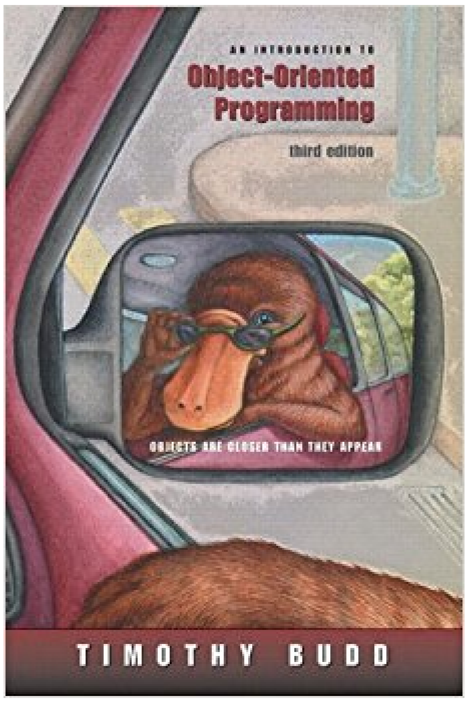
An Introduction To Object-Oriented Programming (3rd Edition)


AN INTRODUCTION TO
Object-Oriented Programming
third edition
OBJECTS ARE CLOSER THAN THEY APPEAR
TIMOTHY BUDD

Synopsis

In An Introduction to Object-Oriented Programming, Timothy Budd provides a language-independent presentation of object-oriented principles, such as objects, methods, inheritance (including multiple inheritance) and polymorphism. Examples are drawn from several different languages, including (among others) C++, C#, Java, CLOS, Delphi, Eiffel, Objective-C and Smalltalk. By examining many languages, the reader is better able to appreciate the general principles that lie beyond the syntax of the individual languages.   This new edition presents examples drawn from a wider range of languages, including Eiffel, CLOS, and Python in addition to the mainstream languages, as well as extensive comparisons between C++, C# and Java. Case studies explore the application of polymorphism in the STL in C++ and the AWT in Java. UML notation and diagrams are integrated and utilized throughout. The book also features advanced sections on design patterns, reflection and introspection, network programming, and the implementation of object-oriented languages.   This book is appropriate for programmers looking to read about the theory behind and functionality of a variety of object-oriented programming languages. It is also useful as a reference.

Book Information

Paperback: 648 pages
Publisher: Pearson; 3 edition (October 22, 2001)
Language: English
ISBN-10: 0201760312
ISBN-13: 978-0201760316
Product Dimensions: 6.5 x 1.5 x 9.1 inches
Shipping Weight: 2.1 pounds (View shipping rates and policies)
Average Customer Review: 4.1 out of 5 stars   17 customer reviews
Best Sellers Rank: #111,007 in Books (See Top 100 in Books)   #50 inÃ Â Books > Textbooks > Computer Science > Object-Oriented Software Design   #69 inÃ Â Books > Computers & Technology > Computer Science > Systems Analysis & Design   #179 inÃ Â Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

Customer Reviews

This slender volume provides a great first taste of object- oriented concepts such as encapsulation and inheritance. An Introduction to Object-Oriented Programming explains all the key technical concepts and goes on to explore the "whys" of programming, such as why a program that one programmer could write in two months probably couldn't be written by two programmers in one month. The reason? Complexity. As a textbook, An Introduction to Object-Oriented Programming does what you would expect--it explains all the key object-orientation concepts clearly and understandably. This book then goes beyond the basics to show why the object concept is strong in terms of design and economics, allowing readers to grasp more than just the technical aspects of the subject. Because examples are in C++, SmallTalk, Objective C, and Object Pascal, this book works well if you're trying to learn object orientation generally, without focusing too much on the mechanics of a particular language. An added attraction is that this book has been recently revised to include some Java information, helping readers to see how object orientation works on the cutting edge as well as in more established languages. This book is useful if you have some experience in programming, but want to expand your knowledge into object orientation by way of clear examples and technical but far-reaching prose. --This text refers to an out of print or unavailable edition of this title.

In An Introduction to Object-Oriented Programming, Timothy Budd provides a language-independent presentation of object-oriented principles, such as objects, methods, inheritance (including multiple inheritance) and polymorphism. Examples are drawn from several different languages, including (among others) C++, C#, Java, CLOS, Delphi, Eiffel, Objective-C and Smalltalk. By examining many languages, the reader is better able to appreciate the general principles that lie beyond the syntax of the individual languages.This new edition presents examples drawn from a wider range of languages, including Eiffel, CLOS, and Python in addition to the mainstream languages, as well as extensive comparisons between C++, C# and Java. Case studies explore the application of polymorphism in the STL in C++ and the AWT in Java. UML notation and diagrams are integrated and utilized throughout. The book also features advanced sections on design patterns, reflection and introspection, network programming, and the implementation of object-oriented languages.This book is appropriate for programmers looking to read about the theory behind and functionality of a variety of object-oriented programming languages. It is also useful as a reference.

I've read a number of introductions to object-oriented design and programming. This one is the best all-around introduction that I have seen. It starts in the real world, with a discussion of how one plans and organizes a task (sending flowers to a significant other) that requires more than a single person to get done. That's a pleasant change from texts that begin with Dauntingly Dry Definitions ("encapsulation", "inheritance", and my favorite, "polymorphism").To the author’s credit, he avoids launching into inheritance until Chapter 8, by which time he has laid enough groundwork to reduce the concept to common sense. Other concepts are presented in a similar manner.Note that this book is a survey book, not an in-depth programming manual. You won’t learn C++ or Delphi, or any of the other half-dozen languages used for the book’s examples. And the book focuses on concepts, rather than implementation. you won’t learn how to implement a Singleton pattern in C#, although you will learn what it is and why it is useful. Finally, the book assumes familiarity with traditional, procedural programming. This is not a Programming 101 text.I would recommend this book enthusiastically as a starting point for anyone making the transition from traditional programming to OOP. If you are moving to the DotNet platform, I have created a list ("So you’d like to ... Transition to DotNet") with some other recommended texts.

its a vey helpful book, for understanding of OOP, without attachment of a programing language, in the book have examples in difference languages

Great

I like the examples that book gave! It’s very descriptive.The book covers from basic to detail, so you will understand the idea of OOP better.

I found the book good, but I didn’t walk away feeling that it provided a good grasp of any but the most general concepts of OO. I would have preferred to see more examples of turning a problem into an OO solution rather than "here’s how to write a class in Java, C++, etc...".

I used this text for an OOP course I taught to college sophomores whose previous background was Pascal and C. I chose this book because, almost uniquely in the field, it was NOT tied to one specific language and that language’s OOP idiom, but rather pointed out significant differences among C++, Java, Smalltalk, Objective-C, and two different Object Pascals in their views of OOP. (I was disappointed by the absence of multi-dispatch languages such as CLOS from the list.) Budd introduces each major principle and programming construct in practical but language-independent terms, then illustrates how that construct is specified in several different languages. I found Budd’s treatment of the basic concepts much simpler, clearer, and less jargon-laden than that in Booch. My students had some trouble, but they got through much of the book, whereas I can’t imagine them wading through Booch at all. I still like the interlingual approach, but I would advise teachers using the book to pick two or three of the languages and simply ignore the rest of the examples, to avoid confusing students too much. I haven't found the ideal text for this course, but Budd is at least a pretty good one.

I've been using classes more as a means of organizing and improving the maintainability, understanding of various applications I've built over the past 3 years (VB). As I am about to develop solutions using the .NET platform (C#, VB.NET), I thought it would do me good to formalize my understanding of OOP/OOD. After reading this text (3rd Edition), I not only formalized my understanding, but was able to see OOP as clearly as I could structured programming (Code Complete). In my opinion, all should use this as the first book before trying to participate/apply J2EE or Microsoft.NET as it will allow you "properly" communicate, design and code systems from abstraction to detail.

I had a great opportunity to know the author of this book in person: I took Dr.Budd's course on Object-oriented programming in Oregon State University in Fall 2000. We used this book in class throughout the term, and thus all of us were compelled to read this book from cover to cover.The principal point I would like to emphasize is that the book really is about the Object-Oriented Concepts. It is not a "guide how to aply OO-principles in C++", it is not an "Object Pascal OO-programming". No. The book is about general concepts of Object-oriented Programming not bounded to any particular OO-language. Although sometimes it was really difficult to understand some of the ideas, the good point was that the author did not try to make the things simpler. If something was difficult to understand - this only meant that it was that Real Blue Thing whose perception makes you a cool programmer.However, the illustrations in the book (at least in the edition I obtained) were not very good, but rather poor. Had they been made by a profeccional designer, I would have rated the book with 5 stars. So far, it's only 4.

Download to continue reading...

Introduction to Object-Oriented Analysis and Design and Iterative Development (3rd Edition) Object-Oriented Programming in C++ (4th Edition) Object-Oriented Programming in Java: A Graphical Approach, Preliminary Edition An Object-Oriented Approach to Programming Logic and Design Beginning Java Programming: The Object-Oriented Approach Python Programming: Python Programming for Beginners, Python Programming for Intermediates, Python Programming for Advanced C++: The Ultimate Crash Course to Learning the Basics of C++ (C programming, C++ in easy steps, C++ programming, Start coding today) (CSS,C Programming, ... Programming,PHP, Coding, Java Book 1) Object-Oriented Analysis and Design with Applications (3rd Edition) The Object-Oriented Thought Process (4th Edition) (Developer's Library) Object-Oriented Modeling and Design with UML (2nd Edition) Tools For Structured and Object-Oriented Design (7th Edition) Object Oriented Software Development Using Java (2nd Edition) Object Lessons for a Year: 52 Talks for the Children's Sermon Time (Object Lesson Series)