Extracted from:

Modern C++ Programming with Test-Driven Development

Code Better, Sleep Better

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina

The Pragmatic Programmers

Modern C++ Programming with Test-Driven Development

Code Better, Sleep Better

Jeff Langr

Edited by Michael Swaine

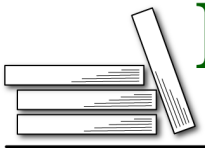
Modern C++ Programming with Test-Driven Development

Code Better, Sleep Better

Jeff Langr

The Pragmatic Bookshelf

Dallas, Texas • Raleigh, North Carolina



Pragmatic Bookshelf

Our Pragmatic courses, workshops, and other products can help you and your team create better software and have more fun. For more information, as well as the latest Pragmatic titles, please visit us at *http://pragprog.com*.

The team that produced this book includes:

Michael Swaine (editor)
Potomac Indexing, LLC (indexer)
Kim Wimpsett (copyeditor)
David J Kelly (typesetter)
Janet Furlow (producer)
Juliet Benda (rights)
Ellie Callahan (support)

Printed in the United States of America.
ISBN-13: 978-1-937785-48-2
Encoded using the finest acid-free high-entropy binary digits.
Book version: P1.0—October 2013

Foreword

Don't let the title mislead you.

I mean, here is a really, really *good* book about design principles, coding practices, Test-Driven Development, and craftsmanship, and they go and give it a title like *Modern C++ Programming with Test-Driven Development*. Sigh.

Oh, don't get me wrong. This *is* a book about modern C++ programming. I mean, if you are a C++ programmer, you're going to love all the code in this book. It's just filled to the brim with really interesting and well-written C++ code. In fact, I think there may be more code than words. Go ahead, thumb through the book. Do you see a page without code on it? Not many I bet! So if you're looking for a good book to teach you modern practices of C++, by example after example after example, then you've got the right book in your hands!

But this book is about a lot more than just modern C++ programming. A *lot* more. First, this book may be the most complete and accessible exposition on Test-Driven Development that I've seen (and I've seen a lot!). Virtually every TDD issue we've uncovered in the last decade and a half is talked about in these pages, from fragile tests to mocks, from the London school to the Cleveland school, and from Single Assert to Given-When-Then. It's all here, plus a whole lot more. Moreover, it's not some academic compendium of disconnected issues. No, this book walks through the issues in the context of examples and case studies. It shows the problems and the solutions in *code*.

Do you need to be a C++ programmer to understand it? Of course you don't. The C++ code is so clean and is written so well and the concepts are so clear that any Java, C#, C, or even Ruby programmer will have no trouble at all.

And then there are the design principles! For goodness sake, this book is a design tutorial! It takes you on a step-by-step voyage through principle after principle, issue after issue, and technique after technique. From the Single Responsibility Principle to the Dependency Inversion Principle, from the Interface Segregation Principle to the Agile principles of simple design, from

DRY to Tell-Don't-Ask—this book is a gold mine of software design ideas and solutions. And again, these ideas are presented in the context of real problems and real solutions in real code.

And then there are the coding practices and techniques. This book is just chock-full of them, from small methods to pair programming and from coding katas to variable names. Not only is there a ton of code from which to glean all these good practices and techniques, but the author drives each point home with just the right amount of discussion and elaboration.

No, the title of this book is all wrong. It's not a book about C++ programming. It's a book about good software craftsmanship that just happens to use C++ as the language for its examples. The name of this book should really be *Software Craftsmanship: With Examples in Modern C++*.

So if you are a Java programmer, if you are a C# programmer, if you are a Ruby, Python, PHP, VB, or even a COBOL programmer, you want to read this book. Don't let the *C++* on the cover scare you. Read the book anyway. And while you are at it, read the code. You won't find it hard to understand. And while you are learning good design principles, coding techniques, craftsmanship, and Test-Driven Development, you might also discover that a little C++ never hurt anybody.

—"Uncle Bob" Martin
Founder, Object Mentor Inc.