Abstract:

C++ started as a simple good idea and was almost immediately a useful tool for seriously constrained applications. Its early success made language stability essential. Then, how can C++ be improved? The ultimate aim of the evolution of C++ is – as ever – to make code simple, elegant, and efficient. The constraints on any improvement is neither to compromise compatibility nor C++’s efficiency and ability to work directly with hardware.

I describe C++’s evolution over the last 25 years from a language supporting C-style programming, simple data abstraction, and object-oriented programming to today’s support for resource management, generic programming, and concurrent programming. I emphasize the use of resource handles and static types. I will show examples of the STL (the standard containers and algorithms library) and several features from the new ISO C++11 standard, such as move semantics, uniform initialization, and lambda expressions.

Biography:

Bjarne Stroustrup is the designer and original implementer of C++ and the author of several books (including "Programming -- Principles and Practice using C++" and "The C++ Programming Language") and many popular and academic publications. His research interests include distributed systems, design, programming techniques, software development tools, and programming languages. He is actively involved in the ISO standardization of C++. Dr. Stroustrup is a University Distinguished Professor at Texas A&M University and the holder of the College of Engineering Chair in Computer Science. He is a member of the US National Academy of Engineering, an IEEE Fellow, and an ACM Fellow. Professor Stroustrup was born in Aarhus Denmark and received a Cand. Scient. degree in Mathematics and Computer Science from the University of Aarhus, Denmark. Dr. Stroustrup was awarded a Ph.D. in Computer Science from Cambridge University, England.