Theory of Computing
CS 381 - Spring 2023
Credits: 3 hours

Instructor: David Furcy **Office:** Halsey 221 **Email:** furcyd@uwosh.edu

Office Hours: **MWF** 9:30 - 10:30, **TuTh** 12:30 - 2:00, or **by appointment**

Class Meetings: **TuTh** 9:40 - 11:10 in Halsey 208

Prerequisites: CS 271 and either CS 212 or Math 222, all with a grade of C or better.

Class Web Page: Canvas
You are expected to check the CS 381 Canvas course web site daily since it will be constantly updated with materials for this class including daily slides, daily reading assignments, daily practice/review problems, homework assignment handouts, etc. **You are also expected to check your email daily** for course announcements (with subject line always starting with "***CS 381***")

Required Text: *Introduction to Computer Theory* (**second** edition - 1997)
-- Daniel I. A. Cohen (J. Wiley & Sons, Publisher)

Tests: Exam #1: Week of March 6 (± one week) with 10-day prior notice
Exam #2: Week of April 10 (± one week) with 10-day prior notice
Exam #3: Week of May 8

Course Description: An introduction to the basic concepts in the theory of computing. Topics covered will include automata theory, formal languages, Turing machines, the Chomsky hierarchy, and undecidability.

Course Overview: As part of a liberal education, this course brings the rigor and formalism of mathematics to bear on philosophical and practical questions related to the complexity and even the limits of computation. We will analyze problem solving and computation from a language-processing perspective. We will discuss fundamental results in theoretical computer science and interpret them in the context of real-world computing. We will be concerned, for example, with answers to the following questions:

- What, in the abstract, is a *computing machine*?
- When are two such machines equivalent in computing power?
- What does it mean for a problem to be *computable*?
- Which problems are/are not computable and how can we tell them apart?
- What is an *algorithm*?
- What is the difference between languages and machines?

Course Outcomes:

1. Given a language recognizer and a string, the student will be able to decide whether or not the string is accepted by the recognizer.

2. Given an informal description of a language, the student will be able to categorize it (as a regular, context-free, recursively enumerable, etc. language) and to prove the correctness of this categorization.
3. Given an informal description of a language, the student will be able to build a computational model to recognize it.
4. Given a computational model and an alphabet, the student will be able to describe the language that it recognizes or generates.
5. Given a computational model, the student will be able to transform it into an equivalent computational model of a different type (for example, transforming a regular expression into a transition graph, or a context-free grammar into a pushdown automaton).
6. Given two regular languages, the student will be able to design a computational model that recognizes their intersection and union.
7. Given a formal language and a set of languages, the student will be able to apply the appropriate pumping lemma or a known closure property to prove that the language is not a member of the given set.
8. Given a formal grammar, the student will be able to judge whether it is ambiguous or not.
9. Given an ambiguous grammar for a language that is not inherently ambiguous, the student will be able to eliminate the ambiguity.
10. Given a formal grammar and a set of requirements pertaining to the allowed format of productions (for example, whether lambda or unit productions are allowed, or whether the grammar should be in Chomsky normal form), the student will be able to transform the given grammar into an equivalent grammar that meets the format requirements.
11. Given a regular or context-free language formally specified by a computational model, the student will be able to solve the finiteness and emptiness problems for this language.
12. Given a context-free grammar, the student will be able to generate the set of useless non-terminals and productions in the grammar.
13. Given a context-free grammar and a string, the student will be able to trace the CYK algorithm to solve the membership problem.
14. Given a deterministic Turing machine, the student will be able to encode it into a string (and vice-versa), and determine whether the machine accepts its own encoding.

Topic Coverage:

We will cover the following topics:

- Recursive definitions
- Regular expressions, finite automata (DFAs, TGs, NFAs), Kleene's theorem
- Context-free grammars, pushdown automata
- Regular and context-free languages and corresponding pumping lemmas
- Turing machines
- Recursive and recursively enumerable languages
- Church's thesis
- Decidability and algorithms

Course Grading Policy: Your overall grade for this course will depend on frequent (think, daily) quizzes, homework assignments, and exams. All assignments (quizzes, exams, respectively) will carry the same weight when computing your overall assignment (quiz, exam, respectively) grade. Your overall numerical grade for the course will be computed as the weighted sum of the component grades using the following weights:

Component	Weight
Quizzes	15%
Assignments	25%
Exams	60%

Your final letter grade for the course will be computed using the following mapping:

Numerical Score	Grade	Numerical Score	Grade
≥ 92	A	≥ 72	C
≥ 90	A-	≥ 70	C-
≥ 88	B+	≥ 68	D+
≥ 82	B	≥ 62	D
≥ 80	B-	≥ 60	D-
≥ 78	C+	< 60	F

While this overall grading scheme is fixed, I will be happy to discuss any issue you may have with individual grades. If you notice a mistake or have a question regarding a specific grade, please contact me *as soon as possible*. Do not wait until the end of the semester to bring up grading issues. Also, I will *not* be available to discuss grades after the end of the final week.

Attendance and Participation: You are expected to not only attend **every** class meeting but also to come **prepared** for and **participate** actively in it. Necessary preparation requires you to have studied and assimilated the material covered in previous sessions, to have met with me outside of class to discuss any questions you may have, to have done the assigned reading, and to have completed the assignments on time.

It is hard to imagine how a student could do well in this course while missing classes, attending them unprepared, or not participating.

On the positive side, I have high expectations for my students and will always support and encourage you. I **strongly encourage** you to **ask any question** or raise any issue you have with the course either during or at the end of class, or during my office hours. I will also gladly meet with you by appointment. Send me email or give me a call to make an appointment. While I will meet with you as soon as my schedule permits, do not expect me to be widely available just before an exam or assignment's due date, since (many) other students may have waited until the last minute to ask for help.

Late Submissions: I will describe the submission procedure for your assignments when the time comes. However, let me point out right away that each assignment will come with a deadline (day and time, typically 8:00 AM) after which any submission will be considered late. The late-submission policy works as follows:

Turned in	Penalty
On the due date but after the deadline	30%
The day after the due date	60%
More than one day after the due date	100%

Note that assignments that are two or more days late receive no points. Weekend days and holidays count as "regular days" when computing late penalties. The late day starts precisely at midnight. So, each one of the following timestamps: 12:00:00 AM, 12:00:01 AM, etc., is considered to be “the next day.” Extensions on assignments may be granted at my discretion if you provide a valid justification (in the form of a written excuse from a medical doctor or the Dean of Students Office) before the due date. Late submissions can easily be avoided by starting to work on each assignment right away and asking for help early if you get stuck.

If you miss an exam, you **may** be able to take a make-up exam provided you give me a valid justification (see above) ahead of time if possible. Only one make-up exam will be given. It will be a comprehensive exam scheduled at the end of the semester. If you miss a quiz, you **may** be able to take a make-up quiz, provided you have a valid justification (in writing) for your absence.

Accommodations

The University of Wisconsin Oshkosh supports the right of all enrolled students to a full and equal educational opportunity. It is the University’s policy to provide reasonable accommodations to students who have documented disabilities that may affect their ability to participate in course activities or to meet course requirements.

Students are expected to inform instructors of the need for accommodations as soon as possible by presenting an Accommodation Plan from either the Accessibility Center, Project Success, or both. Reasonable accommodations for students with disabilities is a shared instructor and student responsibility.

The Accessibility Center is part of the Dean of Students Office and is located in 125 Dempsey Hall. For more information, email accessibilitycenter@uwosh.edu, call 920-424-3100, or visit the Accessibility Center Website.

Disclosure: Students are advised to see the following URL for disclosures about essential consumer protection items required by the Students Right to Know Act of 1990:

`https://uwosh.edu/financialaid/resources/consumer-information/`

Collaborating versus Cheating

All submissions must be the work of only one student, namely the one whose name appears on the submission. While it is acceptable and encouraged to discuss (with other students) the problem statements, premises, goals, constraints, etc., in an assignment, you must submit your **own** work **exclusively** unless you can live with a zero and the other potential academic sanctions of cheating (see the UWO Student Discipline Code, Chapter UWS 14).

You are of course encouraged to ask me for help if you get stuck after making a good faith effort at solving a problem. That is why I have daily office hours and available appointments.

In conclusion, remember that computer science classes require a lot of work in addition to active participation in class. It takes considerable practice to develop the technical and analytical skills targeted by this course. You will need to spend **at least (and typically more than) three hours**

of effort outside of class for each in-class hour. Having said this, I expect every hardworking student to do well in this course.

Have fun this semester and good luck!