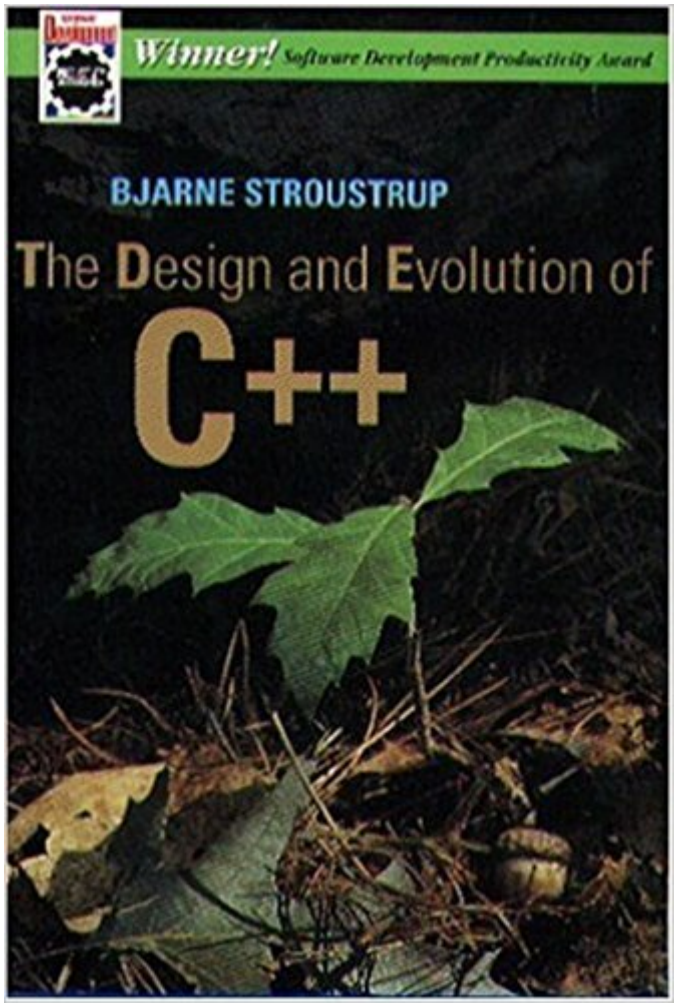
The Design And Evolution Of C++



Synopsis

This book focuses on the principles, processes and decisions made during the development of the C++ programming language. As the inventor of the language, Stroustrup presents his insight into the decisions which resulted in the features of C++ - the praised, the controversial and even some of the rejected ones. By writing this book the author presents his object-oriented programming philosophy to the interested programming community. His vehicle is the C++ language but his focus is on real object-oriented programming language development for the working programmer rather than as a abstract approach to the OOP paradigm.

Book Information

Paperback: 480 pages

Publisher: Addison-Wesley Professional; 1 edition (April 8, 1994)

Language: English

ISBN-10: 0201543303

ISBN-13: 978-0201543308

Product Dimensions: 6.2 x 1.1 x 9.2 inches

Shipping Weight: 1.5 pounds (View shipping rates and policies)

Average Customer Review: 4.7 out of 5 starsÂ Â See all reviewsÂ (29 customer reviews)

Best Sellers Rank: #271,645 in Books (See Top 100 in Books) #182 inÂ Books > Computers & Technology > Programming > Languages & Tools > C & C++ > C++ #214 inÂ Books > Computers & Technology > Programming > Microsoft Programming > C & C++ Windows Programming #408 inÂ Books > Computers & Technology > Programming > Software Design, Testing & Engineering > Object-Oriented Design

Customer Reviews

This is probably the most well written B. Stroustrup book. Without dwelling too much on the arcane, Bjarne goes over all of the reasons for the additions to the "C" language to create C++. Bjarne comes off as a guy just trying to get a job done and yet do what is right for the rest of the programming community. Its a nice change from the almost religious furor discussions that occurr on USNET.There are still open issues among the users of C++, people who want a feature found in another language, or wish that their personal idea would be incorporated into the general language. Before posting a proposal to comp.std.c++ you should read this book. There you will most likely find a discussion on the idea and why it is either not implemented, or was rejected. Then you can organize your counter argument without wasting everyone’s time. (Also one of the first counter posts

will be a citation to this book.) It's not that C++ is the perfect language, it isn't, after all my pet idea of overloading operator.() was rejected, but in ammending the ISO99 C++ standard you need to know what has already been discussed. So we can go forward without rehashing.Intermediate C++ programmers would also benefit from the discussions on casting, use of private/public/protected inheritence and scoping, and exceptions. Bjarne goes over why these things changed over time and what problems these features are intended to solve.

This book is very interesting in that it doesn't tell you how to program in C++ but rather highlights why C++ is the way it is today. It starts with the very roots, an extension to the C language ('C with classes'') Bjarne devised back in 1979, because he faced a software engineering problem at the time where all currently available tools seemed inappropriate. This highly real world oriented design attitude was kept throughout the evolution of C++ - Bjarne specifically didn't want to produce an 'academic' language. This view and the absolute necessity for C compatibility and efficiency explain lots, if not all, of C++s more ugly syntactic and semantic constructs. While the book has chapters dealing with very specific parts of the language, I found the philosophical chapters the most interesting. These explain the author's personal views on programming and design in general and consequently why certain things were accepted or rejected into C++. Bjarne stresses the point that C++ was designed from the beginning to be a 'multiple paradigm' language. Object oriented programming was never meant to be, and is not, the only valid - holy grail - style of programming, that many make it out to be. It's quite frustrating to see features devised ten years ago still not properly supported by the current crop of compilers, templates for example (export anyone?).The book is not for the novice programmer, but for the experienced C++ user who wants to know the whys behind the language. While a novice might be interested in that information too, it is not an advisable lecture for those readers, since they might easily get confused with the source code examples showing directions in which C++ did not evolve.To quote one of the design goals: 'C++ is a general-purpose language designed to make programming more enjoyable for the serious programmer' - I think it succeeded.

While this title probably won't help your development skill, it provides a great deal of insight into the design of C++. Parts of C++ may seem a bit odd (crufty, overly complex, however you want to think of it). In this book Stroustrup clearly explains the motivations and tradeoffs that went into every feature of the language. While you may not agree with the decisions, understanding the thought process behind them is incredibly interesting and will give you a better appreciation for the

language.

I am a CS student and this is the one C++ book I always carry with me. I find it useful as an explanation to "how and why" C++ is the way it is, and also useful as a reference. The 34 page index, which is about 1/12th of the book, is exhaustive. Almost any aspect of the C++ language is listed along with a description of how it works and why.This book offers key insight into the class layout in memory, vtables, multiple inheritance and the type-checking system.Bjarne talks about what he wanted to add, but was not allowed to. He also explains how C++ was written mostly in C++, which I found weird and amusing.If you’re looking for info on the STL, this book has none; this is strictly C++ language related. If you’re interested in computer language development or compiler writing, this book would be wonderful.Lastly, if you plan on teaching C++, you should really read this book so you can understand the language well enough to explain it.