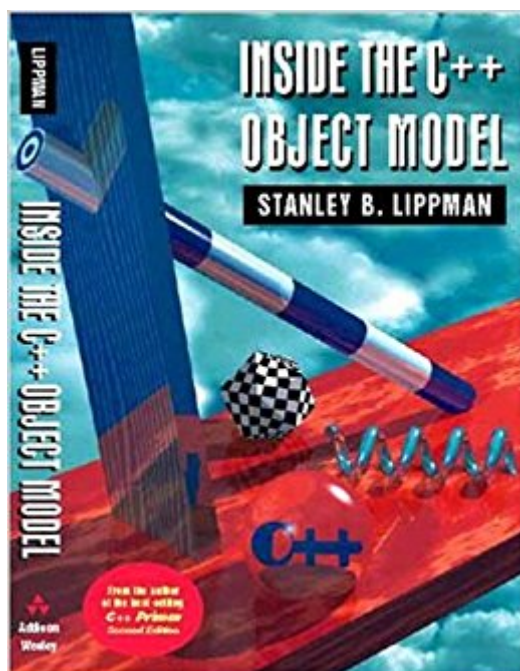# Inside The C++ Object Model

Synopsis

Aimed at the beginner/intermediate C++ programmer who wants to understand the semantic implications of the C++ object model and how the model affects their programs, Inside the C++ Object Model explains where overhead costs reside, and what they actually consist of.

Book Information

Paperback: 304 pages

Publisher: Addison-Wesley Professional; 1 edition (May 13, 1996)

Language: English

ISBN-10: 0201834545

ISBN-13: 978-0201834543

Product Dimensions: 6.3 x 0.7 x 9.1 inches

Shipping Weight: 1 pounds (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 starsÂ Â See all reviewsÂ (17 customer reviews)

Best Sellers Rank: #330,449 in Books (See Top 100 in Books)   #137 inÂ Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C   #216 inÂ Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C++   #228 inÂ Books > Computers & Technology > Programming > Microsoft Programming > C & C++ Windows Programming

Customer Reviews

...truly excellent. If you are seeking to truly UNDERSTAND C++, not just increase your familiarity with the syntax, INSIDE THE C++ OBJECT MODEL is one of the first books I would recommend reading. Stan Lippman, besides being one of the most C++ - knowledgeable humans on the planet, is an excellent writer. His style is quick, to the point, and non-repetitive (to some, this might indicate a difficult read, of course). INSIDE examines the features and additions C++ brought to the world of C - from the inside - demonstrating the creation of objects, instantiation of templates, and more through comparisons with C-based code generated by the CFRONT compiler. For the reader with a good, intermediate understanding of C++, this will be a revelation; seeing a representation of an object - its vtables and internal structure - as a construct that actually exists in memory is simultaneously enlightening and delightful. Few 1500-page texts TEACH so much as this book does in 270-odd pages. One of the best.

This book has a very good explanation of the trade-offs involved in implementing C++.Problem areas however are: * Cursory explanation of how templates, RTTI and exceptions are handled. *

Reads more like a `cfront' rationale with examples from other compilers interspersed. That doesn't mean that the examples are limited. Just that it reads more like the author's experiences, rather than a totally objective view. * Doesn't clearly separate run-time effects that are artifacts of a particular implementation from things required by the Standard. * Lots of typos. Many of the figures don't co-incide with the text -- the text explains with one set of variables, and the figures show some other set, with some names transposed.

I have thought that I had enough knowledge of C++, but after reading this book, I know I was wrong. Detailed discussion of constructor, data member, virtual table, multiple inheritance etc, there are a lot of mysterious details in C++ covered in this book. Yet this book is only 280 pages, which means there are very few redundant words seen in a lot of other tech books.Recommend reading it with a compiler. I use Visual C to compile sample code and generate the asm file to see the real code compiler generates. It is rewarding.

If you use C++ in real time environments, you need this book. Real time programmers need to understand "what is going on under the covers." This is really the only book I've found that explains it in depth.Reading this book is important if you are trying to decide whether to code in C or C++. You need to understand the overhead of C++, where it comes from and, at times, how to avoid it.When debugging an understanding of how the compiler translates C++ code into machine language is essential. Lippmans book does not delve into code generation but it does provide a framework for understanding how the C++ object model is represented in machine language.I'll be the first to admit that Lippman can go off on to some arcane tangents. He can spend a few pages discussing details that really have minimal effect on code execution. But I'm willing to forgive these diversions because so much of his book has contributed to my understanding of C++ and how it works.

This book is excellent in telling the underlying implementation of C++ and how the cfront compiler changed C++ into C. While this is not an everyday task and one that most people could care less about, it is truly one of the most important things you could learn because it shows an important comparison between the code you don't have to write and the execution time you lose. Every C++ programmer should have read this book at least 10 times to decide what structures should be used under what circumstances

Have you ever wondered how RTTI or virtual functions work? What are the rules for determining if a default constructor is provided. This book will answer those questions and more.The C++ Object Model gets to the "how" of C++. It covers the details of how virtually (ha!) every language features is implemented and what rules a compiler uses to make decisions.This book is not an easy read. You need to be very comfortable with C++ and be willing to work through lots of code. The explanations are ok, but could be a bit much if you aren't familiar with the full C++ lexicon. That said, if you want to take you C++ knowledge to a new level, this book is the ticket. There is nothing else like it available.

This is the second book one should read after reading books like C++ programming language or C++ primer as first book on C++. This book is a complement to the above titles. It covers exactly that portion of the language which remains hidden in other books. The "real C++" is here, in this book. One cannot come across an equivalent of "Inside C++ object model". One cannot be a complete C++ programmer without knowing the facts mentioned in this book. It's unique in the sence that it carries with it irreproducible experiece and enjoyment Stan Lippman had while working on the wonderful C++ compiler. Reading the book was an unforgettable experience for me.

Download to continue reading...

Inside the C++ Object Model Object Relations Couple Therapy (The Library of Object Relations) Object Relations Individual Therapy (The Library of Object Relations) Inside CORBA (Addison-Wesley Object Technology) Model Cars You Threw Away: Guide to Pricing and Collecting Obsolete Diecast Model Cars Basic Painting and Weathering for Model Railroaders, Second Edition (Model Railroader Books: Essentials) Basic Model Railroad Benchwork (Model Railroader Essentials Series) Building N Gauge Model Railway Layouts (Building Model Railway Layouts Book 1) Introduction to Model Railroading (Model Railroader's How-To Guides) Painting Backdrops for Your Model Railroad (Model Railroader's How-To Guides) Designing & Building Multi-Deck Model Railroads (Model Railroader) Model Railroad Electronics: Basic Concepts to Advanced Projects (Model Railroad Handbook) Ho Scale Model Railroading: Getting Started in the Hobby (Model Railroader Books) Building a Ready-To-Run Model Railroad (Model Railroader's How-To Guides) How to Build Model Railroad Benchwork, Second Edition (Model Railroader) Basic Structure Modeling for Model Railroaders (Model Railroader Books) Model Locomotive Boilermaking (Model Engineering Guides) The Jennifer Nicole Lee Fitness Model Diet: JNL's Super Fitness Model Secrets To A Sexy, Strong, Sleek Physique Microsoft Windows Security Inside Out for Windows XP and Windows 2000 (Bpg--Inside Out) Microsoft® Windows® XP Inside Out (Bpg-Inside Out)