



Workshop on Quality Software:
A Festschrift for Bjarne Stroustrup

Texas A&M University
College Station, Texas
April 27-28, 2012

Organizing Committee:

Nancy Amato

Jaakko Jarvi

Lawrence Rauchwerger

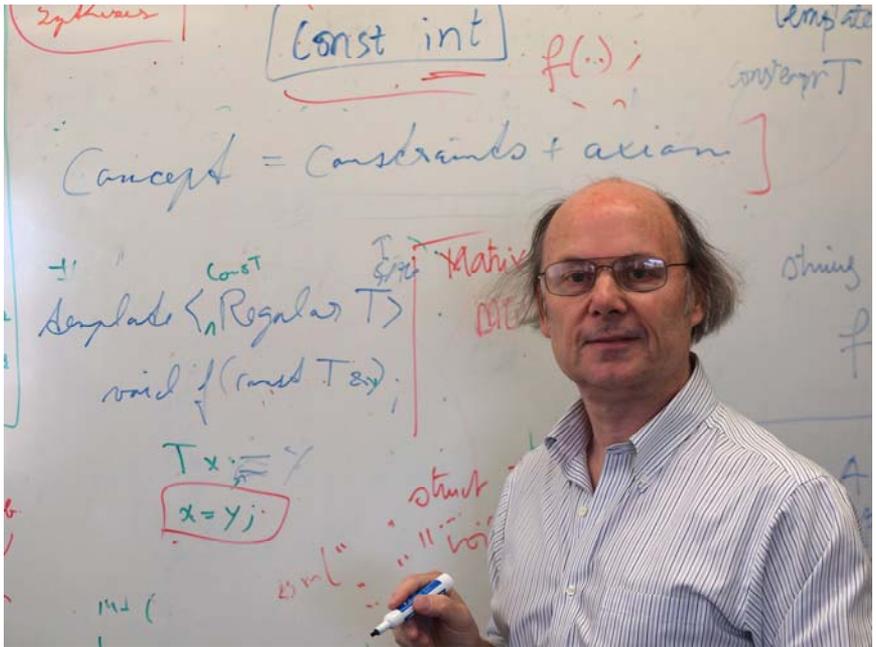
Jennifer Welch

Sponsored by:

The Parasol Lab

The Department of Computer Science
and Engineering at Texas A&M

CSE Industrial Affiliates Program



Workshop on Quality Software: A Festschrift for Bjarne Stroustrup

Texas A&M University
College Station, Texas
April 27-28, 2012

Dr. Bjarne STROUSTRUP

Dr. Bjarne Stroustrup is a Distinguished Professor and holder of the College of Engineering Chair in Computer Science in the Department of Computer Science and Engineering at Texas A&M University. From 1979 until he joined Texas A&M in 2002, he was a member of Bell Labs and, after the break-up of the Bell System, AT&T Labs-Research, to which he retains a link as an AT&T Fellow. His research interests include distributed systems, design, programming techniques, software development tools, and programming languages.

Bjarne Stroustrup is the designer and original implementer of C++. He took an active role in the creation of the ANSI/ISO standard for C++ and continues to work on the maintenance and revision of that standard. Over the last decade, C++ has become the most widely used language supporting object-oriented programming by making abstraction techniques affordable and manageable for mainstream projects. Using C++ as his tool, Stroustrup has pioneered the use of object-oriented and generic programming techniques in application areas where efficiency is a premium; examples include general systems programming, switching, simulation, graphics, user-interfaces, embedded systems, and scientific computation. For about two decades, C++ has been among the most widely used programming languages. The influence of C++ and the ideas it popularized are clearly visible far beyond the C++ community. Languages including C, C#, Java, and Fortran99 provide features pioneered for mainstream use by C++, as do systems such as COM and CORBA.

His book "The C++ Programming Language" (Addison-Wesley, first edition 1985, second edition 1991, third edition 1997, "special" edition 2000) is the most widely read book of its kind and has been translated into at least 19 languages. A later book, "The Design and Evolution of C++" (Addison-Wesley, 1994) broke new ground in the description of the way a programming language was shaped by ideas, ideals, problems, and practical constraints. His recent programming textbook *Programming -- Principles and Practice using C++*, has (so far) been translated into 7 languages. In addition to his five books, Stroustrup has published more than a hundred academic and more popular papers.

Bjarne Stroustrup was elected member of The National Academy of Engineering in 2004 "for the creation of the C++ programming language". As the first computer scientist ever, he was awarded the 2005 William Procter Prize for Scientific Achievement from Sigma Xi (the scientific research society). He was given the IEEE Computer Society's 2004 Computer Entrepreneur Award "for pioneering the development and commercialization of industrial-strength, object-oriented programming technologies, and the profound changes they fostered in business and industry." He is an AT&T Bell Laboratories Fellow and an AT&T Fellow. He received the 1993 ACM Grace Murray Hopper award "for his early work laying the foundations for the C++ programming language. based on those foundations and Dr. Stroustrup's continuing efforts, C++ has become one of the most influential programming languages in the history of computing". Member of the Texas Academy of Medicine,

Engineering, and Science. He is an ACM fellow and an IEEE fellow. He received the 2008 Dr. Dobb's Excellence in Programming award for "advancing the craft of computer programming". He served on the Danish Research Council. He was named one of "America's twelve top young scientists" by Fortune Magazine in 1990 and as one of "the 20 most influential people in the computer industry in the last 20 years" by BYTE magazine in 1995. In 1990, "The Annotated C++ Reference Manual" received Dr Dobb's "Jolt Cola" award for excellence in technical documentation. In 1995, "The Design and Evolution of C++" received a Dr Dobb's "Productivity Award" for helping programmers to improve their code.

Bjarne was born in 1950 and grew up in Aarhus, the second largest city in Denmark. He went to Aarhus University studying in the department of computer science DAIMI, gaining the Danish equivalent to a good Masters degree in 1975 (a Cand.Scient. degree is rarely taken in significantly less than six years - at least it wasn't then). Aarhus is a wonderful town of about 250,000 people beautifully sited on the East coast of Jutland.

Bjarne did his doctoral work on design of distributed systems in the Computing Laboratory of Cambridge University, England, receiving his Ph.D. in computer science in 1979. He is a member of Churchill College, where he and his wife, Marian, spent some wonderful and busy years and where their daughter, Annemarie, was born. Cambridge is another wonderful town and one of the magical places of the world. His thesis advisor in Cambridge was David Wheeler and he also spent significant time talking with (learning from) Roger Needham. He didn't really get to know Maurice Wilkes until years later. He shared an office with Bruce Croft, Jeremy Dion, Neil Grey, David Harper, Andy Hopper, and Mark Pezzaro.

In 1979, Bjarne Stroustrup, together with his wife and daughter, moved to New Jersey to join the Computer Science Research Center of Bell Telephone Laboratories (colloqually know as 127 or 1127). Over the years, they lived in Summit, Meyersville, and Watchung - all about 10 minutes drive from Bell Labs' main research site in Murray Hill. A son, Nicholas, was born in Meyersville. After the 1984 break-up of the Bell System, Bell Labs became AT&T Bell Labs, and after the 1995 break-up of AT&T, AT&T Bell Labs was itself split into AT&T Labs and Lucent Technologies Bell Labs. From its inception, Bjarne was a member of AT&T Labs-Research, the half of Bell Labs Information Sciences Research that AT&T kept to itself as Lucent and NCR were spun off. Bjarne was the head of the Large-Scale Programming Research Department from its creation in AT&T Bell Labs until late 2002 when he joined the Department of Computer Science of Texas A&M University.

Bjarne is now the College of Engineering Chair in Computer Science Professor at Texas A&M University. He is a member of the Parasol Lab in the Department of Computer Science and Engineering (the department's name changed a few years back). He retains links with AT&T Labs-Research as an AT&T Fellow.

He and his wife live in College Station, Texas. Their daughter is a medical doctor and their son is a graduate student in systems biology. His non-research interests include general history, light literature, photography, hiking and running, travel, and music.

Schedule of Events

Thursday, April 26, 2012

6:30pm **Opening Reception**, hosted by Bjarne and Marian Stroustrup at their home.

Friday, April 27, 2012

Technical sessions will be held in room 2005 of the Emerging Technologies Building (ETB) on the campus of Texas A&M University. Lunches and breaks will be held in the same location as the technical sessions. The banquet on Friday evening will be held at the Brazos Cotton Exchange in Bryan and will be catered by Madden's.

8:30am **Registration Opens**

8:50am **Opening Remarks**

9:00am-10:30am **Session 1: Software Quality—Challenges and Solutions**

Session Chair: Bjarne Stroustrup

End-to-End Software Quality, Ravi Sethi (Avaya)

Lessons from Bugs, Lawrence Crowl

Educating Programmers: A Customer Perspective,
Alex Stepanov (AV9.com)

10:30am-11:00am **Break**

11:00am-12:00pm **Session 2: Panel: Teaching Programming**

Moderator: Jan Christiaan van Winkel (Google)

Panelists: Alex Stepanov (A9.com), Bjarne Stroustrup (Texas A&M)

12:00pm-1:30pm **Lunch (included)**

1:30pm-3:00pm **Session 3: C++, in Parallel**

Session Chair: Lawrence Rauchwerger (Texas A&M)

C++11 Threads Surprises, Hans Boehm (HP Labs)

Design and Evaluation of Scalable Software, Damian Dechev (UCF)

Large-scale distributed computation at Google,
Matt Austern (Google)

3:00pm-3:30pm **Break**

Friday, April 27, 2012 *continued*

3:30pm-4:00pm **Session 4: C++, It's Everywhere**

Session Chair: Gabriel Dos Reis

The Programming Languages Beacon, Vincent Lextrait (Amadeus)

4:00pm-5:00pm **Session 5: Panel: C++ Standards**

Moderator: Alisdair Meredith

Panelists: Matt Austern (Google), Jonathan Caves (Microsoft),
Lawrence Crowl, Jan Christiaan van Winkel (Google)

7:00pm **Banquet, Brazos Cotton Exchange**

7:00pm *Reception*

7:45pm *Dinner*

Saturday, April 28, 2012

Technical sessions will be held in room 2005 of the Emerging Technologies Building (ETB) on the campus of Texas A&M University. Lunches and breaks will be held in the same location as the technical sessions.

9:00am-10:30am **Session 6: Bugs and more**

Session Chair: Jaakko Jarvi (Texas A&M)

Dynamic Bug Detection for C, C++, and UPC,
Peter Pirkelbauer (LLNL)

Domain Engineering with Concepts,
Magne Haverlaen (Univ. Bergen)

enable_if: Extending a language in four lines or less,
Andrew Lumsdaine (Indiana Univ.)

10:30am-11:00am **Break**

11:00am-12:00pm **Session 7:**

Places, People, and Code, Bjarne Stroustrup (Texas A&M)

12:00pm **Concluding Remarks**

12:00pm-1:30pm **Lunch (included)**



Sessions

SESSION 1

Software Quality—Challenges and Solutions



Session Chair:
Bjarne Stroustrup



End-to-End Software Quality
Ravi Sethi

Abstract

"Improve the State of Software and Know It" is the guiding principle for software technology research in Avaya. This talk is an overview of lessons learned over the past decade. We begin with a Customer Quality Metric, a measure of quality as perceived by the

customer, and work backward in the software life-cycle to improvements in development processes. Annual assessments of the state of software provide data to drive improvements. [Based on work with Audris Mockus, Randy Hackbarth, and John Palframan.]

Biography

Ravi Sethi is a co-author of the textbook, *Compilers: Principles, Techniques & Tools*, popularly known as the "dragon book." It is "widely regarded as the classic definitive compiler technology text" [Wikipedia]. He has a Ph.D. from Princeton University and a B. Tech. from the Indian Institute of Technology (IIT), Kanpur. Ravi Sethi has worked in academia - Penn State and Univ. of Arizona - and industry - Bell Labs and Avaya Labs. He came to Avaya when it became an independent company in 2000 to start Avaya Labs Research.



Lessons from Bugs

Lawrence Crowl

Abstract

We encounter bugs often in our daily work, but far less often do we ask what we really learned from those bugs. I describe some of the bugs I have encountered, and some I have avoided, and some general lessons on software quality that came from the experience.

Biography

Dr. Lawrence Crowl has expertise in programming languages, compilers, and parallel programming. He has been mixing threads with C++ for twenty-five years and is a major contributor to C++11.



Educating Programmers:

A Customer Perspective

Alex Stepanov

Abstract

Many programmers lack the skills required for producing quality software. Programming must be taught as a serious discipline, with an extensive core curriculum covering such topics as system decomposition and component design. Developing a sense of programming aesthetics, including the study of real world examples of beautiful programs, is at the heart of this discipline. Grounding in elementary mathematics (algebra and Euclidean geometry) provides the necessary intellectual and aesthetic foundation for the curriculum.

Biography

Alexander Stepanov (Александр Александрович Степанов) studied mathematics at Moscow State University from 1967 to 1972. He has been programming since 1972, first in the Soviet Union and then, after emigrating in 1977, in the United States. He has programmed operating systems, programming tools, compilers, and libraries. His work on the foundations of programming has been supported by General Electric, Polytechnic Institute of New York, AT&T, Hewlett-Packard, Silicon Graphics, and Adobe Systems. He is presently employed by A9.com as a software engineer. In 1995, he received the Dr. Dobb's Excellence in Programming Award for the design of the C++ Standard Template Library.

SESSION 2

Panel: Teaching Programming

Abstract

Many people programming today never had a real programming education. The programming forums are full of questions (and bad answers) of long since solved problems and trivia. Some materials even point people in the wrong direction. Point in case: a C++ book skipping treatment of the copy constructor because that subject was deemed too difficult.

This is an undesirable situation. But can the colleges and universities deliver enough graduates with sufficient CS and Math background? Are the employers willing to pay the salaries that these schooled programmers may expect? How can we educate the current people in programming jobs without a proper background? Are employers willing to pay for the education of these people? What should courseware for current non-educated programmers look like?



Jan Christiaan van Winkel

Moderator

Biography

After a "sinful" start in High school programming in Basic and a Texas Instruments SR56 programmable calculator, JC quickly came back to the right path with some Algol and a solid CS education at Hogeschool Eindhoven and at the free University of Amsterdam. He taught C++ for 20 years at AT Computing, a small courseware company in the Netherlands, and he has been the Dutch representative in the C++ standardization committee since 1997. He currently works for Google in Zurich.





Bjarne Stroustrup

Panelist

Biography

Not necessary!



Alex Stepanov

Panelist

Biography

Alexander Stepanov (Александр Александрович Степанов) studied mathematics at Moscow State University from 1967 to 1972. He has been programming since 1972, first in the Soviet Union and then, after emigrating in 1977, in the United States. He has programmed operating systems, programming tools, compilers, and libraries. His work on the foundations of programming has been supported by General Electric, Polytechnic Institute of New York, AT&T, Hewlett-Packard, Silicon Graphics, and Adobe Systems. He is presently employed by A9.com as a software engineer. In 1995, he received the Dr. Dobb's Excellence in Programming Award for the design of the C++ Standard Template Library.



SESSION 3

C++, in Parallel



Session Chair:
Lawrence Rauchwerger



*Design and Evaluation
of Scalable Software*
Damian Dechev

Abstract

Software applications need to change and adapt as modern architectures evolve. Nowadays advancement in chip design translates to increased parallelism. Exploiting such parallelism is a major challenge in modern software engineering. Multicore processors are about to introduce a significant change in the way we design and use fundamental data structures. In this talk we outline the design of a library of highly concurrent and scalable nonblocking data containers with associated programming interface and optimization support to significantly enhance the productivity and performance of multi-threaded C/C++ applications on multicore architectures. In addition, we introduce our performance modeling and simulation approach, where the efficiency and scalability of optimized code is evaluated on a reconfigurable simulator to help boost the overall application's performance and identify communication and synchronization bottlenecks.



Biography

Damian Dechev is an Assistant Professor at the EECS Department at the University of Central Florida in Orlando, FL and a joint Senior Member of Technical Staff at the Scalable Computing R&D Department at Sandia National Labs in Livermore, CA. Damian's main research interests are in the areas of non-blocking synchronization, programming tools and techniques, and program analysis. Damian earned his Ph.D. degree from Texas A&M University in 2009 under the guidance of Dr. Bjarne Stroustrup.



C++11 Threads Surprises

Hans Boehm

Abstract

C++11 adds support for threads to the languages and carefully defines the meaning of shared variables, providing a "memory model" for threads. I'll review the basics of the approach we took, and then focus on those characteristics that often surprise users. I'll try to convince you that those surprises either represent clear improvements to more traditional approaches, or at least one of several reasonable choices. For example, I'll explain why synchronization- and IO-free infinite loops are essentially treated as programmer errors, why trylock() can fail spuriously, and why the C++11 standard treats "detached" threads with caution, and you should, too.

Biography

Hans Boehm is a research manager at HP Labs. He is probably best known as the primary author of a commonly used garbage collection library. Experiences with threads in that project eventually led him to initiate the effort to properly define threads and shared variables in C++11. He is an ACM Distinguished Scientist and a former Chair of ACM SIGPLAN. He holds a B.S. degree from the University of Washington, and M.S. and Ph.D. degrees from Cornell University.



Large-scale distributed computation at Google

Matt Austern

Abstract

Distributed processing of large datasets on large clusters of computers poses challenges of scaling, fault tolerance, communication, and latency tolerance. Google has developed several frameworks, including MapReduce and Pregel, that enable efficient distributed computation while presenting an abstract and relatively simple model to the programmer. This talk presents several of those frameworks and describes how they achieve reliable computation using unreliable components.

Biography

Matt Austern works in Google's Systems Infrastructure group, where he is currently tech lead of the Pregel project and previously worked on MapReduce. Before coming to Google he worked at Apple, at SGI, and in Bjarne Stroustrup's Large-Scale Programming Research Department at AT&T. He also participates in C++ standardization, and is the author of a book about the STL.



The Programming Languages Beacon

Vincent Lextrait

Abstract

Vincent will present the Programming Languages Beacon, a page he maintains on the Internet which describes the impact of C++ on familiar applications. He will explain where it comes from, what is the process to maintain it, he will show analytics on the consultations of the page, and will try to give a few pieces of explanation on why, beyond its intrinsic qualities, C++ has succeeded to take this overwhelming share of the software that surrounds us.

Biography

Vincent Lextrait is Director of Software Development at Amadeus, the leading Travel Technology supplier in the world. After being responsible for the move to open systems, designing and delivering the transactional infrastructure (up to 170,000 transactions a second, one billion and a half transactions a day) and delivering the first Airline IT solutions to British Airways and Qantas, he is currently responsible of Hotel, Rail, Car, Cruise and Insurance reservations, with solutions engineered in 6 main locations in the world. Vincent has spent his entire career in the software product industry, and in parallel to that has been for years the head of C++ standardization in France. He organized in 1997 the C++ ISO meeting in Sophia Antipolis, France, which led to the first C++ standard. He has been member of ACM and IEEE for more than a decade, and holds an engineering degree from École des Mines de Paris, and a PhD in Computer Science from University of Nice.

SESSION 5

Panel: C++ Standards

Abstract

The first C++ standard was ratified in 1993, preceded by years of work by the standards committee. Since then, new revisions of the standard have been published in 2003 and in 2011; the latter a major milestone that introduced a large number of new language features. The standardization process is quite heavy-weight and completely based on voluntary effort of different organizations (compiler and library vendors, software companies, programming language researchers, individual hackers, etc.) and their representatives.

This panel will discuss the role of the standards committee in C++'s well-being and its future; the impact of the largely "consensus-seeking" mode of operation of the committee on a language's evolution; and many other topics about the workings and impact of this varied community of software professionals.



Alisdair Meredith

Moderator

Biography

Alisdair Meredith has been an active member of the C++ standards committee for 9 years, and is the current chair of the Library Working Group. By a fortunate coincidence, his first meeting was also the first meeting to consider proposals for the planned C++0x standard! In that time, he has worked in fields as diverse as Formula 1 motor racing, compiler tools, and now financial news, all made possible by C++!



Matt Austern

Panelist

Biography

Matt Austern works in Google's Systems Infrastructure group, where he is currently tech lead of the Pregel project and previously worked on MapReduce. Before coming to Google he worked at Apple, at SGI, and in Bjarne Stroustrup's Large-Scale Programming Research Department at AT&T. He also participates in C++ standardization, and is the author of a book about the STL.



Jonathan Caves

Panelist

Biography

Jonathan first encountered C++ in 1989 at Fermi National Laboratory and immediately knew this was the language for him. Very soon afterwards he found a job working on the C-front based C++ compiler at Glockenspiel in Dublin, Ireland. In 1993 he joined Microsoft where, except for one brief foray into the realm of C# and domain specific languages, he has been working on the Visual C++ compiler.



Lawrence Crowl

Panelist

Biography

Dr. Lawrence Crowl has expertise in programming languages, compilers, and parallel programming. He has been mixing threads with C++ for twenty-five years and is a major contributor to C++11.



Jan Christiaan van Winkel

Panelist

Biography

After a "sinful" start in High school programming in Basic and a Texas Instruments SR56 programmable calculator, JC quickly came back to the right path with some Algol and a solid CS education at Hogeschool Eindhoven and at the free University of Amsterdam. He taught C++ for 20 years at AT Computing, a small courseware company in the Netherlands, and he has been the Dutch representative in the C++ standardization committee since 1997. He currently works for Google in Zurich.

SESSION 6

Bugs and more



Session Chair:
Jaakko Jarvi



Dynamic Bug Detection for C, C++, and UPC
Peter Pirkelbauer

Abstract

At compile-time, main-stream programming languages can detect software flaws in source code to a varying degree. While finding all software bugs statically is a noble goal, in practice absolute safety is hard to attain for compilers of general purpose languages. I will present a dynamic error detection tool CIRM (code instrumentation and runtime monitor) for C, C++, and UPC source codes. Built on top of the ROSE source-to-source transformation infrastructure, CIRM instruments source files with code that monitors operations and tracks changes of the system state. CIRM can detect uninitialized variables, arithmetic overflow/underflow, invocation of C standard library functions with precondition violating arguments, memory errors (involving stack, heap, and UPC's global address space), and similar software defects that would be otherwise hard to spot. During program execution, the runtime monitor keeps track of changes to the overall system state and flags operations that could compromise a system's safety. I will present ROSE-CIRM's implementation for both sequential and parallel codes and results obtained by checking against an error detection benchmark test suite. We also measured the runtime performance overhead by running error-free test codes. To overcome the slowdown incurred by monitoring the program execution, I will discuss how relatively simple static analysis techniques can eliminate unnecessary safety checks.

Biography

Peter Pirkelbauer graduated with a PhD from Texas A&M University in 2010, where he was a member of Dr. Stroustrup's research group. Currently, he is a post-doctoral researcher in Dr. Quinlan's compiler group at the Lawrence Livermore National Laboratory. At LLNL, he is working on dynamic error detection and on source code rejuvenation of large scale parallel systems using MPI. His research interests include programming languages, source code analysis, transformation systems, and non-blocking programming techniques.





Domain Engineering with Concepts Magne Haveraaen

Abstract

Domain engineering is discovering the terminology that describes a domain and organising it into highly reusable artefacts - enabling the development of software product lines. Capturing the domain entails declaring the involved types and functions and their relationships - C++ "concepts". Here we show how we developed coordinate-free numerics doing "domain engineering with concepts" on the domain of partial differential equations (PDEs). Coordinate-free numerics has proven to be a stable library design for the PDE domain within scientific computing, naturally aligning the variation points of high performance PDE solvers with the library structure.

Biography

Magne Haveraaen is a full professor at the Department of Informatics, University of Bergen (UoB), where he is heading the Bergen Language Design Laboratory (BLDL). Haveraaen's research interests are in domain engineering, language design and formal methods, both the theoretical and the pragmatical aspects. He was instrumental in the development of coordinate-free numerics, and responsible for its C++ library implementation.



enable_if: *Extending a language in four lines or less* Andrew Lumsdaine

Abstract

In many ways, C++ can be considered an open programming language platform rather than just a programming language. A number of its features enable programmers to extend the basic language in remarkable ways. Since these extensions can be made within the language, the resulting features can be used and evaluated in the context of the complete C++ ecosystem. This openness, flexibility, and extensibility all contribute to the richness of C++ and to its wide applicability across numerous domains. In this talk I will discuss `enable_if`: four lines of C++ that provide a concept-controlled polymorphism mechanism for C++.

Biography

Andrew Lumsdaine is a Professor of Computer Science in the School of Informatics & Computing at Indiana University and Director of the Center for Research in Extreme Scale Technologies (CREST) at the Pervasive Technology Institute. Lumsdaine received his Ph.D. from MIT in 1992. His research interests include computational science and engineering, parallel and distributed computing, software engineering, generic programming, mathematical software, numerical analysis, and computational photography. Lumsdaine is a member of ACM, IEEE, and SIAM, the MPI Forum, and the ISO C++ standards committee.

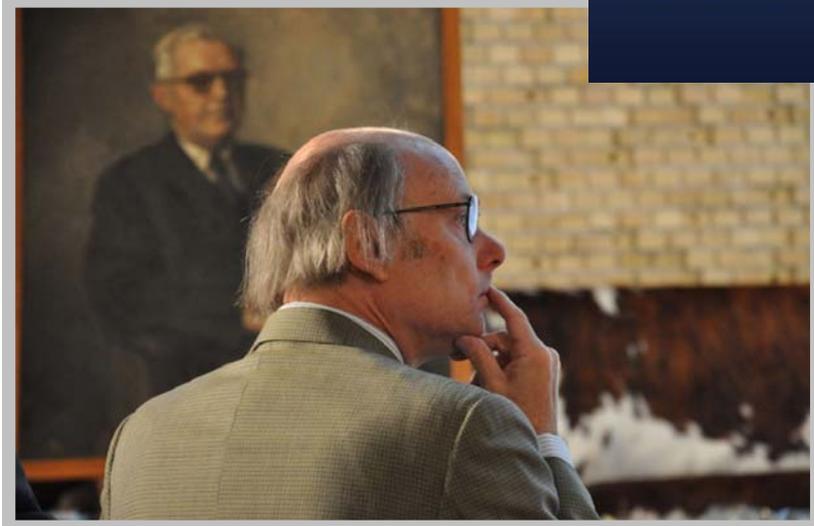
SESSION 7

Places, People, and Code

Bjarne Stroustrup

Abstract

A few reminiscence about my work with an emphasis on people.





Bell Laboratories

Computing Science Technical Report No. 84

Classes: An Abstract Data Type Facility for the C Language

Bjarne Stroustrup



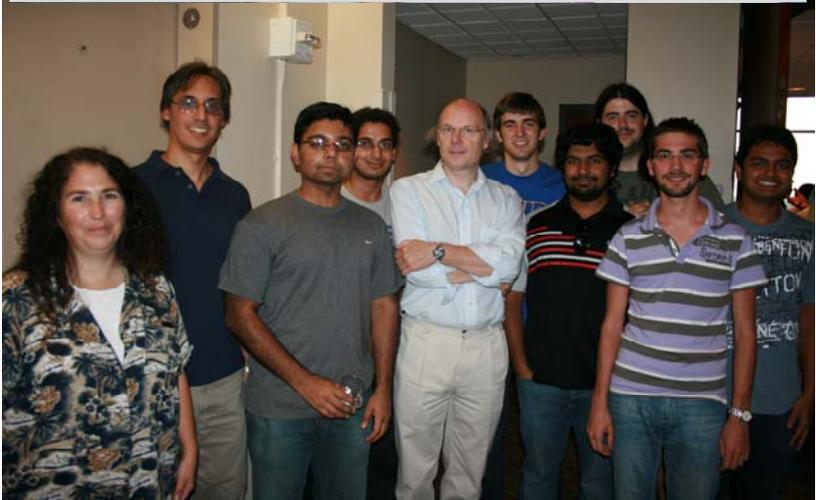
Bell Laboratories

Computing Science Technical Report No. 90

A Set of C Classes for Co-routine Style Programming

Bjarne Stroustrup









I am so happy that you could come for this "Festschrift" and hope that we'll all have a great time, professionally and socially. I'm a bit overwhelmed by this rare honor. Thanks for coming and thanks to all the people from the TAMU CSE department who made it possible. Enjoy!

