What I do

Bjarne Stroustrup
COE professor in Computer Science, TAMU
AT&T Fellow
http://www.research.att.com/~bs
What I do

• Standardizing C++
  • Designing C++0x

• Research
  • Into programming techniques, tools, and languages

• Teaching
  • Beginners (freshmen) are the most fun and the most challenging

• About 20 minutes introduction
  – then questions and answers
What do I find interesting?

• Software: building it – figuring out how to build it – tools for building it

• Neat applications!
  Mars rovers, animation, graphics, Photoshop, GUI, OS, SDE, compiler, slides, chip design, chip manufacturing, semiconductor tools, etc.

• Neat people
  Who build such applications, often in neat places
Ships

- Design
- Construction
- Management

- Monitoring
- Engine
- Hull
- Pumps
Aircraft

- Communication
- Control
- Display
- Signal processing
- Gadget control
- Monitoring
Phones

- Voice quality
- User interfaces
- Billing
- Mobility

- Switching
- Reliability
- Provisioning
- Images
Power

- Control
- Monitoring
- Analysis
- Design

- Communications
- Visualization
- Manufacturing
PC/workstation

- There’s a lot more to computing than games, word processing, browsing, and spreadsheets!
Biomedical
ISO Standard C++

• C++ is a general-purpose programming language with a bias towards systems programming that
  – is a better C
  – supports data abstraction
  – supports object-oriented programming
  – supports generic programming

• A multi-paradigm programming language
  (if you must use long words)
  – The most effective styles use a combination of techniques
C++ ISO Standardization

• Current status
  – Library TR 2005, Performance TR 2005
  – C++0x in the works – ‘x’ is scheduled to be ‘9’
  – Documents on committee website (look for WG21 on the web)

• Membership
  – About 22 nations (8 to 12 represented at each meeting)
    • ANSI hosts the technical meetings
    • Other nations have further technical meetings
  – 160+ active members (60+ at each meeting)
    • 200+ members in all

• Process
  – formal, slow, bureaucratic, and democratic
  – “the worst way, except for all the rest” (apologies to W. Churchill)
  – only protection from corporate lock-in
The (real) problems

• Help people to write better programs
  – Easier to write
  – Easier to maintain
  – Easier to achieve acceptable resource usage
Overall Goals

• Make C++ a better language for systems programming and library building
  – Rather than providing specialized facilities for a particular sub-community (e.g. numeric computation or Windows-style application development)

• Make C++ easier to teach and learn
  – Through increased uniformity, stronger guarantees, and facilities supportive of novices (there will always be more novices than experts)
template<class T> using Vec= vector<T,My_alloc<T>>;

Vec<double> v = { 2.3, 1, 6.7, 4.5 };

sort(v);

for (auto p = v.begin(); p!=v.end(); ++p) cout<< *p << endl;

for (auto x : v) cout<< x << endl;
A possible solution: SELL languages

- Superset: Add libraries to provide application-specific facilities
- Subset: Subtract features (outside the library implementation) to provide semantic guarantees
- The result is a subset of a superset of a language
SELL language advantages

• Semantics that closely matches application concepts
  – Restrictions, constraints
  – Optimizations
• Composable extensions
  – Given care
• Familiar syntax
• Existing infrastructure
  – Compilers, debuggers, libraries, education, etc.
• Relatively cheap to produce
  – Write library
  – Implement constraints in general-purpose tool
• Relatively easy to produce
  – Skills needed closely related to those of good application designers
• Notation that closely matches application concepts
  – The expressiveness of a general-purpose can be strained
The Pivot

C++ source

Compiler

Object code

Tool 1

IDL

Tool 2

C++ source

Tool 3

“information”
Hard-real-time C++

• Add
  – Fixed-sized arrays (potentially range checked)
  – Constant-time allocators
    • Stacks
    • Pools
  – Templated device drivers and other hardware-access facilities
    • See WG21 performance TR

• Subtract
  – Arrays
  – new (except during startup)
  – delete
  – Exceptions (with suitable tools we can ‘reclaim’ those)
  – Unions (maybe)

• Transforms
  – None, but generate data for help with timing estimates
But!

C++ is shockingly misused

• “If that’s C++, then I don’t like it either!”
  – Bjarne Stroustrup, 2004
    • (after surveying a couple of dozen C++ textbooks)
Teaching

• What should a student know when he/she starts their first real project?
  – A real project is a project aimed at providing a service for someone else
  – Someone might get hurt or ruined if the program fails

• A student can’t stay in school playing with safe toys forever
  – Good students don’t want to
Geek heroes

• Brian Kernighan
  – Programmer and writer extraordinaire

• Dennis Ritchie
  – Designer and original implementer of C
Another geek hero

- Kristen Nygaard
  - Co-inventor (with Ole-Johan Dahl) of Simula67 and of Object-oriented Programming
Yet another geek hero

• Alex Stepanov
  – Inventor of the STL and generic programming pioneer
First modern program – first compiler

- David Wheeler
  - University of Cambridge
  - Exceptional problem solver: hardware, software, algorithms
  - First computer science Ph.D. (1951)
  - (Thesis advisor for Bjarne Stroustrup 😊)
Places: AT&T Bell Labs Murray Hill
Places: Cambridge University