

Translated from Jordi Pérez Colomé's interview in El País: [Nuestra civilización depende igual del software que del agua](#).

“Our civilization is as reliant on software as it is on water”

If computers failed, we would die of starvation according to this legendary Danish computer scientist, the author of one of the world's most commonly used programming languages.

If there were a prize for the person having the greatest impact on our lives that remained practically anonymous, Bjarne Stroustrup would be a strong candidate. Not only is [Stroustrup](#) unknown to the public, his illustrious creation – the C++ programming language - remains obscure outside IT circles. However, it's everywhere: “The best approximation would be to say that everyone in the world has used it and many are using it all the time. And you never see it” says Stroustrup.

Stroustrup created C++ in the 80s. Today it is the fifth most popular programming language behind JavaScript, Java, C and Python, according to a survey by HackerRank of 71,000 developers in 100 countries.

The programmer was born 68 years ago in Aarhus (Denmark). Following graduation, he relocated to the UK and the United States, where he pursued his career. The Carlos III University in Madrid has just awarded him an *honorary doctorate*.

Question. Of the things that surround us, what has been written in C++?

Answer. You're recording this on your telephone. The signal processing is in C++. A large number of the apps are in C++. If you came here by car, some of the control systems - fuel injection, steering, breaks - could be in C++. If you've watched television today, it could also be there: the cameras and communications systems will probably have C++. A lot of the software that your mobile uses to connect to the transmission tower is probably C++. The GPS system has some C++. It's like household chores: you only notice it if something is not right.

Q. When you began, it didn't seem like you would become a great computer scientist. You were not an outstanding student.

A. I was a good student. I needed a bit of luck, though. Perhaps it's more accurate to say that the harder I worked the luckier I was.

Q. Today we think of 15-year-olds when talking about programming wizards...

A. I didn't even see a computer until I was 18 or 19 years old.

Q. When you finished your studies you didn't want to become a teacher. You saw it as a trap.

"I've seen hordes of brilliant, working-class kids with no role models to follow that ended up in the only trade that they knew for smart people with an education: schoolteacher"

A. I've seen hordes of brilliant, working-class kids with no models to follow that ended up in the only trade that they knew for smart people with an education: schoolteacher. When I was 17 or 18 years old, I thought that it was wrong that the previous work of a teacher was that of being a student. There had to be something else in their career journey. I dreaded becoming a teacher. I wanted to make things.

Q. "Our civilization depends on software", you've said.

A. And that dependency is growing. We would die of starvation if software stopped working. A city like New York has food for no more than two or three days. If trucks began to malfunction, trains came to a standstill or if traffic signals broke down, all of these being computerized systems, people would starve. I'm not joking. Agriculture depends on computers. We are as reliant on software as we are on water.

Q. On winning [the Draper Prize](#), you called for society to better acknowledge the importance of engineers. Artists and scientists are more renowned than engineers. Why?

"Science is wonderful, but it never does anything on its own. Someone then has to use the science to make a device that we are able to use"

A. When people think about art and science, they usually forget engineering. Science is wonderful, but it never does anything on its own. Someone then has to use the science to make a device that we are able to use. You have to apply the science. People think about Einstein and that's fantastic. But they don't realize that it took a long time to create the GPS system that uses his ideas. Different societies value engineering differently. If you're an engineer in Germany, you're somebody; if you're one in the UK, people think you drive trains. That's an exaggeration, but only partly so.

Q. An engineer's point of view is practical: You said that "there are two types of [programming] languages: ones that everyone complains about and those that nobody uses".

A. Thomas Edison said that invention is 1% inspiration and 99% perspiration. It means that you may have a good idea, but it takes a lot to turn it into something that people can benefit from. People forget that 99%. If, for example, you write a story to inspire children, it's always about that 1%. That's the exciting bit.

Q. Another phrase: "The only thing that grows faster than computer performance is human expectation". Wouldn't knowing how to program help to avoid this?

A. Yes and no. It's good for people to know something about everything: science, history, math, coding. But the danger lies in people who have studied math at high school thinking that they are mathematicians. I know what I'm talking about: I have a

degree in mathematics, I've met great mathematicians and I'm not in the same league. On the other hand, there is an enormous number of teenagers, nearly always boys, who think that they know everything about programming and building systems so that nobody can teach them anything. A little bit of knowledge is a dangerous thing. It can make you think you're an expert when you're not.

Q. You learned 20 programming languages before finishing university. It's even said that in the doctor's waiting room you would flick through a manual and later use it for a couple of weeks to learn it. If that's still the case?

A. You can't do that today. Languages are much more complicated. I wouldn't recommend anyone learn 20 languages. You begin by learning one, two or three, and you pick them up as you progress. It's a bit like natural languages.

Q. What would you do today if you began working in a place like Bell Labs, where you created C++?

A. There isn't anywhere like that. It was a place that was unique in world history, where applied science and applied engineering were performed on a huge scale. We built the first communications satellites. We conceived how to talk with something that was out in space.

Q. Doesn't Google have something similar?

"The best approximation [regarding C++] would be to say that everyone in the world has used it and many are using it all the time. And you never see it"

A. No. There are some very interesting things but nothing as wide in scope or on that scale. We had astrophysicists, pure mathematicians. [Bell Labs](#) worked with a horizon of 10, 20 years and more. It wanted to improve worldwide communications. They defined the problem that they wanted to solve with such a wide scope that you could begin there and work for 40 years without solving the problem, even though you had made good progress.

Q. Would you be able to write C++ now?

A. It was always somewhat impossible. I had a good boss who believed that I was doing something else and, but when he saw what I was doing, thought that it might work. Instead of telling me to publish it straight away, he asked me to continue working because something good might come out of it. It was unexpected. I had a problem to solve. I used C++ as a tool. I never managed to solve the problem, but it had this collateral effect

Q. You owe quite a lot to Denmark. Though you've lived your entire life in the USA. Would things have been different if you'd been born there?

A. Probably very different. Denmark is perhaps the most fair and open society on earth. And one of the most favorable societies for people who want to do things. If you have a dream, Denmark is probably [the place to pursue it](#). They value hard work, education [is](#)

[free](#), they don't put hurdles in your way, they don't tell you what you have to do. It's more complicated today but when I was there it was a more egalitarian, simple society. I'm not sure what I would have done in the US with origins such as mine. People believe that they have done everything by themselves. No. You need a system in which to live: your education, your teachers, friends who don't stab you in the back. I don't feel I have the skills to survive in a society ruled by the law of the jungle.

Q. You are 68 and still working with Morgan Stanley bank in New York. Won't you retire?

A. I've already retired twice, from AT&T and University of Texas A&M University. To do what I do, I have to understand how software is made in real places. [For that I need colleagues and a job.] That way, I can see what people are programming instead of what it is supposed that they should be doing according to one theory or another. It is very important for my work in the design of the C++ language. In order to improve C++, it's necessary to watch people using it rather than people who write books on how it should be done.

Q. From reading your interviews, it seems that you have always done what suited you.

A. Yes, I think it's worked out well for me. If I retired, what could I do that would be this much fun?