Introduction to the Theory of Computation

Department: Fudan International Summer Session 2026

Course Code	GEIS20001.01						
Course Title	Introduction to the Theory of Computation						
Credit	2	**Experiment (including Computer) Credit**	0	**Practice Credit**	0	**Aesthetic Education Credit**	0
Credit Hours Per Week	9 credit hours per week; 36+3 tutorial hours in total (one credit hour is 45 minutes)	**Education on The Hard-Working Spirit Credit Hours**	0	**Language of Instruction**	English	**Honors Course**	□Yes ☑No
Course Type	□Core General Education Course □Specific General Education Course □Basic Course in General Discipline ☑Others			2+X Major ： □Professional Core Course □Professional Advanced Course Non 2+X Major ： □Professional Compulsory Course □Professional Elective Course			
Course Objectives	（Including value, knowledge and ability objectives） Computation is everywhere. Oftentimes, people overlook the theory behind it. This course provides a systematic and mathematically rigorous introduction to the Theory of Computation. The Theory of Computation studies the fundamental capabilities and limitations of algorithms by classifying computational problems according to their inherent difficulty, measured in terms of time and space resources. It also investigates formal relationships between problems, including reducibility, completeness, and computational equivalence. Students will develop a precise understanding of computation through formal models such as finite automata and Turing machines. The course further explores computability,						

	undecidability, and complexity theory, including the theory of NP-completeness. By the end of the course, students will understand both what can be computed in principle and what can be computed efficiently. Topics include finite automata and regular languages, Turing machines and its variants, undecidability, reductions, NP-completeness, and complexity classes such as P and NP. This course is offered exclusively as part of the Fudan International Summer Session program.
Course Description	By the end of this course, students will be able to rigorously address the following foundational questions: 1). How can computation be defined through formal mathematical models? 2). Which functions are computable, and which problems are undecidable? 3). Which problems admit efficient algorithms, and which are inherently intractable?

Course Requirements:
You need a background in linear algebra, calculus, and some basic programming experience.

Teaching Methods:
The instructor will give in-class lectures, give self-contained proofs for the key results, and handwrite all the board notes. The instructor will ask questions and provide in-class exercises, to ensure students have understood the material.

Course Director's Academic Background:
Yuan Li is an assistant professor at Fudan University. Prior to this, he was a software engineer at Google from 2017 to 2022. He received his BSc in Computer Science from Fudan University in 2011 and his PhD in Computer Science from the University of Chicago in 2017. His research interests include computational complexity and coding theory. He has published papers in conferences and journals such as IEEE Transactions on Information Theory, FOCS, SIAM Journal on Computing, Information and Computation, and DCC (Designs, Codes and Cryptography). A list of publications can be found on his Google Scholar profile: https://scholar.google.com/citations?user=j5q7RPgAAAAJ&hl=en

Instructor's Academic Background:
The instructor is the same as the Course Director. Please refer to the above section for detailed academic background.

Members of Teaching Team

Name	Gender	Professional Title	Department	Responsibility
Yuan Li	Male	Assistant Professor	Department of Computer Science	Course design, instruction, and assessment

Course Schedule (Please supply the details about each lesson)**:**

Lecture 1. Big-O notation, alphabet and languages

Lecture 2. Finite automaton

Lecture 3. Regular language

Lecture 4. Turing machine and its variants

Lecture 5. Universal Turing machine, computability

Lecture 6. Turing halting problem, undecidable problems

Lecture 7. Reductions, Church－Turing thesis

Lecture 8. Complexity class P, NP

Lecture 9. Cook-Levin theorem

Lecture 10. Turing reductions and Cook reductions

Lecture 11. More NP-complete problems

Lecture 12. Final exam

The design of class discussion or exercise, practice, experience and so on:

Each lecture will integrate guided problem-solving sessions. Emphasis will be placed on rigorous argumentation and proof techniques. Students will implement and experiment with finite automata and Turing machines using a simulator to connect formal definitions with executable computational processes. Continuous interaction is encouraged through in-class discussion and online communication channels. The instructor and teaching assistant will provide academic support and clarification as needed.

These activities are designed to make abstract theoretical concepts accessible and intellectually engaging for students from diverse academic backgrounds.

If a teaching assistant is assigned, the responsibilities will include:

Grading homework assignments and the final examination.

Responding to student inquiries via email or through the course WeChat group.

Grading & Evaluation (Provide a final grade that reflects the formative evaluation process)**:**

Homework: 5 assignments, each worth 6 points

Exams: 50 points. We will have a final exam in the course. The final exam will be held in class.

In-class participation: 10 points

Attendance: 10 points

Total: 100 points

Usage of Textbook: ☑Yes(complete textbook information form below) ☐No

Textbook Information (No more than two textbooks) **:**

Title	Author	ISBN	Publis	Publishe	Type I	Type II

			hing Time	r		
Introduction to the Theory of Computation	Michael Sipser	9781133187790	June 27, 2012	Cengage Learning	□Self-compiled Textbook (Published) ☑Non-mainland Textbook □Other Textbook (Published)	□National Planning Textbook □Provincial and Ministerial Planning Textbook □School Level Planning Textbook □Others
Teaching References (Including author, title, publisher, publishing time,ISBN)**:** Sanjeev Arora, Boaz Barak, Computational Complexity: A Modern Approach, Cambridge University Press, April 20, 2009, 0521424267.						

Table column size can be adjusted according to the content.