One Last Compile...

I've just come back from an advanced Delphi course. I get huffy if I don't get sent on a course every six months. I mutter darkly about career development and up-to-date skill sets, and stare meaningfully at the jobs pages of *Computing* and *Computer Weekly* until I finally get permission to do one. I don't want anything difficult, just a good excuse to get out of the office for a bit.

I thought a nice week or so designing Web pages would be just the ticket, but was foiled when it was pointed out that we didn't have a Web site. I was a bit hurt when it was suggested that some more Delphi training could be in order, as I'd pretty much taught myself Delphi and thought I'd done a fairly good job. I'd written two applications in it, and after a heroic marketing effort we'd even managed to sell a copy of one of them. "You've done very well on your own," said my boss tactfully, "but perhaps it's a good time to consolidate what you've learnt. You know: write a few components, a DLL or two. And, ah, maybe improve your error handling a bit."

After my initial sulk, I resigned myself to getting to grips with Delphi properly, which meant Objects. I knew OO was important, and that I really should devote some time to learning about it, but somehow I always got sidetracked by important interface issues, like if my Bevels should be Raised or Lowered or if Edit Boxes using Algerian Bold Italic were really as unpleasant for the user as everybody else said they were.

But I liked the idea of writing my own components. It seemed to me that it would be very cool and impressive to be able to immortalise my code on my very own components page, even if I didn't have a completely clear idea as to which bits, if any, were really worth saving. The bits that broke least often, I supposed.

I'd fibbed massively when filling in the application form, cheerfully ticking every box in the 'pre-requisites' column. Now I listened in panic on the first morning as the unfortunate Carol expounded on publishing properties, on the difference between `override` and `virtual`, and how in some cases I needed a destructor but in others I didn't. I could only nod feverishly when she asked if I knew the difference between classes and objects and instances. Because an understanding of that is absolutely essential, she said.

I don't remember much else after that. Two days later I emerged clutching a disk of half-written components, a file of notes with lots of frantic doodles around the margins, and a free mug. I've since written my first component, which sits proudly in my very own page of the components palette, although I have my doubts as to just how much time an Edit Box with its default font set to Algerian Bold Italic will really save me. But objects are still a bit of a haze, and I suspect, despite Carol's best efforts, I'll carry on bodging along just like I did before.

I'm not telling anybody else that, of course. The trick with courses is to come back with at least ten new bits of jargon, and ideally a new way of drawing diagrams that nobody else can understand. If anybody looks like they're coming over to ask me to do something, I can now scribble little cloud diagrams with lots of lines coming out of them, and mutter to myself about virtual constructs. They cough nervously, and head for the drinks machine instead. And I go back to reading the jobs page in *Computing*...