Syllabus in Spring 2019 for CS 420/520 Theory of Computation

General Information

Contact Your Instructor

Name: Jeff Kinne
Email: jkinne@cs.indstate.edu
Phone: 812-237-2126 (but this is not the best way to get ahold of me)
Office: Root Hall A-140D

Lecture, Exam, Office Hours

Lecture: Tuesdays and Thursdays 11am-12:15pm in Root Hall A-012
Section and CRN: section 001, CRN 12735
Credit Hours: 3

Exam: Thursday May 9 from 10am-11:50am.

Instructor Office Hours: MWF 9am-noon (except the 4th Wednesday of the month). I am also normally around MWF afternoons. Tuesdays and Thursdays are mostly full of meetings.

GA Tutoring: See http://cs.indstate.edu/info/labs.html

Website: http://cs.indstate.edu/~jkinne/cs420-s2019

Prerequisites

A grade of C or better in both CS 202 and CS 303, or consent of instructor.

Note - A 500 level course cannot be taken as part of the MS program if you have completed the same 400 level course at ISU. For example, if you completed CS 451 as an undergrad, you **cannot** take CS 551 as an MS student and count it towards the degree.

Recommended and/or Required Text

The recommended textbook that we will follow is Introduction to the Theory of Computation by Michael Sipser. Note that the book is not required. There is sufficient material from online sources to supplement the lectures if you choose not to purchase the book. If you do choose to purchase the book, you can purchase any edition (there are 3 editions currently).

We will also loosely follow the course structure of a version of this course taught by Michael Sipser at MIT which has lecture notes online. For mathematical background (i.e., discrete mathematics) you can check the MIT course Mathematics for Computer Science which has lecture notes and solved problems. If you want a discrete math book, try out an older edition of http://www.mhhe.com/math/advmath/rosen/

Course Announcements

Announcements regarding the course will be made both during class and via email to your @sycamores.indstate.edu email address. You should regularly check this email account or have it forwarded to an account that you check regularly.

Classroom conduct

You may not use cell phones, iPods/music players, etc. during class. You should be civil and respectful to both the instructor and your classmates, and you should arrive to class a few minutes before the scheduled lecture so you are ready for lecture to begin on time. You may use your computer during class if you are using it to follow along with the examples that are being discussed. You may not check email, facebook, work on other courses, etc. during class.

Course Description

The official description of this course from the catalog is

"A sampling of the different areas of theoretical computer science: finite state concepts, formal grammars and automata, computability, Turing machines, and program verification."

Theory of computation studies different ways that computation can be performed. We start the semester by studying very simple computing devices (finite state machines) and explore what can and cannot be done by these machines. Throughout the semester we consider different computational devices of varying power, ultimately concluding by studying problems that are impossible to solve using any type of computational device.

Traditional theory of computation put the most focus on being able to formally prove what can and cannot be accomplished on each type of computational device, meaning the theory of computation course encompasses techniques and knowledge from both computer science and mathematics. Our version of this course will keep to this tradition to a certain extent, but we will also explore the concepts at times using programs. There will be some homework assignments given using the Python programming language (Python 3).

Course Outline and Learning Outcomes

We begin the course by following along with this Theory of Computation Course at MIT. Along the way, we will get a view of each of the following topics.

- **Review of discrete mathematics:**
  - Proof types – induction, contradiction, construction
  - What is a proof…
- **Regular languages:**
  - Finite automata (DFA):
    - Formal definition of finite automata, examples
    - Formal definition of computation, designing finite automata
    - Regular operators
  - Nondeterminism:

        - Formal definition of a nondeterministic finite automaton (NFA)
        - Equivalence of NFAs and DFAs (with proofs)
        - Closure under regular operations (with proofs)
    - Regular expressions:
        - Formal definition of a regular expression
        - Equivalence with finite automata (with proofs)
        - Use case of how to use regular expressions in a modern programming language
    - Nonregular languages
        - The *pumping lemma for regular languages* (with proofs, students are expected to master how to use this lemma that certain languages are nonregular)
    - *Outcomes - Given the description of a language, able to classify as regular or non-regular and give a proof. Able to give correct regular expressions for simple languages in a modern programming language. Able to convert between DFAs, NFAs, and REs.*
- **Context-free languages:**
    - Context-free grammars (CFG)
        - Formal definition of CFGs, examples of CFGs
        - Chomsky normal form
    - Pushdown automata (PDA)
        - Formal definition of PDA, examples
        - Equivalence of PDAs with CFGs, proof not covered in class
    - Non-context-free languages: Pumping lemma for CFGs (students are expected to master how to use this lemma that certain languages are not context-free)
    - *Outcomes - Given description of a language, able to classify as context free or not context free and give a proof. Able to convert between CFGs and PDAs.*
- **Church-Turing thesis and decidable languages**
    - Turing machines: formal definition, examples
    - Variants of Turing machines, nondeterministic Turing machines, enumerators
    - Definition of an algorithm
    - Decidable languages concerning regular languages, context-free languages
    - *Outcomes - Able to apply the definition of a Turing Machine and explain how it is a reasonable approximation of computers (and what is different in real computers).*
- **Undecidability:**
    - Diagonalization method, examples of undecidable and Turing-unrecognizable languages
    - *Outcomes - Can explain diagonalization proofs to prove a language is not decidable.*
- **Reducibility:**
    - Some undecidable problems, reduction via computation histories
    - Computable function, mapping reducibility
    - *Outcomes - Able to give reductions between undecidable problems, and between decidable problems.*
- **Complexity Theory:**
    - L, NL, P, NP, PSPACE, EXP: definitions and basic properties
    - Approximation algorithm, hardness of approximation: basic examples and properties

- *Outcomes - Given the description of a language, able to prove which complexity classes the language is within. Able to explain the difference between complexity classes, and what some key languages are in each complexity class.*

Expected Amount of Work

If you take this class seriously and get what you should out of it, most weeks you will likely be spending ***3-9* hours/week** on the class. The students who get A's in their CS courses and have an easy time finding jobs do spend this much time on this course. Not everyone would need to spend this much time and not all weeks will be the same, but you should plan on putting in whatever time it takes.

Note - your classes should be more important than your part-time job.

Grading and Assignments

The students of this course have the following responsibilities: read assigned readings before lecture, attend lecture, complete homework assignments, take in-class quizzes, take exams, and complete a project (maybe on the project).

Your grade in each of the following categories will be calculated. Your "total" grade will be the minimum of these.

1. **Exams** - we will have three exams, which will all be cumulative. The total exams grade will be max(exam3, .6*exam3 + .4*exam2, .5*exam3 + .3*exam2 + .2*exam1).
2. **Projects** - some of the homework assignments may be labeled as projects. The projects grade will be the simple average of the project grades. Each project will receive a score for style, correctness, inclusion of required features, and inclusion of advanced features.
3. **HW** - each HW is given a number of points, and the total HW grade comes from just adding up all of the HW's (so HW's worth more points are worth more in the total HW grade)
4. **Quiz** - same as with HW's, but for in class quizzes.
5. **Attendance** - I will take attendance every lecture for the first three weeks, and then regularly afterward. Your attendance grade is just the % of time you were in class (on time, paying attention, I'll give you 2 free absences).

CS Course Policies

Note that this course follows all standard CS course policies. In particular check the CS course policies related to - cheating/plagiarism, attendance, missing exams. See http://cs.indstate.edu/info/policies.html for details.

Late Homeworks

When an assignment is given, the assignment will state if late work will be accepted. For some assignments, a "full credit" date and "late credit" date will be given. By default, "late credit" is at most ½ of the points. If no "late credit" date is given for an assignment, then no late credit is given for the assignment.

Some key assignments will be labeled as "**checkpoint assignments**". For these assignments, if you do not complete the assignment correctly by the due date, **you fail the course**. These assignments are labeled as "checkpoint assignments" to make it clear to you that you cannot pass the course if you do not complete them on time.

Start Homeworks Early

We suggest attempting a homework assignment the day it is given, or the day after, so that if you have a problem you can ask early. If you continue to have problems in trying to complete the assignment, you will have time to ask again. Many of the homework assignments require thought and problem solving, which takes "time on the calendar" not just "time on the clock". By that we mean that spending two hours on 3 consecutive days may be more productive than trying to spend 6 hours at once on the assignment.

Grade Cutoffs

We try to design homework assignments and exams so that a standard cutoff for grades will be close to what you deserve. After the first exam a grade will be created in Blackboard called "Letter Grade" that is what your letter grade would be if the semester ended today. Initially, I will likely assign the following grades: 93-100 A, 90-93 A-, 87-90 B+, 83-87 B, 80-83 B-, 77-80 C+, 73-77 C, 70-73 C-, 67-70 D+, 63-67 D, 60-63 D-, 0-60 F

Our goal is that the different grades have the following rough meaning.

A+/A
You can do *all* the assignments *on your own*.

B+/A-
You understand nearly everything, and should be all set to use this knowledge in other courses or in a job.

B-/B
Most things you understand very well and a few you might not (more towards the former for a B and more towards the latter for a C).

C/C+
Learned enough and have the minimum skills to move on in the subject.

D+/C-
You did put some effort in, and understand many things at a high level, but you haven't mastered the details well enough to be able to use this knowledge in the future.

D-

Students will normally *not* get an F if - you attend 80% of the lectures, complete some of the assignments up through the end of the course, and get nearly half of the problems on the final exam correct.

F
Normally, students that get an F simply stopped doing the required work at some point.

Blackboard

The course has a blackboard site. Click here to go to blackboard. You should see this course listed under your courses for the current term. The blackboard site is only used for giving you your grades (go to the course in blackboard, then click "My Tools", and then "My Grades"). All course content, schedule, etc. is kept in this google doc (which you are currently viewing).

Academic Integrity

Follow the standard CS course policies in terms of what is and is not allowed on assignments: http://cs.indstate.edu/info/policies.html

Please ask the instructor if you have doubts about what is considered cheating in this course.

Special Needs / Student Disabilities

Standard language included in the syllabi for ISU courses.

Indiana State University recognizes that students with disabilities may have special needs that must be met to give them equal access to college programs and facilities. If you need course adaptations or accommodations because of a disability, please contact us as soon as possible in a confidential setting either after class or in my office. All conversations regarding your disability will be kept in strict confidence. Indiana State University's Student Support Services (SSS) office coordinates services for students with disabilities: documentation of a disability needs to be on file in that office before any accommodations can be provided. Student Support Services is located on the lower level of Normal Hall in the Center for Student Success and can be contacted at 812-237-2700, or you can visit the ISU website under A-Z, Disability Student Services and submit a Contact Form. Appointments to discuss accommodations with SSS staff members are encouraged.

Once a faculty member is notified by Student Support Services that a student is qualified to receive academic accommodations, a faculty member is obligated to provide or allow a reasonable classroom accommodation under ADA.

Disclosures Regarding Sexual Misconduct

Standard language included in the syllabi for ISU courses.

Indiana State University fosters a campus free of sexual misconduct including sexual harassment, sexual violence, intimate partner violence, and stalking and/or any form of sex or gender discrimination. If you disclose a potential violation of the sexual misconduct policy I will need to notify the Title IX Coordinator. Students who have experienced sexual misconduct are encouraged to contact confidential resources listed below. To make a report or the Title IX Coordinator, visit the Equal Opportunity and Title IX website: http://www.indstate.edu/equalopportunity-titleix/titleix.

The ISU Student Counseling Center – HMSU 7th Floor | 812-237-3939 | www.indstate.edu/cns
The ISU Victim Advocate – Trista Gibbons, trista.gibbons@indstate.edu
HMSU 7th Floor | 812-237-3939 (office) | 812-230-3803 (cell)
Campus Ministries - United Campus Ministries | 812-232-0186
http://www2.indstate.edu/sao/campusinistries.htm
www.unitedcampusministries.org | ucmminister2@gmail.com
321 N 7th St., Terre Haute, IN 47807

For more information on your rights and available resources
http://www.indstate.edu/equalopportunity-titleix/titleix