Review

Borland C++ 3.1 object-oriented programming
Cantù M., Tendon S., Bantam Books, Inc., New York, NY, 1992. Type: Book (9780553370867)

Date Reviewed: Aug 1 1993

Full Text

Related Topics
Browse | Alerts
C++ (D.3.2 ...) Add
Object-Oriented Programming (D.1.5) Add
Programming Environments (D.2.6) Add
Manage Alerts | More Alerts

Although my instinct is to flee from the dozens of (generally poor) books concerning specific implementations of C++ for microcomputers, this "official Borland book" gave me a bit more hope. It includes a foreword by Borland CEO Philippe Kahn, and Cantu` and Tendon, Borland, and Bantam cooperated closely during its writing. The book has elements of advertising, as its attention rarely strays from the Borland products, but it is a serious effort, and it includes some discussion of other object-oriented systems. So, while I am disappointed with the book, some of my criticism is of the general attempt to cover both the language in general and a specific implementation.

The book is essentially an extensive tutorial for those using Borland's C++ 3.1 development environment for DOS or Windows. In 16 chapters, it covers the gamut of C++ programming using Borland products, including the operation of the Integrated Development Environment, compiler, linker, and maker; object-oriented philosophy; both C and C++ language constructs; object-oriented methods in C++, including single and multiple inheritance, polymorphism and dynamic binding, abstract classes, and genericity; and finally both standard C++ and Borland-specific class libraries for both DOS and Windows. Program examples tailored to the Borland environment and libraries are tightly interwoven with the discussion of general concepts.

The attempt to cover so much material tutorially makes it difficult to identify readers who would benefit most from this book. Although the discussion of object-oriented techniques is strong, the focus on Borland makes it useless as a general book on object-oriented programming or C++. While Cantu` and Tendon claim that no knowledge of C is necessary, the C novice will be bewildered by the slew of concepts and language constructs, many of which are neither developed fully nor presented in a logical order, and some of which are introduced before their explanations, which appear pages later if at all. The C++ language constructs are not developed in enough depth for the knowledgeable C programmer, however. Details of specific keystroke sequences and menu and installation options compete with concepts for the reader's attention. An experienced C++ programmer looking specifically for a better reference on the Borland tools and libraries will find the other material pedantic and unnecessary. Even so, much of the environment-specific material is redundant if one has Borland's documentation.

The book is not well written. Especially when discussing the C family of languages, clear use of terms in a logical order is crucial. On page 67, however, we are told that local scope is within a block, but not what a block is; on page 75 some legal but implicit type casts are described as "incorrect," while a note explains that no error will be generated; the section on pointers and arrays, an infamous subject, is confusing and insufficient; and on page 136, the **new** operator is introduced with essentially no explanation. Some dubious programming practices are also demonstrated, such as **#include**ing .cpp files, and the book uses a style of pointer declaration (char* p, rather than the standard char *p) that, while it is attractive for simple declarations, becomes entirely impossible for declarations of any complexity.

I found a number of copyediting errors and inadequacies in the index. For example, only one entry is provided for the term "virtual," and that is for the **virtual** keyword; although both virtual functions and inheritance are prominent in the text, they are omitted from the index, while other instances of the keyword are missed.

The book has redeeming values, though. The many examples are generally useful and illustrative. The discussion of design methodology (including general object-oriented design and event-driven and "contractual" methods) is clear, the examples are cogent, and useful references and quotations from the literature are provided. Many details of C++ methods are mentioned, including the relative advantages and disadvantages of polymorphic versus template genericity, and the dependency relations created by class inclusion and inheritance. The bibliography is also helpful.

Perhaps the most useful parts of the book are in the last few chapters, which discuss the architecture and use of Borland's Turbo Vision and Object Windows interface libraries for text-based DOS and graphical Windows environments, respectively. The examples here will be helpful for those preparing to use these libraries.

Cantu` and Tendon say "This book is not the sum of many manuals, but a road map that C++ programmers can use to put everything in its place" (p. xvii). Perhaps the lesson is that C++ programmers would be better served by a set of distinct works: a language tutorial, a language reference, a book on design methods and techniques, appropriate library references (with examples), and finally a dedicated reference manual on the specific operation of the development environment.

Reviewer: Cliff Joslyn

Review #: CR116853

Would you recommend this review? yes no Enter

Other reviews under "**C++**":

	Date
Beginning C++17: from novice to professional (5th ed.) Horton I., Weert P., Apress, New York, NY, 2018. 771 pp. Type: Book (978-1-484233-65-8)	Sep 16 2019
C++ templates: the complete guide (2nd ed.) Vandevoorde D., Josuttis N., Gregor D., Addison-Wesley Professional, Boston, MA, 2017. 832 pp. Type: Book (978-0-321714-12-1)	May 24 2019
Computer programming with C++ Pimparkhede K., Cambridge University Press, New York, NY, 2017. 1014 pp. Type: Book (978-1-316506-80-6)	Jan 19 2018

more...