Arizona State University

CSE 355 SLN 86927 – **Introduction to Theoretical Computer Science** – Fall 2024

Session C: August 22nd – December 6th, 2024

Instructor:	Joshua J. Daymude (`jdaymude@asu.edu`)
TAs:	Siva Kumar Katta (`skatta14@asu.edu`)
	Nidhi Thakkar (`nthakka9@asu.edu`)
	Tanay Jaiman (`tjaiman@asu.edu`)
Office Hours:	W 11:30am–12:30pm AZ, Online (Instructor)
Recitation:	TBD AZ, Online (TA Thakkar)

Course Description

In this course, students will develop their computational thinking skills and learn about fundamental models of computation. These models will guide students' intuitions about whether computational problems are "easy" or "hard," and students may be surprised to learn that many problems are impossible for any computer to solve. This intuition is essential for framing the rest of computer science, including hardware and software design, artificial intelligence, and cybersecurity.

In more detail, this course covers formal *languages*, *grammars*, and *algorithms* along with *computability* and *decidability* theory. These concepts are abstract and may seem distant, but help us answer the central question of computer science: "What are the fundamental capabilities and limitations of computers?"

Prerequisites. MAT 243 (Discrete Mathematical Structures) and CSE 310 (Data Structures and Algorithms), or approved equivalent courses from another university if you transferred. To enroll in this course, you must be an undergraduate student majoring in Computer Science (B.S.) or Computer Systems Engineering (B.S.E.) via ASU Online.

Course Outcomes. By the end of this course, you will be able to:

- Articulate the central question of computer science (What are the fundamental capabilities and limitations of computers?) and how the "map of computability" (the hierarchy of regular, context-free, Turing-decidable, and Turing-recognizable languages) addresses that question.
- Employ tools for computational thinking, especially defining, analyzing, and manipulating formal models and abstractions.
- Ground the theoretical perspective on computer science in practical examples and systems thinking.

Textbook and Other Resources

The textbook for this course is:

- *Introduction to the Theory of Computation* by Michael Sipser, 3rd Edition. Errata available online.

This textbook is recommended as an additional resource, but is not necessary for any assessments in this course. The lecture slides are self-contained and serve as a primary resource for students.

Students should ensure they have the necessary technology for an online course, including a laptop/desktop computer with a webcam and microphone, stable internet connectivity, an internet browser (Firefox, Chrome, or Safari), a PDF reader (e.g., Adobe Acrobat Reader), and Zoom.

Course Schedule and Learning Objectives

The following dates and topics are flexible and subject to change based on course pacing, exams, and holidays.

#	Dates	Topics & Objectives
0	8/22 – 8/30	Introduction. Read the syllabus, meet the instructor, and orient to Canvas.
1	8/22 – 8/30	Discrete Math Review. Review mathematical structures and proof techniques.
2	8/26 – 9/1	Finite Automata. Formally define *finite automata* (DFAs), the computations they perform, and the *regular languages* they accept. Formulate your own finite automata. Define the *regular operations* and apply them to example languages.
3	9/2 – 9/8	Nondeterminism. Formally define *nondeterministic finite automata* (NFAs) and reason about their branching logical executions. Prove that DFAs and NFAs are equivalent. Revisit the closure properties for regular languages.
4	9/9 – 9/15	Regular Expressions. Define *regular expressions*, prove they are equivalent to finite automata, and describe their practical uses.
5	9/16 – 9/22	Nonregular Languages. Connect the central question of computer science to the notions of machines recognizing or being unable to recognize classes of languages. Apply the *pumping lemma* to prove nonregularity.
–	9/25 – 9/27	**Exam 1**: Modules 1–5.
6	9/23 – 9/29	Context-Free Grammars. Formally define *context-free grammars* (CFGs), formulate your own for specific languages, convert arbitrary CFGs to Chomsky Normal Form, and identify their real-world applications.
7	9/30 – 10/6	Pushdown Automata. Formally define *pushdown automata* (PDAs), formulate your own for specific languages, and prove that they are equivalent to CFGs.
8	10/7 – 10/13	Non-Context-Free Languages. Define the *pumping lemma for context-free languages* and apply it to prove non-context-freedom.
–	10/16 – 10/18	**Exam 2**: Modules 6–8.
9	10/21 – 10/27	Turing Machines. Formally define *Turing Machines* (TMs) and the associated *Turing-recognizable* and *Turing-decidable* languages. Formulate low-level TM specifications for target languages.
10	10/28 – 11/3	Variants of Turing Machines. Define multi-tape TMs, nondeterministic TMs, and enumerators and prove that they are all equivalent to deterministic TMs.
11	11/4 – 11/10	Algorithms and the Church-Turing Thesis. Explain the *Church–Turing thesis* and its connection to the central question of computer science.
12	11/4 – 11/10	Decidable Languages. Define at least three Turing-decidable languages (e.g., A_{DFA}, A_{NFA}, E_{DFA}, EQ_{DFA}, etc.) and sketch their proofs of decidability.
13	11/11 – 11/17	Undecidability. Define A_{TM} and use *diagonalization* to prove it is not Turing-decidable. Diagram and explain the complete map of computability.
–	11/20 – 11/22	**Exam 3**: Modules 9–13.
14	11/18 – 11/24	Undecidability via Reductions. Explain the concept of a *reduction* and sketch the proofs by reduction that $HALT_{\mathsf{TM}}$, E_{TM}, $REGULAR_{\mathsf{TM}}$, and EQ_{TM} are undecidable. Formally define *linear bounded automata* (LBAs) and compare their computational power to TMs.
15	11/25 – 12/6	Mapping Reducibility. Formalize *mapping reductions*. Prove that some languages are not Turing-recognizable.
–	12/10 – 12/12	**Final Exam**: Comprehensive, Modules 1–15.

Other important dates to be aware of (see the ASU Academic Calendar) are:

- **August 28, 2024.** Last day to add/drop a class without college approval.
- **September 4, 2024.** Last day to drop a class without receiving a W grade.
- **October 12–15, 2024.** Fall break; no new material will be covered this week.
- **November 6, 2024.** Course withdrawal deadline.
- **November 28–29, 2024.** Thanksgiving break; Module 15 will have extended deadlines to accommodate this holiday.
- **December 9–14, 2024.** Finals week; see the ASU Final Exam Schedule for details.

Evaluation

We will use standard A–E, +/− letter grades for this course as defined by the ASU Grading Policy. The standard ASU letter grade cutoffs are listed below, though the instructor reserves the right to shift these cutoffs lower (e.g., ≥85% could count as an A− instead of ≥90%) depending on the difficulty of assignments and exams in this particular offering of the course.

A+	**A**	**A−**	**B+**	**B**	**B−**	**C+**	**C**	**D**	**E**
$\geq 97\%$	$\geq 94\%$	$\geq 90\%$	$\geq 87\%$	$\geq 84\%$	$\geq 80\%$	$\geq 77\%$	$\geq 70\%$	$\geq 60\%$	$< 60\%$

Students will be evaluated according to the following rubric:

Lectures & Participation	15%
Problem Sets	20%
Machine Design	10%
Formal Conversions	10%
Concept Explanations	10%
Quizzes	10%
Exams	25%

Lectures & Participation. Each module has a sequence of lectures for you to watch before diving into the learning activities and assessments. In total, you'll watch about an hour and a half of lectures each week. All lecture videos are hosted in PlayPosit, an interactive video player and notetaking tool. Completion of lecture videos and the associated interactives are autograded by PlayPosit. This category also includes miscellaneous Canvas discussion board posts, surveys, and interactives that will be graded for completeness. Specific rubrics will be provided per assignment.

Problem Sets. Roughly half of our modules will have associated problem sets. Students will work on these problem sets in *groups of four*. During the week, students will work together on all problems in the set, collaborating on a single set of solutions to be submitted as a group. Submissions and grading will use Gradescope to ensure clear, consistent applications of grading rubrics across the class. Students are encouraged to seek help from the instructor and TA after carefully reviewing the relevant material.

All problems in a problem set will be graded for *completeness* (20%): did the group make a best-effort on all problems? The instructor will then choose one problem to grade for *correctness* (40%), but you won't know which one ahead of time, so you'll need to make a best effort on all problems. Finally, after submission, you'll perform *group member evaluations* (40%) to assess your and your group members' contributions.

Machine Design. Throughout this course, we will be tackling problems of the form "Formulate a machine M of type X that recognizes/decides language L." Although these machines are purely abstract, simulating them in a programming language is a good way to master how they work. Thus, students will be provided a Python programming interface for defining various types machines (DFAs, NFAs, and PDAs). Once a machine is defined, the programming interface can perform machine verification (is the machine definition valid?), machine visualization (what does the machine look like?), and input string evaluation and tracing (how does the machine behave on specific input?).

In each machine design assignment, students will *individually* design machines for a series of target languages. They will then upload their definitions to an automated grading environment (Gradescope) which will compute grades based on correctness.

Formal Conversions. Throughout this course, we will also see problems of the form "Convert a computational model M of type X into an equivalent model M' of type Y." To practice these conversions, students will first create their own small example model of type X and submit it to a peer-review tool called Peerceptiv. After the due date for this first submission, Peerceptiv will randomly assign each student another student's submission to grade for quality and then convert into an equivalent model of type Y. The second student will then submit their conversion work to another Peerceptiv assignment, which will be randomly assigned to a third student for peer-review. This ensures each student sees three examples (the one they create, the one they solve, and the one they grade), establishing familiarity with the material.

Concept Explanations. This course is theoretical, and thus one of the most important skills students will learn is explaining abstract concepts. Throughout the course, students will be provided prompts (usually one-sentence questions) that they will respond to *individually* in short videos. Some prompts may allso involve making a simple visual aid. These videos will be submitted to Peerceptiv and peer-graded based on rubrics provided by the instructor.

Quizzes. Most modules will have a Canvas quiz that students will take *individually.* All quizzes are *closed book*; no "cheat sheets" or course materials are allowed.

Exams. We will have three unit exams and one comprehensive final exam. All exams will be completed *individually* on Canvas using the Honorlock proctoring service. These are *closed book*; no "cheat sheets" or course materials are allowed during testing. Exam viewing sessions will be held after each exam for students to review their mistakes and understand their exam grades better, if desired.

Late Work. *No late work will be accepted*; all deliverables must be submitted on Canvas by the posted due dates to receive credit. Note that all due dates follow Arizona Standard Time (UTC-7) and Arizona does not observe Daylight Savings Time. Students are encouraged to proactively communicate any circumstances that may hinder timely completion of work, if possible, and additional late days or other special exceptions may be granted at the instructor's discretion according to ACD 304–02: Missed Classes Due to University-Sanctioned Activities, ACD 304–04: Accommodation for Religious Practices, ACD 304–11: Missed Class Due to Military Line-of-Duty Activities, and SSM 201–18: Accommodating Active Duty Military.

Communication & Office Hours

There are four venues of communication between students and instructors: email, office hours, Canvas announcements, and a dedicated Slack workspace. The instructor and TA will communicate with the class via announcements on Canvas and will respond to emails and Slack messages daily during business hours.

Regardless of the form of communication, the ASU Student Code of Conduct requires students to maintain a cordial atmosphere and use tact in expressing differences of opinion. This is especially critical in an online environment where visual cues, such as a smile or nod, are not available to provide additional context to the message. Please keep questions and comments relevant to the course material and communicate respectfully, both with the instructional team and and your peers.

Email. Emails to the instructor must originate from your `@asu.edu` email address, have a subject line beginning with `CSE 355:`, and communicate in a *professional, clear, and concise* manner. All emails should CC the TA(s) unless there is a specific reason why they should not be CC'ed. Emails should not ask questions whose answers can be found in the syllabus or lecture notes. The instructor and TA(s) *will not* reply to email that does not follow these guidelines.

Office Hours. Students are encouraged to bring in-depth questions and concerns to the instructor or TA during office hours. All office hours will be held online via Zoom (see the links at the top of this syllabus). Additional office hours are available by appointment.

Canvas & Slack. Canvas announcements will be the authoritative and complete source of updates from the instructor and TA to the students. Specific guidelines for Slack will be posted as pinned messages in each channel when applicable. Note that Canvas announcements will not be reposted in the Slack channel, though useful information arising from conversations on Slack may be consolidated into Canvas announcements.

Course Policies

Academic Integrity

Students in this class must adhere to the ASU Academic Integrity Policy. Students are responsible for reviewing this policy and understanding each of the areas in which academic dishonesty can occur. In addition, all engineering students are expected to adhere to both the ASU Academic Integrity Honor Code and the Fulton Schools of Engineering Honor Code. All academic integrity violations will be reported to the Fulton Schools of Engineering Academic Integrity Office (AIO). The AIO maintains record of all violations and has access to academic integrity violations committed in all other ASU college/schools.

Artificial Intelligence (AI)—including ChatGPT and other tools used for generating automated text, images, computer code, audio, or other media—are not permitted for use in any work in this class. Use of generative AI tools in this course will be considered a violation of the ASU Academic Integrity Policy and students may be sanctioned for confirmed, non-allowable use.

Violations of academic integrity regarding assignments and exams in this class will be severely dealt with. The *minimum* penalty for violations will be a zero-score for the assignment or exam in question. For individual (resp., group) assignments, students are *encouraged to discuss* problems with others, but each individual (resp., group) is expected to turn in the results of *their own effort* and not that of a friend's (resp., other group's). Especially in a theory course like this one, it is easy to spot when two submissions have similar answers. It is always better to submit your own (group's) work—even if your answers are incomplete or incorrect—than to violate academic integrity.

Copyright

All course content and materials — including recorded lectures, supplemental Zoom recordings, slides, notes, and assessments — are copyrighted materials. Students may not share outside the class, upload to online websites not approved by the instructor, sell, or distribute course content or notes taken during the conduct of the course (see ACD 304–06: Commercial Note-Taking Services for more info). Students must obtain verbal consent from the instructor before recording any live sessions or office hours, and such recordings are for personal use only. Students may not upload to any course shell, discussion board, or website used by the course instructor or other course forum, material that is not the student's original work, unless the students first comply with all applicable copyright laws. Faculty members reserve the right to delete materials on the grounds of suspected copyright infringement.

Threatening Behavior

Students, faculty, staff, and other individuals do not have an unqualified right of access to university grounds, property, or services. Interfering with the peaceful conduct of university-related business or activities or remaining on campus grounds after a request to leave may be considered a crime. All incidents and allegations of violent or threatening conduct by an ASU student (whether on- or off-campus) must be reported to the ASU Police Department and the Office of the Dean of Students. See SSM 104–02: Handling Disruptive, Threatening, or Violent Individuals on Campus for more details.

Accessibility and Disability Accommodations

Suitable accommodations will be made for students with disabilities or other short- or long-term accessibility needs. Students needing accommodations must register with the ASU Student Accessibility and Inclusive Learning Services (SAILS) and provide documentation of that registration to the instructor. Students should communicate the need for an accommodation in sufficient time for it to be properly arranged. See ACD 304–08: Classroom and Testing Accommodations for Students with Disabilities for more details.

Discrimination, Harassment, and Retaliation

Arizona State University is committed to providing an environment free of discrimination, harassment, or retaliation for the entire university community, including all students, faculty members, staff employees, and guests. ASU expressly prohibits discrimination, harassment, and retaliation by employees, students, contractors, or agents of the university based on any protected status: race, color, religion, sex, national origin, age, disability, veteran status, sexual orientation, gender identity, and genetic information.

Title IX is a federal law that provides that no person be excluded on the basis of sex from participation in, be denied benefits of, or be subjected to discrimination under any education program or activity. Both Title IX and university policy make clear that sexual violence and harassment based on sex is prohibited. An individual who believes they have been subjected to sexual violence or harassed on the basis of sex can seek support, including counseling and academic support, from the university. If you or someone you know has been harassed on the basis of sex or sexually assaulted, ASU encourages you to consult the ASU Sexual Violence FAQs for information and resources.

Please be aware that many university officials (including the instructor) are *mandated reporters*, meaning they are obligated to report any incidents of discrimination, harassment, or assault they become aware of to the Offices of Student and University Rights and Responsibilities. If you would instead like to report an incident confidentially, trained professionals at ASU Counseling Services are available to you.

Syllabus Disclaimer

The syllabus is a statement of intent and serves as an implicit agreement between the instructional team and the student. The instructor reserves the right to update this syllabus as needed. Any changes will be communicated to the class.

