CS 2:
How did we get here?
Where should we be going?

Kim B. Bruce

Williams College
& Princeton University

CS 2 Workshop
OOPSLA '98

What a long strange trip it's been!

My introduction:
Data Structures from Knuth:
MIXAL & FORTRAN

Taught Data Structures in:

- FORTRAN (boxes of cards)
- Pascal (batch jobs on mainframe)
- Modula-2 (personal computers)
- ML & Object Pascal
  C++ alternative available
- Java

Other possibilities I haven't explored: Ada, LISP/Scheme, C++, Eiffel, . . .

Programming & data structures

- How to save bits (packed words)
- Code everything by hand from pseudo-code
- Code provided in some detail in text,
  students type and modify
- Code provided in on-line libraries
  associated with text
- Data structure implementations supplied
  with language

But what were we teaching?

- Abstract Data Types
- Implementation of data structures
- Specification vs. implementation
- Advanced programming
  writing large programs
- Analyze complexity
- Evaluate trade-offs in selection of
  data structures
- Other topics: correctness, files, etc.

What new is worth considering?

- Event-driven programming (GUI)
- Concurrency
- Object-oriented design
- Patterns

Building *systems* should be new focus.

May end up with less emphasis on algorithms.

What can we drop?

- array implementation of queues?
- pointer implementations of stacks?
- Detailed analysis of hash tables?
- Advanced graph algorithms?
- Balanced binary trees?

Some can be reclaimed in algorithms.

Won't give up some things

- Analysis of complexity
- Some original implementation by students
- Proofs of correctness
- Examining tradeoffs.

What is the solution?

How many of the new topics can occur in CS1?

- Event-driven programming
  with provided GUI components

- Simple concurrency

- Simple patterns

See Lynn Andrea Stein's "Rethinking CS101"
at `http://www.ai.mit.edu/projects/cs101/`

CS2 cannot be changed in isolation.

Changes affect CS1 and more advanced courses.

We do live in interesting times!