

Byte - A Brief History of Programming Languages

Comments on:

Byte Magazine (1995). A Brief History of Programming Languages. *Byte Magazine*, September 1995 (20th Anniversary Issue).

Good Reflections

I was intrigued by the notion of “hand-compiled” language - Short Code. Given such a huge restriction, what was the nature of computation that the language was used for?

It was more sensible to write short code than to write machine code directly. By writing in short code, you could think about the problem. Given that the translation to machine code was relatively straightforward, it helped you avoid some problems. And it's not a huge restriction; anything you can write, you can write in machine code.

In 1960, ALGOL60, the first block-structured language appears and becomes the most popular language in Europe in the mid- to late- 1960s. Later, in 1968, ALGOL 68 appears which is as described a monster compared to ALGOL 60 and some members of the specification committee were protesting its approval. What made ALGOL 60 such a success and ALGOL 68 as described by the author a "monster"?

As Sammet notes, Algol60 was notable, not least, because it included an elegant formal syntax. Unfortunately, Algol-68 took that to yet another level, and it had a meta-syntax. While beautiful in theory, it is clear that few could understand it.

I think this article was a little too wordy for me.

Funny. Don't count on funny getting you far in most responses, though.

I'd always been told the history of computing went back into the 1890's. It's hard for me to picture a way "computing" can exist without some kind of "programming", but Sammet and Wiki seem to both take for granted that nothing before FORTRAN was a language, though the Byte.com article starts a bit earlier.

You can send the history of computing back pretty far. Babbage's Analytical Engine had a language with which to describe programs, and Lady Augusta Ada, Countess of Lovelace, certainly wrote some interesting programs in that language. The focus of each seems to be on high-level languages, rather than assembly and machine languages. In some sense, it's a software/hardware dichotomy.

Many of the popular programming languages listed in this article underwent a number of revisions throughout the years. Languages such as ALGOL and FORTRAN were in revision for up to 10 years. C++ also had a number of versions, but has remained static for a while. However, C seems to remain unchanged since its initial conception. While I have no experience in ALGOL nor FORTRAN, C is clearly simpler than C++. Is it this simplicity that has kept it from needing to be redesigned?

Well, there were a variety of changes to C. The article mentions (explicitly or implicitly) three+ versions: The original C from 1972, the C described in 1974 (often called K&R C), and ANSI C from 1989. One could also conclude that C++ is an extension to C.

Nonetheless, the changes do seem a bit smaller. One might argue that Kernighan and Ritchie just got it right. The primary design goals of C (which I call jokingly “high-level assembler”) also don’t seem particularly amenable to change. It may also be that those who wanted change focused on changing C++, rather than C.

The first reading, from byte.com does not really offer much contextual information. It describes some of the important aspects of each language, but not why it was so important. This article also takes a broader definition of programming languages, as it includes the work of Konrad Zuse, whereas the other two articles focus mainly on high-level languages.

There’s not much for me to respond to hear, and it’s a bit at the bottom in terms of acceptable, but it is acceptable.

Less-Good Reflections

I didn’t know computer scientists were so obsessed with acronyms.

Boy, you haven’t been paying attention.

This article also included lots of history that I previously never knew anything about. For example, I had no idea Scheme was so old.

I would like to hear a bit more about what you thought about the age of Scheme, and why you find its age surprising.

Franz Lisp is a great name for a language.

Unenlightening.

How come this article does not include information about Java? I read somewhere that Java was developed in the early 1990s.

A question you should be able to answer on your own. Java had not had widespread distribution by 1995. In fact, it may still have been Oak at that time. Let’s see ... less than a minute on Wikipedia shows me that “The first public implementation was Java 1.0 in 1995.” A new language is unlikely to be described in a history.

Byte.com’s chronological list of events and big names in the computer industry doesn’t have a lot of detail but lets you see the progression of languages like Smalltalk, ALGOL, and SNOBOL. Almost all events on the timeline are either names of people that developed compilers and programming languages. BYTE even helped a language hit “the scene like a nuclear blast”.

I’d like a bit more reflection.

I didn't know most of the languages mentioned in the text.

I'd like a bit more reflection here. Perhaps even just "It is surprising that in just ten years, the languages that people are expected to know have changed so significantly."