C++ PROGRAMMING LANGUAGE

-by Owen Hughes, TechRepublic

C++'s origins date back to 1979, when Bjarne Stroustrup, the programming language's creator, first began work on the language that was then known as "C with Classes". The language was initially designed as an improvement on the C programming language that added features based on object-oriented programming.

"My focus was on the software. I needed to write low-level, close-to-hardware code, such as memory managers, process schedulers, and device drivers to separate software components, so that they could be running on separate computers communicating in well-defined ways."

1998 marked the year C++ was formally standardized and became a solid workhorse. By this point, C++ was one of the most commonly used programming languages in the world, a position it retains to this day.

Yet it was the arrival of C++11 thirteen years later that laid the foundations for the language's future evolution, says Stroustrup. "C++11 made C++ feel like a new language," he says.

"The type-safe support for concurrency was essential. C++11 supplied a dense web of mutually supporting features such as [constant expression] functions for compile-time computation, lambdas, automatic type deduction, and variadic templates."

While C++11 introduced several new features and capabilities that made the language simpler, faster and more expressive, the language maintains its reputation for being difficult to master – a fact its creator readily attests to.

"C++ is indeed complex, and it takes effort to learn to use it well," says Stroustrup. "Unfortunately, people don't just want simplicity, they what something impossible: a simpler language, with more features, and no breakage of their existing code."

Stroustrup's answer to this "trilemma," as he calls it, is adding features that "make simple things simple to do" – for example, though generalization or direct support for common use cases – while also maintaining both compatibility and stability.

That's easier said than done with a beast as large as C++, or any major programming language for that matter.

C++ IN PRACTICE

As of September 2020, C++ is the fourth most popular programming language globally behind C, Java and Python, and – according to the latest TIOBE index – is also the fastest growing. C++ is a general-purp ose programming language favored by developers for its power and flexibility, which makes it ideal for operating systems, web browsers, search engines (including Google's), games, businesses applications and more.

Stroustrup summarizes: "If you have a problem that requires efficient use of hardware and also to handle significant complexity, C++ is an obvious candidate. If you don't have both needs, either a low-level efficient language or a high-level wasteful language will do."

Yet even with its widespread popularity, Stroustrup notes that it is difficult to pinpoint exactly where C++ is used, and for what. "A first estimate for both questions is 'everywhere'," he says.

"In any large system, you typically find C++ in the lower-level and performance-critical parts. Such parts of a system are often not seen by end-users or even by developers of other parts of the system, so I sometimes refer to C++ as an invisible foundation of everything."

He also highlights the relative unreliability of developer surveys in estimating the popularity of a given programming language: "Counting programmers is hard and simple. Web surveys typically just measure 'noise'; that is, what is being talked about as opposed to what is being used."