The Object-Oriented Thought Process, Fourth Edition

An introduction to object-oriented concepts for developers looking to master modern application practices

Object-oriented programming (OOP) is the foundation of modern programming languages, including C++, Java, C, Visual Basic .NET, Ruby, and Objective-C. Objects also form the basis for many web technologies such as JavaScript, Python, and PHP.
It is of vital importance to learn the fundamental concepts of object orientation before starting to use object-oriented development environments. OOP promotes good design practices, code portability, and reuse—but it requires a shift in thinking to be fully understood. Programmers new to OOP should resist the temptation to jump directly into a particular programming language (such as Objective-C, VB .NET, C++, C .NET, or Java) or a modeling language (such as UML), and instead first take the time to learn what author Matt Weisfeld calls “the object-oriented thought process.”

Written by a developer for developers who want to make the leap to object-oriented technologies, The Object-Oriented Thought Process provides a solutions-oriented approach to object-oriented programming. Readers will learn to understand the proper uses of inheritance and composition, the difference between aggregation and association, and the important distinction between interfaces and implementations.

While programming technologies have been changing and evolving over the years, object-oriented concepts remain a constant—no matter what the platform. This revised edition focuses on interoperability across programming technologies, whether you are using objects in traditional application design, in XML-based data transactions, in web page development, in mobile apps, or in any modern programming environment.

“Programmers who aim to create high quality software—as all programmers should—must learn the varied subtleties of the familiar yet not so familiar beasts called objects and classes. Doing so entails careful study of books such as Matt Weisfeld’s The Object-Oriented Thought Process.”
–Bill McCarty, author of Java Distributed Objects, and Object-Oriented Design in Java

Contents at a Glance

1 Introduction to Object-Oriented Concepts
2 How to Think in Terms of Objects
3 Advanced Object-Oriented Concepts
4 The Anatomy of a Class
5 Class Design Guidelines
6 Designing with Objects
7 Mastering Inheritance and Composition
8 Frameworks and Reuse: Designing with Interfaces and Abstract Classes
9 Building Objects and Object-Oriented Design
10 Creating Object Models
11 Objects and Portable Data: XML and JSON
12 Persistent Objects: Serialization, Marshaling, and Relational Databases
13 Objects in Web Services, Mobile Apps, and Hybrids
14 Objects and Client/Server Applications
15 Design Patterns

My Personal Review:
I found this to be a very good book introducing the Spring Framework. I really wish I had this when I was first learning the Framework some years ago, it would have simplified my work quite a bit and saved me many hours of frustration.
The author, Ashish Sarin J Sharma, is quite knowledgeable and well aquainted with the Framework. His writing is very clear and to the point. I also appreciate the insights he offers when he points out that in the real world one would do this... I would have liked to see a bit more explanation of when one particular technique is better than another in different situations in the real world.

As for the contents, the first chapter, 49 pages, is an obligatory introduction to the basics of the framework and yet manages to get into quite a lot of material. Security, JMX and JMS as well as caching are introduced as well as Dependency Injection, configuration Metadata, and the Spring IoC container. Chapter 1 does not stop there but goes on to discuss &34;Programming to Interfaces&34;. As I recall I first heard this concept and understood it in a talk by Peter Coad in Washington D.C around 1995. Chapter 1 goes on to discuss the techniques of Bean instantiation, Dependency Injection and bean Scope. This seemed like q kind of whirlwind introduction but the author is pretty good about keeping the explanations clear and in noting what it means in the &34;real world&34;.

Chapter 2 is a pretty detailed description of the various means available for configuring Spring Beans. I would hesitate to refer to it as an introduction since it seems very comprehensive to me. chapter 3 is a similarly detailed discussion of Dependency Injection and includes some notes as to when one technique might be preferred over another in real world programming. Chapter 4 covers bean customization and definitions quite well. Annotation driven development is covered in Chapter 5 and was very much appreciated in that I have never competently bought into XML as the end all be all in configuration methods. chapter 6 introduces Database interactions which includes Springs JDBC modules, Hibernate and springs support for JTA. JMS is the focus of chapter 7 and chapter 8 wraps up with Aspect-oriented programming.

In summary this book represents a very good and practical introduction to the the Spring Framework. The examples in the text and the supporting source code are very clear and include practical advice for most real world situations. I recommend this book for those seeking a good solid and practical approach to learning the Spring Framework.

For More 5 Star Customer Reviews and Lowest Price: