Object-Oriented Software Engineering Using UML, Patterns, And Java
**Synopsis**

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).

**Book Information**

File Size: 10187 KB  
Print Length: 800 pages  
Simultaneous Device Usage: Up to 2 simultaneous devices, per publisher limits  
Publisher: Pearson; 3 edition (January 11, 2011)  
Publication Date: January 11, 2011  
Language: English  
ASIN: B008VIXZGY  
Text-to-Speech: Not enabled  
X-Ray for Textbooks: Enabled  
Word Wise: Not Enabled  
Lending: Not Enabled  
Enhanced Typesetting: Not Enabled  

**Customer Reviews**

This is NOT a book on Unified Modeling Language (UML). It's not a book on Object Constraint Language (OCL). It’s also not a book on Capability Maturity Models (CMM),
Class-Responsibilities-Collaborators (CRC) cards, Decision Representation Language (DRL), Extreme Programming (XP), Gantt charts, Issue-Based Information Systems (IBIS), Joint Application Design (JAD), Key Process Areas (KPA), the Liskov Substitution Principle, Model-View-Controller (MVC) architectural styles, Nonfunctional Requirements (NFR) Frameworks, Object Design Documents (ODD), PERT charts, the Questions-Options-Criteria (QOC) model, Requirements Analysis Documents (RAD), Royce’s methodology, Software Configuration Management Plans (SCMP), System Design Documents (SDD), Software Project Management Plans (SPMP), the Unified Software Development Process, User Manuals, V-Models, Work Breakdown Structures (WBS), or any of the myriad other tools introduced in the book. This IS a book to introduce newly-minted programmers to the kind of things, tools, and processes they can look forward to (with either anticipation or dread) in the real world of software development. As the authors state on page viii of the Preface: "We have observed that students are taught programming and software engineering techniques in isolation, often using small problems as examples. As a result, they are able to solve well-defined problems efficiently, but are overwhelmed by the complexity of their first real development experience, when many different techniques and tools need to be used and different people need to collaborate." It’s been many years since I was involved in major software development projects (and those were all in the military).

Download to continue reading...

Beginners, Your Step-By-Step Guide to Learning Java Programming (Java Series) ActionScript 3.0
Design Patterns: Object Oriented Programming Techniques (Adobe Developer Library) Learn PHP
7: Object Oriented Modular Programming using HTML5, CSS3, JavaScript, XML, JSON, and
MySQL Small Memory Software: Patterns for systems with limited memory (Software Patterns
Series) PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide Ada 95: The
Craft of Object-Oriented Programming Microsoft Visual C# 2015: An Introduction to Object-Oriented
Programming Object-Oriented Programming in ColdFusion

Dmca