

PROGRAMMING LANGUAGES IN CS CURRICULA

KIM BRUCE
POMONA COLLEGE

OUTLINE

- ACM / IEEE CS curricula
- Liberal Arts CS curricula

IN THE BEGINNING ...

- Curriculum 1968
 - 8 required courses + 2 constrained electives + math + 3 free electives
 - Intro (4): programming I & II, discrete structures, and numerical calculus
 - Intermediate (4): data structures, programming languages, computer organization, systems programming
 - Constrained electives: compiler construction, switching theory, sequential machines, numerical analysis I & II.
 - Math (6): 2 semesters calculus, probability, & linear algebra, + 2 restricted electives

CURRICULUM 1968

- Programming Languages:
 - Course I2. Programming Languages
Formal definition of programming languages including specification of syntax and semantics. Simple statements including precedence, infix, prefix, and postfix notation. Global properties of algorithmic languages including scope of declarations, storage allocation, grouping of statements, binding time of constituents, subroutines, coroutines, and tasks. List processing, string manipulation, data description, and simulation languages. Run-time representation of program and data structures.
Prerequisites: Programming, Discrete Structures.

PL TOPICS

- Structure of simple statements (*includes parsing & precedence*) (10)
- Structure of algorithmic languages (*Traditional PL topics*) (20)
- List Processing and string manipulation languages (7)
- Topics in PLs: Simulation, algebraic manipulation, parallel languages, formal semantics (*Floyd-Hoare*) (8)

CURRICULUM 1978

C'78 WIDELY REVILED

☉ Requirements:

- ☉ CS 1. Computer Programming I
- ☉ CS 2. Computer Programming II
- ☉ CS 3. Introduction to Computer Systems
- ☉ CS 4. Introduction to Computer Organization
- ☉ CS 5. Introduction to File Processing
- ☉ CS 6. Operating Systems and Computer Architecture I
- ☉ CS 7. Data Structures and Algorithm Analysis
- ☉ CS 8. Organization of Programming Languages

☉ Math requirements, but independent of CS

CURRICULUM 1978

☉ Programming Languages:

- ☉ CS 8. Organization of Programming Languages (3-0-3)
The objectives of this course are:
(a) to develop an understanding of the organization of programming languages, especially the run-time behavior of programs;
(b) to introduce the formal study of programming language specification and analysis;
(c) to continue the development of problem solution and programming skills introduced in the elementary level material.
Prerequisite: CS 2 (CS 3 and CS 5 highly recommended)

PL TOPICS

☉ Topics

- ☉ Language Definition Structure. (15%)
- ☉ Data Types and Structures. (10%)
- ☉ Control Structures and Data Flow. (15%)
- ☉ Run-time Consideration. (25%)
- ☉ Interpretative Languages. (20%)
- ☉ Lexical Analysis and Parsing. (10%)
- ☉ Examinations. (5%)

ACM/IEEE CS CURRICULA 1991

KNOWLEDGE UNITS

- ☉ AL & DS (47) - *includes computability*
- ☉ AR (59)
- ☉ AI (9)
- ☉ DB (9)
- ☉ HU (8)
- ☉ NU (7)
- ☉ OS (31)
- ☉ PL (46) - *includes grammars & automata*
- ☉ SE (44) - *includes intro programming*
- ☉ SP (11) - *Social, ethical, & professional issues*

PL TOPICS (46 HOURS)

- ☉ PL1: History and Overview of Programming Languages (2)
- ☉ PL2: Virtual Machines (2)
- ☉ PL3: Representation of Data Types (2)
- ☉ PL4: Sequence Control (4)
- ☉ PL5: Data Control, Sharing, and Type Checking (4)
- ☉ PL6: Run-time Storage Management (4)
- ☉ PL7: Finite State Automata and Regular Expressions (6)
- ☉ PL8: Context-Free Grammars and Pushdown Automata (4)
- ☉ PL9: Language Translation Systems (3)
- ☉ PL10: Programming Language Semantics (2)
- ☉ PL11: Programming Paradigms (10)
- ☉ PL12: Distributed and Parallel Programming Constructs (3)

ACM/IEEE CS CURRICULA 2001

2001 CORE TOPIC AREAS

- Discrete Structures (43)
- Programming Fundamentals (38)
- Algorithms & Complexity (31) - *inc. 6 hrs computability*
- Architecture & Organization (36)
- Operating Systems (18)
- Net-Centric Computing (15) *networking, security, e³ client-server*
- Programming Languages (21)
- HCI (8)
- Graphics & Visual Computing (3)
- Intelligent Systems (10)
- Information Management (10) *database*
- Social & Professional Issues (16)
- Software Engineering (31)
- Computational Science (0)

PL TOPICS (21 HOURS)

- PL1. Overview of programming languages (2)
- PL2. Virtual machines (1)
- PL3. Introduction to language translation (2)
- PL4. Declarations and types (3)
- PL5. Abstraction mechanisms (3)
- PL6. Object-oriented programming (10)
- *PL7. Functional programming*
- *PL8. Language translation systems*
- *PL9. Type systems*
- *PL10. Programming language semantics*
- *PL11. Programming language design*

Optional!

SLEIGHT OF HAND

<i>Objects-first intro, compressed approach</i>	CS 111: OO Programming	CS112: OO Design	CS 210: Algorithm Analysis	CS 226: OS & Networking
PL1: Overview of PLs		2		
PL2: Virtual Machines		1		
PL3: Intro to Language Translation			2	
PL4: Declarations & Types	2	1		
PL5: Abstraction Mechanisms	1	2		
PL6: Object-Oriented Programming	8	4		2

*Other sample curricula similar:
PL topics came from Programming Fundamentals e³ relabeled!*

LIBERAL ARTS CS CURRICULA

LACS

- Founded in 1984 w/ Sloan Foundation grant
 - Create degree programs in CS appropriate for liberal arts colleges and to help support faculty.
- Originally faculty from 9 liberal arts colleges
 - Currently 20
- Three sets of degree proposals: 1986, 1996, & 2007.
 - Published in CACM (twice) & JERIC
- Partially in reaction to ACM curriculum 1978

LACS MODEL CURRICULA

⊗ LACS 1986

- ⊗ Intro (2): Std intro, data structures, & discrete math
 - ⊗ 2nd math elective from calculus, discrete math 2, linear algebra
- ⊗ Core (4): Organization, Algorithms, Theory, PL's
 - ⊗ PL includes functional, OO, logic programming
- ⊗ Electives (3)

REVISED LACS MODEL CURRICULUM

⊗ LACS 1996

- ⊗ Recommend formal labs in first 2 courses
 - ⊗ Cover two of procedural, OO, or functional paradigms
- ⊗ Increased coverage of concurrency & distributed systems
- ⊗ Add OS topics
- ⊗ Better integrate theory throughout core
- ⊗ Increase math requirements
- ⊗ PL remains core course
- ⊗ Senior project

CO 3: PL & SYSTEMS

- ⊗ Language paradigms:
 - ⊗ procedural, functional, OO, logic
- ⊗ Language design & implementation
 - ⊗ traditional PL topics
- ⊗ Language issues related to parallelism
 - ⊗ includes processes, process management, message passing & shared memory, etc.
- ⊗ Proving properties of programs:
 - ⊗ formal semantics & program proofs

2007 LACS MODEL CURRICULUM

⊗ LACS 2007

- ⊗ Software development as 3rd intro
- ⊗ Theory material redistributed to Theoretical Foundations and PL courses
- ⊗ Functional programming early:
 - ⊗ Objects-first approach integrates functional programming w/ discrete math course.
 - ⊗ Functional-first approach as well
- ⊗ PL remains core course
- ⊗ FC1: Discrete Math, FC2: Theoretical Foundations of CS
 - ⊗ includes fsa, computability, prob & stats, number theory, & proofs

PROGRAMMING LANGUAGES (39 HRS)

- ⊗ Overview (3)
- ⊗ Formal grammars & parsing (6)
- ⊗ Types (4)
- ⊗ Semantics (2)
- ⊗ Language translation (3)
- ⊗ Control of execution (3)
- ⊗ Declarations & modules (3)
- ⊗ OO languages (3)
- ⊗ Run-time storage management (3)
- ⊗ Declarative programming paradigms (3)
- ⊗ Distributed & parallel programming paradigms (6)