classical and Object-Oriented Software Engineering with UML and Java + Code Warrior

## Object-oriented Software Engineering

## Essays on Object-oriented Software Engineering

## Object-oriented Software Engineering

This 1998 book presents the underlying principles associated with object-orientation and its practical application.

## Principles of Object-oriented Software Development

## Object-oriented Construction Handbook

For courses in Software Engineering, Software Development, or Object-Oriented Design and Analysis at the Senior or Graduate level. This text explores the theoretical foundations of software engineering and the principles and practices of various object-oriented tools, processes, and products.

## Object-Oriented Software Engineering: An Agile Unified Methodology

Venturing beyond C++ programming, this text shows how to engineer software products using object-oriented principles. It covers gathering requirements, specifying objects, object verification, defining relations between objects, translating object design into code, object testing, and software maintenance.

## Object-Oriented Software Engineering: Practical Software Development using UML and Java

## Growing Object-Oriented Software, Guided by Tests

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

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 Construction

Test-Driven Development (TDD) is now an established technique for delivering better software faster. TDD is based on a simple idea: Write tests for your code before you write the code itself. However, this "simple" idea takes skill and judgment to do well. Now there's a practical guide to TDD that takes you beyond the basic concepts. Drawing on a decade of experience building real-world systems, two TDD pioneers show how to let tests guide your development and “grow” software that is coherent, reliable, and maintainable. Steve Freeman and Nat Pryce describe the processes they use, the design principles they strive to achieve, and some of the tools that help them get the job done. Through an extended worked example, you’ll learn how TDD works at multiple levels, using tests to drive the features and the object-oriented structure of the code, and using Mock Objects to discover and then describe relationships between objects. Along the way, the book systematically addresses challenges that development teams encounter with TDD—from integrating TDD into your processes to testing your most difficult features. Coverage includes Implementing TDD effectively: getting started, and maintaining your momentum throughout the project Creating cleaner, more expressive, more sustainable code Using tests to stay relentlessly focused on sustaining quality Understanding how TDD, Mock Objects, and Object-Oriented Design come together in the context of a real software development project Using Mock Objects to guide object-oriented designs Succeeding where TDD is difficult: managing complex test data, and testing persistence and concurrency

## Object-oriented Software Engineering

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.

## Advances in Object-oriented Software Engineering

## Classical and Object-oriented Software Engineering with UML and C++

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 with Eiffel

Software -- Software Engineering.

## Object-Oriented Software

Object-Oriented Software Engineering: An Agile Unified Methodology, 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.

## Software Engineering (Sie) 7E

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.

## Object-Oriented Software Engineering

## Object-oriented Software and Engineering

Examines object-oriented methods, practices, terminology, and concepts

## Object-oriented Software Engineering

## A First Course in Object-oriented Software Engineering

Software -- Software Engineering.

## Project-based Software Engineering

Provides full coverage of object-oriented technology, paying equal attention to the underlying theory and to programming practice. The author shows object-oriented concepts at all stages of the software life-cycle. Separate tutorial sections on C++, Smalltalk and Eiffel are included.

## Object-Oriented Software Engineering: An Agile Unified Methodology

## Design Patterns for Object-oriented Software Development

Object-oriented programming (OOP) has been the leading paradigm for developing software applications for at least 20 years. Many different methodologies, approaches, and techniques have been created for OOP, such as UML, Unified Process, design patterns, and eXtreme Programming. Yet, the actual process of building good software, particularly large, interactive, and long-lived software, is still emerging. Software engineers familiar with the current crop of methodologies are left wondering, how does all of this fit together for designing and building software in real projects? This handbook from one of the world's leading software architects and his team of software engineers presents guidelines on how to develop high-quality software in an application-oriented way. It answers questions such as: * How do we analyze an application domain utilizing the knowledge and experience of the users? * What is the proper software architecture for large, distributed interactive systems that can utilize UML and design patterns? * Where and how should we utilize the techniques and methods of the Unified Process and eXtreme Programming? This book brings together the best of research, development, and day-to-day project work. "The strength of the book is that it focuses on the transition from design to implementation in addition to its overall vision about software development." -Bent Bruun Kristensen, University of Southern Denmark, Odense

## Object-Oriented Software Engineering: Practical Software Development Using Uml And Java

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

Software -- Software Engineering.

## Designing Object-oriented Software

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.