ACM vs USI
- The Core of CS
- Curricula
- Introductory Courses
- Intermediate Courses
- Advanced Courses
The Core of CS Curricula
CS Body of Knowledge

- Area > Unit > Topic
- Core vs elective units
- Units => courses
- Courses != lectures
The Core of CS

- Set of **units** known by **all** CS students
- Core < Curriculum
- Core != introductory
Courses

- Introductory
- Intermediate
- Advanced
Sample Course

Core: DS 1, DS 2, PF 2, AR 8, PF 3
Elective: OS 6, OS 7
### Non-Sample Courses

- It’s our job!
- Take an area
- Assign units to USI courses

---

#### Figure 5-1. Computer science body of knowledge with core topics underlined

**CC2001 Computer Science**

- **Programming Fundamentals (38 core hours)**
  - PL11. Programming language design
  - PL10. Programming language semantics
  - PL9. Type systems
  - PL8. Language translation systems
  - PL2. Virtual machines
  - PL1. Overview of programming languages

- **Operating Systems (18 core hours)**
  - OS12. Scripting
  - OS10. Fault tolerance
  - OS9. Real-time and embedded systems
  - OS8. File systems
  - OS7. Security and protection
  - OS4. Scheduling and dispatch

- **Architecture and Organization (36 core hours)**
  - AR6. Functional organization
  - AR5. Interfacing and communication
  - AR3. Assembly level machine organization
  - AR2. Machine level representation of data

- **Discrete Structures (43 core hours)**
  - DS6. Discrete probability
  - DS5. Graphs and trees
  - DS4. Basics of counting
  - DS2. Basic logic
  - DS1. Functions, relations, and sets

- **Algorithms and Complexity (31 core hours)**
  - AL11. Parallel algorithms
  - AL10. Geometric algorithms
  - AL9. Cryptographic algorithms
  - AL8. Advanced algorithmic analysis
  - AL7. Automata theory

- **.Net-Centric Computing (15 core hours)**
  - NC1. Introduction to net-centric computing
  - NC2. Communication and networking
  - NC3. Network security
  - NC4. High-performance computing

- **Human-Computer Interaction (8 core hours)**
  - HC1. Usability engineering
  - HC2. Human-computer interaction
  - HC3. Computer-mediated communication

- **Graphics and Visual Computing (3 core hours)**
  - GV1. Fundamental techniques in graphics

- **Intelligent Systems (10 core hours)**
  - IS1. Knowledge representation and reasoning
  - IS2. Advanced search
  - IS3. Natural language processing

- **Information Management (10 core hours)**
  - IM1. Information models and systems
  - IM2. Database systems
  - IM3. Data management

- **Software Engineering (31 core hours)**
  - SE1. Software design
  - SE2. Using APIs

- **Computational Science (no core hours)**
  - CN1. Numerical analysis
  - CN2. Operations research
  - CN3. Modeling and simulation
  - CN4. High-performance computing

---

**Note:** The numbers in parentheses represent the minimum number of hours required to cover this material in a lecture format. It is always appropriate to include more.
<table>
<thead>
<tr>
<th>Team</th>
<th>Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Marco &amp; Romain</td>
<td>PF, GV, IM</td>
</tr>
<tr>
<td>Jochen &amp;</td>
<td>AR,</td>
</tr>
<tr>
<td>Jeff &amp; Leonardo</td>
<td>NC, IS</td>
</tr>
<tr>
<td>Cyrus &amp; Paolo</td>
<td>AL, SE</td>
</tr>
<tr>
<td>Matthias</td>
<td>OS, PL</td>
</tr>
<tr>
<td>Chiara &amp; Lasaro</td>
<td>DS, CN, HC</td>
</tr>
<tr>
<td></td>
<td>SP,</td>
</tr>
</tbody>
</table>
All Teams ... Report!

(use whiteboard)
Do we cover the core?
We’re done for today!
The Core of CS

Curricula

Introductory Courses
ACM Curriculum...

- 4 years
- Majors & non-majors

- 1st year:
  - 1 course / semester

- 2nd year:
  - specialization
... != Usi Curriculum

- 3 years
- 1st year specialization
- Intermediate courses on 1st year
- Atelier
- Projects

Università della Svizzera italiana
Curricula Variations

<table>
<thead>
<tr>
<th>Introductory courses</th>
<th>Imperative first</th>
<th>Objects first</th>
<th>Functional first</th>
<th>Breadth first</th>
<th>Algorithms first</th>
<th>Hardware first</th>
</tr>
</thead>
<tbody>
<tr>
<td>Intermediate courses</td>
<td>Topic-based approach</td>
<td>Compressed approach</td>
<td>Systems-based approach</td>
<td>Web-based approach</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced courses</td>
<td>Additional courses used to complete the undergraduate program</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
6 approaches

• OO critical
• discrete mathematics
  ➡ parallel to CS courses
• programming-first?
  ➡ 3 with, 3 without
• 2 or 3 semesters?
  ➡ alternatives
Programming-first?

+ 
- 

- Prerequisite
- Motivating
- Skill for:
  - major & !major
  - employers
  - other faculty

- Restrain view of CS
- Syntactic details
- Oversimplification
- !test, !design ...
- Student inequalities
Imperative-first

Object-oriented paradigm

Data structures and algorithms

Introduction to programming

Data abstraction

Programming fundamentals

- imperative is widespread
  + less OO
Objects-first

Introduction to OOP → Object and data abstraction → Algorithms and data structures

Or:

OOP → OO design and methodology

+ early OO - complex languages
Introduction to functional programming

Objects and algorithms

+ less inequalities
+ less syntax
+ recursion, other concepts

- not mainstream:
  ➔ student skepticism
- too abstract?
Breadth-first

Introduction to programming + Discrete structures

Traditional courses (imperative-first?)

+ larger view of CS
+ more theory

- 1 extra course ➡ but 3 semester
Algorithms-first

Introduction to algorithms and applications

Programming methodology

+ pseudocode
→ no syntax
→ faster
+ more “science”

- less motivation
→ needs lab
- no debugging skills
Hardware-first

Introduction to the computer ➔ Object-oriented programming techniques

+ bottom up
  works well with some

- abstract concepts later
  focus on software
  computer engineering
Meanwhile, at USI ...

PF1: functional

PF2: OO

Computer Architecture

Networking

Mathematics

Discrete Structures
Meanwhile, at USI ...

**PF1:** functional

**PF2:** OO

**Computer Architecture**

**Networking**

**Mathematics**

**Discrete Structures**

**USI** = functional 1st + hardware 1st + theory + networking (intermediate) + atelier
4 approaches
Advanced Courses?