All I Really Need to Know I Learned in CS1

Elliot Koffman
Computer & Information Sciences Dept.
Temple University, Philadelphia, PA
koffman@temple.edu

Thank you SIGCSE

"For those of us who are primarily educators, the SIGCSE award is our Turing award"

Dan McCracken

1992 SIGCSE award winner

for outstanding contributions to Computer Science Education

Outline of Talk

- Acknowledgments
- SIGCSE first language survey
- My experiences as a CS1 student
- Early Years teaching CS1
- Some SIGCSE Experiences
- Later Years teaching CS1
- Future Directions for CS1

All I Really Need to Know I Learned in CS1

40TH ANNIVERSARY

All I really need to know I learned in Kindergarten CS1

Kindergarten	*CS1*
First word in Dick and Jane: *LOOK*	Think before you code
Share everything	Write reusable code
Clean up your own mess	Test and debug your code and remove errors
Play well with others -- Say thank you and please	Solve problems, resolve conflicts, pick good partner

Good Doubles Partners

Not Good Partners

Deans Play Tennis

My Great Partner Caryn 1963

My Writing Partners

Frank Friedman
Temple University
Fortran, BASIC, C++

Bruce Maxim
U. Of Michigan Dearborn
Turbo Pascal

Michael Feldman
George Washington University
Ada, Ada 95

My Writing Partners (cont'd)

Jeri Hanly
University of Wyoming
C

Ursula Wolz
The College of New Jersey
Java

Paul Wolfgang
Temple University
CS2 in Java, C++

CS Graduates – 20XX?

Languages selected most Often by SIGCSE (362 responses)

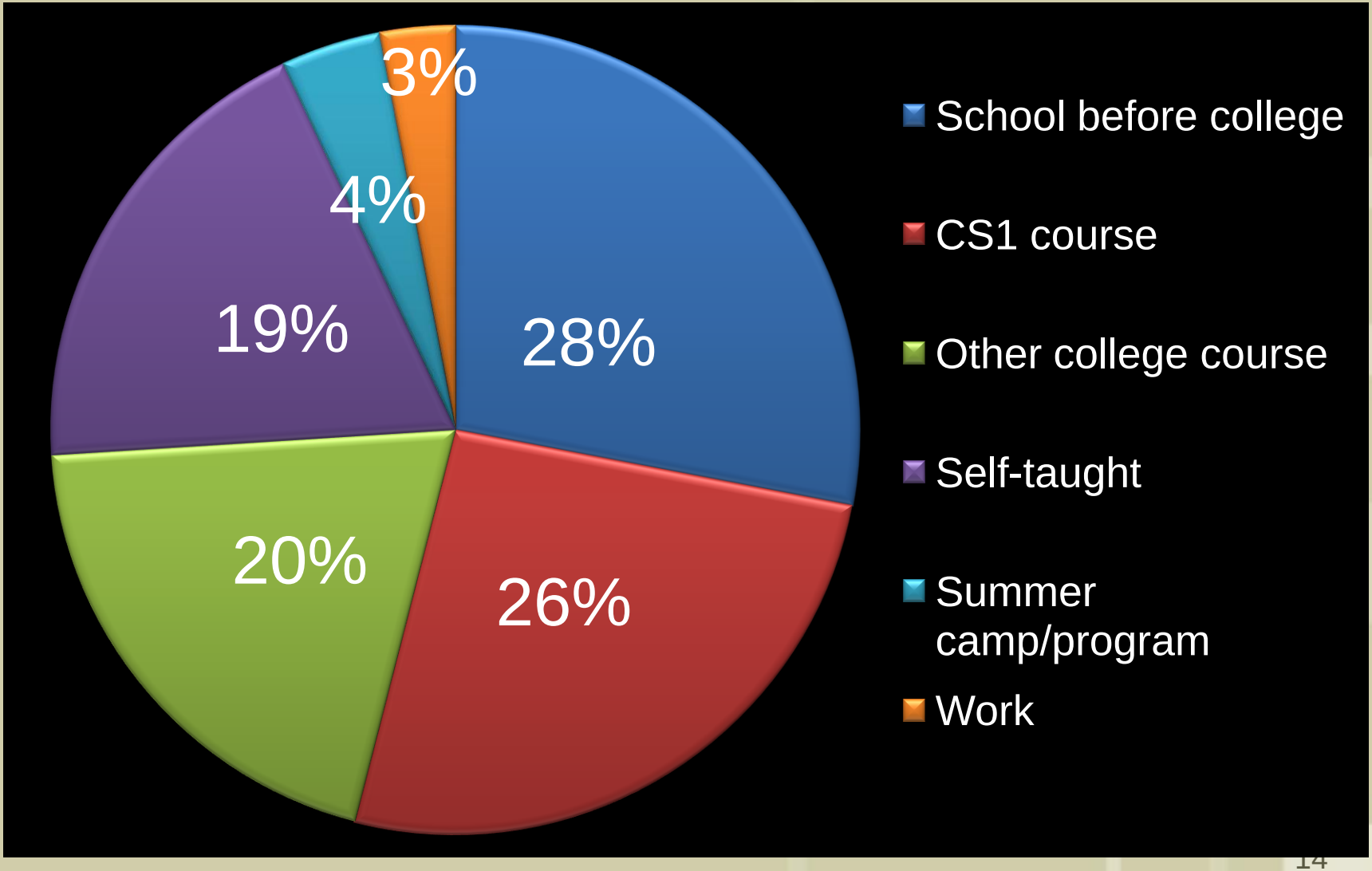
1st Language	Count	Years
Machine	14	1950s, 1960s, 1970s
Assembler	19	1950s, 1960s, 1970s, 1980s
Algol	6	1960s, 1970s
BASIC	129	1950s, 1960s, 1970s, 1980s
Fortran	129	1950s, 1960s, 1970s, 1980s
MAD	2	1960s
PL/I	13	1960s, 1970s, 1980s
Pascal	31	1970s, 1980s, 1990s
C	1	1990s
C++	5	1990s

Other Languages with 1 or 2 Selections (362 responses)

Jovial	Java
Scheme	COBOL
Turing	LISP
APL	Oberon
Logo	Modula-2

Increase with next generation of CS Faculty

Learned to Program At?



My CS1

Fall 1959
False Start

- **1-credit experimental freshman Elective**
- **Taught by John McCarthy, MIT, Stanford**
- **Lead to ARMY ROTC**

Fall 1963
Try Again

- **Digital Logic Design**
- **Loved it!**
- **EE 6.251**
  MAD, Assembly, Fortran with Dave Liu, U of Illinois

EE 6.251: Introduction to System Programming

(still offered)

MAD

Four programming exercises

CAP (Classroom Assembly Program)

Simple Assembler for IBM 7094

Fortran

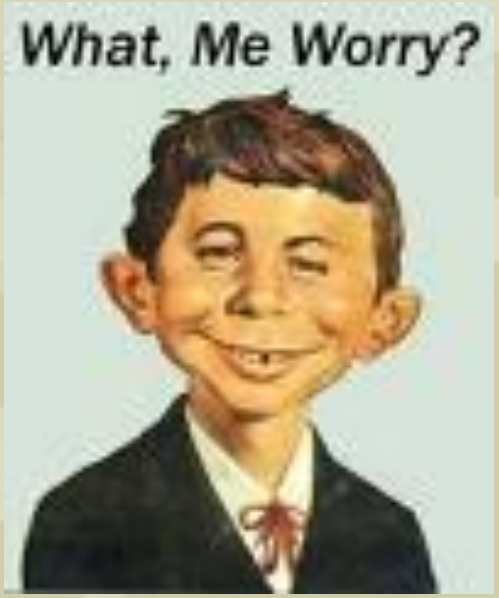
One or two exercises

Four runs per assignment

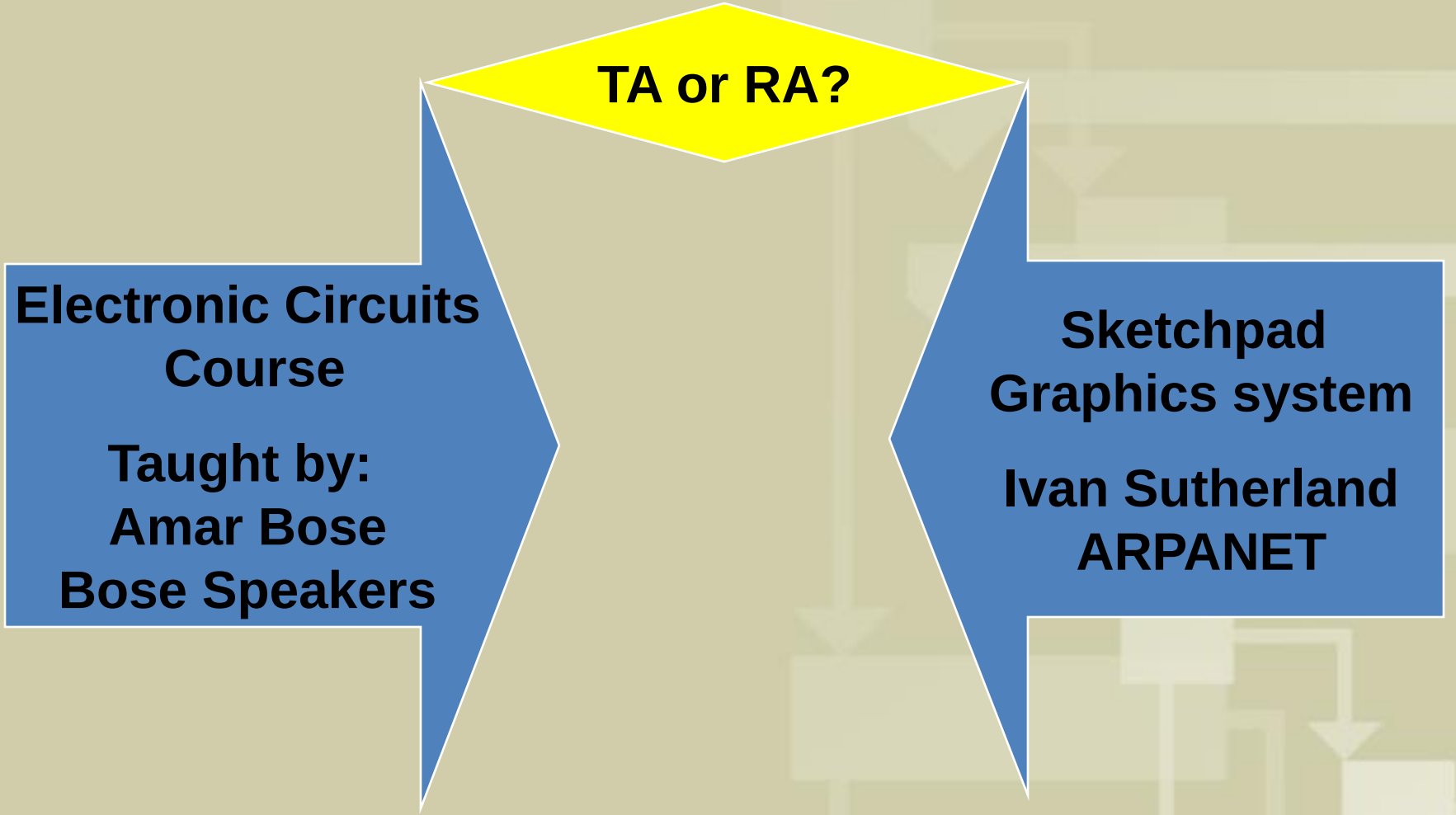
MAD

(Michigan Algorithmic Decoder)

- Developed at University of Michigan in 1959
- Based on Algol 58
- Written for IBM 704 with 8K of core by Graham, Arden, and Galler. Also, versions for IBM 7094 and Univac 1107
- Primer written by Elliot Organick -- 1985 SIGCSE award winner
- Used at Michigan, MIT, Yale, Maryland, Texas, …

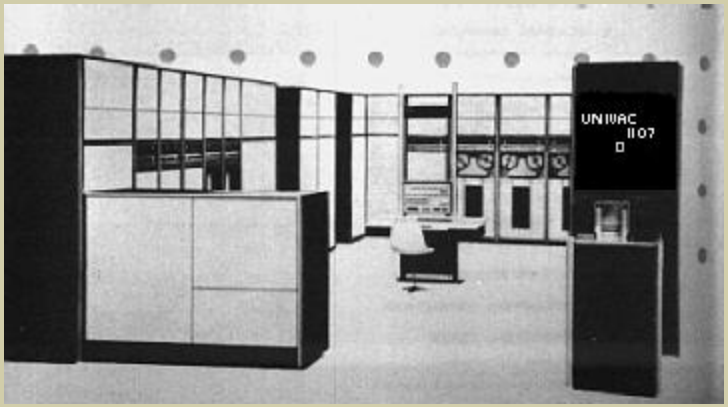


Teaching vs. Research Conflict



Algol at Case Tech

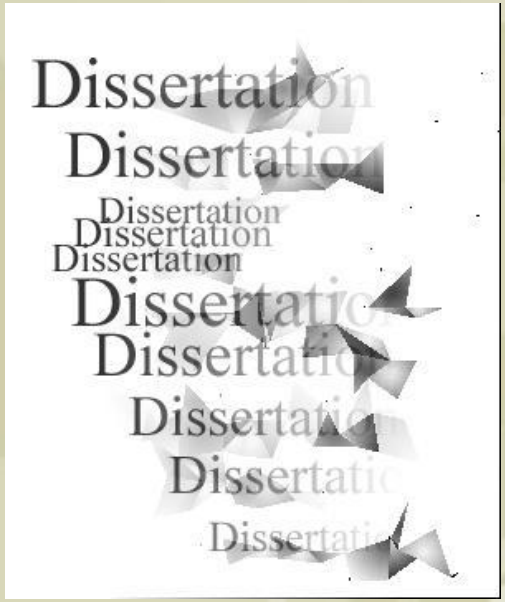
(1964-1967)



Univac 1107

Case Algol Compiler in 1964

Learning Games through Pattern Recognition



My Languages for teaching CS1

University of Connecticut – Taylor Booth

1969-1974:	assembler

Temple University

1974 -1977:	Fortran and assembler
1977-1985:	Structured Fortran
1985-1993:	Pascal
1993-1995:	C (CS0)
1993-1999:	C++
2000-2009:	Java

Structured Fortran Textbook

Used Friedman preprocessor for structured control statements

- **Emphasized problem solving not just Fortran syntax**
- **Adapted software engineering "waterfall model": specification, analysis, design, implementation, and testing**
- **Design phase used structured flowcharts**

Structured Fortran

- Language features found in WATFIV and Minnesota Fortran
  - format-free input/output
  - CHARACTER data type
  - Block IF statement
  - FOR loop (general indexed DO)
  - PARAMETER statement
  - WHILE loop
- All in Fortran 77 except WHILE loop

CS1 – 80's to early 90's Pascal

- Based on Algol W developed by Wirth and Hoare
- Simple language with small grammar - Strongly typed
- Procedures had value and variable parameters
- Pointer type supported indirection
- Few language wars - most Computer Science Departments were on same page – life was good

Pascal Concerns…

Controversy

- Introduce procedures?
  - Procedures early
  - Procedures late
  - The "procedures early" approach in CS 1: a heresy. Pattis, SIGCSE 1993

Disadvantage

- Limited I/O
- Industry Use

Revising the Curricula

ACM Task Force on CS1/CS2

Recommended Curriculum for CS 1 Communications of the ACM 1984	Revised Curriculum for CS2 Communications of the ACM 1985
Koffman	Koffman
Caroline Wardle (BU)	Caroline Wardle (BU)
Philip Miller (CMU)	David Stemple (UMass)

Precursor to Curriculum ’90

Successor to Pascal - Modula-2

Late 1980s - Wirth's Modula-2

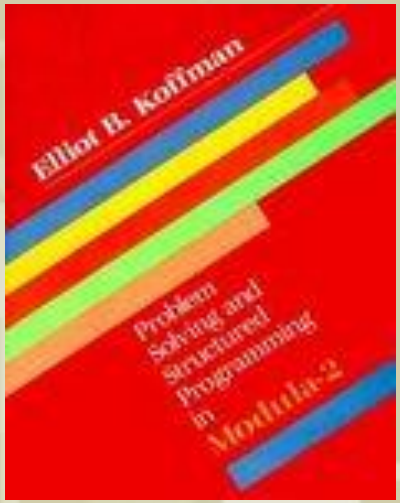
Unsuccessful successor to Pascal

- **Introduced MODULE for true data encapsulation**
- **Widely hyped but not widely used.**

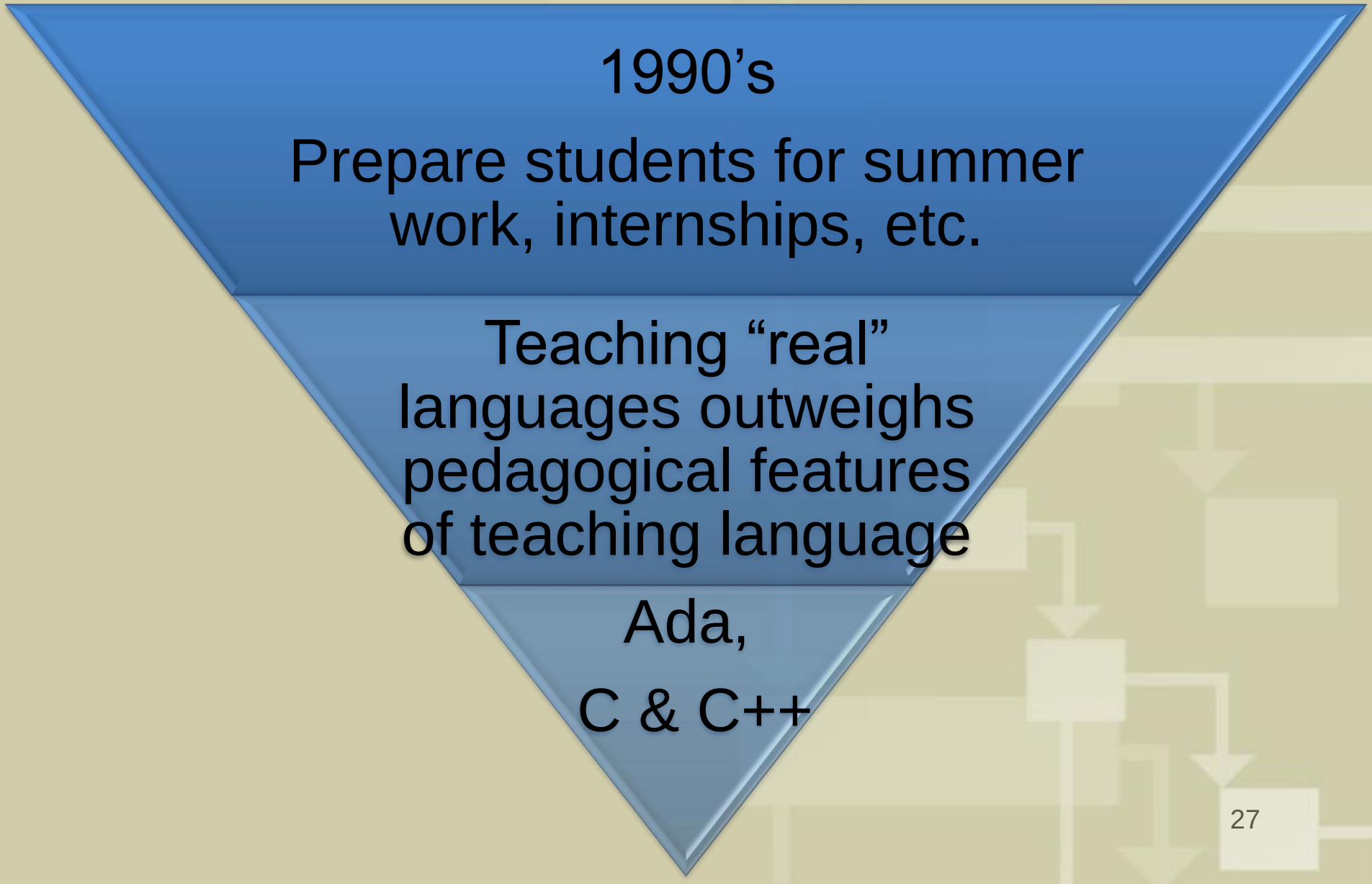
Modula-2 publications

The Case for Modula-2 in CS1 and CS2, 19th Technical Symposium on Computer Science Education, Atlanta, GA, 1988

Problem Solving and Structured Programming in Modula-2, 1988



Industrial Strength Languages



Ada

- Ada influenced by Pascal and DOD
- for embedded and real-time systems
- strong typing, packages, run-time checking, parallel processing (tasks), exception handling, and generics
- Ada 95 added objects and OOP

Ada Disadvantages

Compilers

- Relatively slow

Language

- Verbose & difficult to learn

Academics

- Not accepted

C

- systems implementation language for Unix in 1972
- Small but powerful grammar
- Manipulates data at both high and low levels
- Widely available on a variety of platforms
- Good for later systems courses

C Disadvantages

- No type checking or range checking
- only value parameters
- pointers needed for reference parameters
- Combining operators leads to ambiguous code `while (*s++ = *t++);`
- No boolean data type – uses type int
- Lack of common language features impedes transition to other languages

C++

- Extension of C
- Classes for object oriented programming
- reference parameters
- More user-friendly i/o than C
- Exceptions for detecting errors
- Stronger data typing than C
- Better libraries including STL -- good for data structures course

C++ Disadvantages

- Complex language that is difficult to learn
- Syntax issues are distracting
- No range checking
- Not true strong typing
- Not a true object oriented language
- Strong opposition from CS faculty to replacing Pascal with C++ for AP Exam in 1995

New Paradigm for CS1 with Java

- Replaced C++ for AP exam in 2004
- True object oriented programming language
- Large class library (API)
- Web programming and Applets
- Supports graphics (AWT and Swing)
- Strong type checking
- Arrays have length attribute and range checking
- Collections framework for data structures course

Java Disadvantages

- Large language, complex syntax, extensive API
- Input of numbers is a bit tricky
- Mix of primitive types and objects is confusing
- Students must learn references right away
- Lack of pointer type limits students' ability to understand indirection or addressing in later courses
- Difficult to transition from Java to C++ because students must do explicit housekeeping in C++
- **Resolved: Objects early has failed (Koffman/Reges vs. Bruce/Kölling) during SIGCSE 2005**

CS1 languages and CS Enrollment

Bill Manaris (Inroads, 2007 December)

- Use of C++ and Java in CS1 increased to 76% in 1999-2000
- 50% drop in CS enrollments from 2000-2006
- Is there evidence that OO in CS1 improves student performance or retention?
- In our students' minds, the language / paradigm we expose them to in CS1 is CS
- Does CS seem too hard for beginners?

What's Next for CS1 ???

Alice

- 3D programming environment
- Easy to create an animation
- Teaching tool for introductory computing

Python

- Dynamic object-oriented programming language
- Strong support for integration
- Extensive standard libraries
- Learned in a few days

Matlab

- High-performance technical computing environment
- Provides comprehensive math and graphics functions
- Provides support for OOP

Scratch

- Visual programming language
- Teaching tool for introductory computing

What's next for CS1 ???

Game programming

Robot Programming

Web and multimedia programming

Thank you

Questions

Additional Information

`whenever` conditional statement

The following is an example of a function whose value is $1/x$ if $0 < x \leq 1$ and $1/x^2$ if $x > 1$. If $x \leq 0$, one obtains an error return (see Section 2.9).

```
A   EXTERNAL FUNCTION (X)
J   ENTRY TO INVSF.
G   WHENEVER X.G.0. .AND. X .LE. 1.
C      FUNCTION RETURN X .P. -1
H   OR WHENEVER X .G. 1.
D      FUNCTION RETURN X .P. -2
I   OTHERWISE
E      ERROR RETURN
K   END OF CONDITIONAL

B   END OF FUNCTION
```

The labels above are for discussion purposes only.

Structured Flowchart