American Scientist online
THE MAGAZINE OF SIGMA XI, THE SCIENTIFIC RESEARCH SOCIETY

see full issue: September-October 2006

## LETTERS TO THE EDITOR

To the Editors:

Regarding "The Semicolon Wars" by Brian Hayes *(Computing Science,* July-August), the article is interesting but has a basic proposition that, in my opinion, is wrong: that computer languages are products of design and not evolution.

It could be argued that nothing that people create can be the product of evolution, but I think of it differently. Human beings provide the environment for development, but that development is clearly evolutionary.

The creation of human language, including computer language, is an evolutionary process. Language changes over time and adapts to its environment. The fact that computer languages are "designed" does not, to me, imply that they are not evolving.

The parallels are amazing between evolving nature, evolving natural language and computer language. Computer languages even have their evolutionary dead ends, just like natural evolution processes. In fact, the diagram in the article of the links between programming languages looks remarkably like a map of an evolutionary tree.

Mr. Hayes provides convincing evidence that these languages are evolving and even cites the mechanisms by which this takes place. He makes the statement that his favorite computer language has changed over time and been "… augmented, overhauled, updated, split into multiple dialects, then reassembled.…" He also states that "A few programming languages—most notably Fortran and Lisp—seem to be all but immortal.…" Just like some really successful species, they have survived.

There has never been, in human history, a riper laboratory for the exploration of evolution and linguistics than the development of computer languages. It's well documented, current and in an era of rapid movement. It's an interesting process that I've been watching for years. It seems to me to be obviously evolutionary.

Stephen L. Robinson
Fresno, CA

Mr. Hayes responds:

I'll grant that computer languages do change and diversify over time, that they can be arranged in a sort of family tree, and that their "fitness" is tested in the environment. But is that enough to qualify as evolution? There's one crucial element of Darwinian evolution that's missing here.

In biology, the variations that provide the raw material for natural selection are randomly generated. Nature has no engineer consciously and deliberately trying to improve a species; change is brought about by blind mutation. In contrast, the changes introduced into programming languages are surely not random; they are carefully crafted to address perceived flaws or to improve performance in some way or perhaps to test an idea that's thought to have promise. As I said in the column, "The creators of a new programming language are not just adding variety for its own sake; they are trying to make something demonstrably better."

I believe this difference between biological evolution and the engineering processes is important. For one thing, it implies quite different ways of exploring the space of all possible designs. Random mutation has the advantage that it will try anything, however implausible it might seem *a priori,* and thus evolution can often discover solutions that human designers might well ignore or dismiss. But engineers have an advantage on their side, too: They are not constrained to proceed in small steps from known solutions into immediately adjacent territory. They can make giant leaps.