

Len Dorfman's C++ By Example

Unless you're perfect, or a masochist, it's probably best not to read a new programming book right after you've pushed a project out the door. Each twist of spaghetti code is fresh in your mind, and each of the author's sparkling clean examples only serves to deepen the post-partem depression. If it's a good book, then at some point, one of the author's explanations will cross paths with one of those freshly made mistakes and you'll say to yourself, "Damn, shoulda' done it that way."

For me, that never happened with Len Dorfman's **C++ By Example**. A presentation of the author's C++ course, the book is aimed at DOS programmers with a couple of years of C experience. It breaks down into roughly two parts, text and the example code where he develops a DOS character-mode user interface class library. The text sections, chapters 1, 2 and 4, give a good, brief introduction to OO analysis and design. Chapters 1 and 2 give the problem statement and the structured-query method which he then uses to start solving the problem. Chapter 4 describes the process the class library design went through before assuming its final form.

The "by example" in the title is not an exaggeration. The entire remainder of the book is code. By my conservative estimate, the book is at least 70% source code. As an example, the chapter presenting the Window class, perhaps the key class in a UI class hierarchy, consists of 36 pages with only 6 paragraphs of text - 4 introduction and 2 summary.

The classes (Message, Event, Window, Dialog, MenuBar) will be familiar to anyone who's used any of the current crop of class libraries, (OWL, MFC ...). Within the limitations of character based DOS, they're all here. If you haven't used something like MFC, Dorfman's class lib would provide good preparation for that more arduous pursuit.

The example code, 1700 lines including whitespace and comments, is included on a disk that comes with the book. It was written for Borland C++ for DOS, which, of course, I don't use. Porting to Watcom 10.0 was relatively painless, with all but the last two programs taking only a couple of hours total.

There is no discussion of syntax anywhere here. You need to have written at least some C and probably some C++ to benefit from this book. This approach seems best suited to programmers who come at the topic from having written some C programs under DOS. Since the example the book is written around is a user interface, it presents issues that will be very familiar to anyone who has tried to write a decent-looking, DOS character mode program.

The example code is the book's strength, but it is also its biggest weakness. One of the great strengths of object oriented code is that, when well-written, the objects 'look like' things in the real world and are thus much easier to

understand. The best intro to OOP I ever read used the game of blackjack as its example, with easily accessible objects like cards, decks and games. Using nerdy objects like mice, monitors and keyboards as your examples, needlessly throws away that advantage.

The choice of a DOS user interface as the example has another side-effect - a complete dependence on BIOS calls that fills the code with stuff like:

```
union REGS ir;  
  
ir.h.ah=0x03;  
  
int86(0x10,&ir,&or);
```

Everyone who finds this readable and enlightening, raise your hand. To my mind, these BIOS calls are not only ugly and unnecessary, but also an impediment to the stated goal of teaching C++, not DOS.

That criticism is heartfelt but shouldn't be taken *too* far, because, for what it is, the book is very well done. Having chosen this particular problem, Dorfman presents a laudably pragmatic and complete solution. His classes are logical, simple and clean. They also work, something you can't always assume with code you find in books and magazines. If I needed to write a DOS program with a minimal UI, I wouldn't hesitate to use this code.

Dorfman is pragmatic to a fault, dwelling at length on a couple of situations where OO purity went out the window in order to make the program work. Rather than anticipating problems, Dorfman works around them as they occur. He doesn't gloss over the essentially messy nature of programming either. At one point, for example, he shows seven different cuts at one object hierarchy. He is never dogmatic and readily admits that doing it some other way would work (an admirable personal quality that does not occur naturally in programmers). His style is persistently candid and relaxed and you can almost picture him standing next to the overhead projector adjusting the transparency.

Conclusion

I hate to pan this book, because the example code is well-written and works. If a subordinate turned this code in as work product, you'd pat him on the head and give him 1000 extra stock options at bonus time. Unfortunately, as a learning tool, **C++ By Example** is a bad deal. This is a harsh statement, but the truth is that there is plenty of readable, high-quality C++ code that can be had for free. Had Dorfman included a number of well thought out exercises, emphasizing **using** his class library at the end of each chapter, the book might make the basis for a hands-on, continuing-ed C++ course, which is in fact where it originated. As it stands, though, it reads more like a long magazine article inflated to book length by whitespace and BIOS calls.

Book info

Publisher:

McGraw-Hill Object Technology Series

price: 39.95

pp 300

disk included

lines of code including comment and whitespace ~ 1700