

CS-101

**Advanced information, computation, communication I**

Urbanke Rüdiger

Cursus	Sem.	Type
Communication systems	BA1	Obl.
Computer science	BA1	Obl.

Language of teaching	English
Coefficient	7
Session	Winter
Semester	Fall
Exam	Written
Workload	210h
Weeks	14
<b>Hours</b>	<b>6 weekly</b>
Courses	4 weekly
Exercises	2 weekly
<b>Number of positions</b>	

**Remark**

This course focuses on the foundational, discrete mathematics core of advanced computation.

**Summary**

Discrete mathematics is a discipline with applications to almost all areas of study. It provides a set of indispensable tools to computer science in particular. This course reviews (familiar) topics as diverse as mathematical reasoning, combinatorics, discrete structures & algorithmic thinking.

**Content**

- I. Mathematical reasoning: propositional logic, propositional functions, quantifiers, rules of inference; this includes very basic logic circuits.
- II. Sets and counting: cardinalities, inclusion/exclusion principle, sequences and summations.
- III. Algorithms and complexity: basic algorithms, computational complexity, big-O notation and variants, countability.
- IV. Number representations such as binary and hexadecimal and (postponed to 2nd semester) basic number theory: modular arithmetic, integer division, prime numbers, hash functions, pseudorandom number generation; applications.
- V. Induction and recursion: mathematical induction, recursive definitions and algorithms.
- VI. Basic combinatorial analysis: permutations, binomial theorem, counting using recursions.
- VII. Basic probability: events, independence, random variables, Bayes' theorem.
- VIII. Structure of sets: relations, equivalence relations, power set.

**Keywords**

Propositional logic, counting, complexity, big-O, number representations, sets, functions, relations, induction, basic probabilities, Bayes theorem, combinatorial analysis, recurrences, countability.

**Learning Outcomes**

By the end of the course, the student must be able to:

- Recognize if there is a mistake in a (simple) proof
- Apply general problem-solving techniques
- Recognize the mathematical structures present in applications
- Apply simple recursion and use it to design recursive algorithms
- Apply the tools studied in class to solve problems
- Demonstrate familiarity with mathematical reasoning
- Solve linear recurrences and use generating functions
- Argue about (un)countability
- Formulate complete, clear mathematical proofs

### Transversal skills

- Assess one's own level of skill acquisition, and plan their on-going learning goals.
- Continue to work through difficulties or initial failure to find optimal solutions.
- Demonstrate the capacity for critical thinking

### Teaching methods

Ex cathedra lectures

### Expected student activities

Studying the book, test your understanding by making the exercises, ask questions

### Assessment methods

Continuous evaluations 10% and final exam 90%

### Supervision

Office hours	No
Assistants	Yes
Forum	No
Others	Additional Q&A sessions will take place on Tuesdays from 17:15-18:30 in INM 200 (starting in the second week of the semester)

### Resources

#### Bibliography

"Discrete Mathematics and Its Applications", Kenneth H. Rosen, 8th ed, McGraw-Hill 2019. (You should be able to find the pdf on the web.)

#### Ressources en bibliothèque

- [Find the references at the Library](#)

#### Websites

- [http://will be provided later, if any](#)

#### Moodle Link

- <https://go.epfl.ch/CS-101>