CS 3613: Theoretical Foundations of Computing **Fall 2019**
Course Information (Last Revised on August 19, 2019)

---

1. General Information

	Instructor: H. K. Dai	Teaching Assistants: [refer to class Webpage]
Office Location:	Mathematics, Statistics, and Computer Science Building	
	Room 209	Room
Office Hours:	Monday/Wednesday 10:30 - 11:30 (or by appointment)	
Office Phone:	744-7207	
email Address:	dai@cs.okstate.edu	
Universal Resource Locator:	http://www.cs.okstate.edu/~dai/	

2. Course Description in Current University Catalog

CS 3613: Theoretical Foundations of Computing. Prerequisites: CS 2133 (Computer Science II) and CS 3653 (Discrete Mathematics for Computer Science). Introduction to the classical theory of computer science. Sequential machines and their applications to devices, processes and programming. Models of computation: finite-state automata, push-down automata, Turing machines. The role of non-determinism. Limits of digital computation. Computability and unsolvability. The Church-Turing Thesis.

3. Course Goals

The goal of CS 3613 is to give students an ability to develop and rigorously reason about abstract formal models of computations, and to learn the powers and limitations of such formalism. "Classical" models, such as finite automata / regular expressions, pushdown automata / context-free grammars, and Turing machines will be studied in depth.

4. Course Materials and References

1. Text: [Sip12] M. Sipser. *Introduction to the Theory of Computation.* Cengage Learning, Recent Edition (3rd edition, 2012).
2. Reference: [Mar10] J. A. Martin. *Introduction to Languages and the Theory of Computation.* McGraw-Hill, Recent Edition (2010).
3. Lecture notes (sketchy): from course instructor.
4. Class pages (http://www.cs.okstate.edu/~dai/course/CS3613/2019fall/2019fall.html).

5. Homework and Examinations

There will be about 5-6 homework assignments, a few unannounced quizzes, one test, and one final examination.

6. Course Grade

The course grade is based on the homework (30%), unannounced quizzes (10%) and test (25%), and final examination (35%). The passing letter-grade is determined by the following partition of the course grades:

D : [50, 60); C : [60, 70); B : [70, 85); and A : [85, 100]

7. Miscellaneous

1. **Lectures**: Lectures are not mandatory, but historically, students with active attendance have done significantly better on examinations than their less frequently attending classmates.
2. **Homework**: Problem sets form an important part of the learning in the course, and thus, you are required to do them in order to pass.
3. **Collaboration**: You are encouraged to collaborate in study groups on the solution of the homework. If you do collaborate you must write up solutions on your own and acknowledge your collaboration in the write-up for each problem. If you obtain a solution with help (e.g., through library work, another student, etc.), acknowledge your source, and write up the solution on your own.

8. Student Disability Services

Student Disability Services and other Student Services are committed to providing support services to students with physical and learning disabilities. Please advise the instructor of desired academic accommodations, and notify Student Disability Services.

9. Academic Dishonesty or Misconduct

Refer to the section in "University Academic Regulations" in current "University Catalog"
(http://registrar.okstate.edu/)

10. Adding/Dropping/Withdrawing, Important Dates, and Syllabus Attachment

1. **Test and Final Examination**: Tentative date for the test is October 2 (Wednesday), 2019.

   Adopting "Fall 2019 Final Exam Schedule", the firm time/date for final examination is 10:00 – 11:50 am, December 11 (Wednesday), 2019 in regular class meeting place.

   Refer to the section in "Fall 2019 Final Exams":
   http://registrar.okstate.edu/Exams

2. **Adding/Dropping/Withdrawing and Important Dates**: Refer to the section in "Academic Calendar":
   http://registrar.okstate.edu/

3. **Syllabus Attachment**: Refer to:
   http://academicaffairs.okstate.edu/content/resources-students

CS 3613: Theoretical Foundations of Computing   Fall 2019
Course Outline (Tentative)   (Last Revised on August 19, 2019)

1. ABET (Accreditation Board for Engineering and Technology, Inc.): Student Outcomes

1. Analyze a complex computing problem and to apply principles of computing and other relevant disciplines to identify solutions.
2. Design, implement, and evaluate a computing-based solution to meet a given set of computing requirements in the context of the program's discipline.
6. Apply computer science theory and software development fundamentals to produce computing-based solutions. [CS]

2. ACM (Association for Computing Machinery): The Body of Knowledge

Knowledge Area	Total Hours of Coverage
Algorithms and Complexity (AL)	38

3. Body of Knowledge Coverage

Knowledge Area	Knowledge Unit	Topics Covered	Hours
AL	Basic Automata Computability and Complexity	all core-tier 1: finite-state machines, regular expressions, and the halting problem	3
		core-tier 2: context-free grammars, and introduction to the P and NP classes and the P versus NP problem	3
AL	Advanced Automata Theory and Computability	elective: sets and languages (regular languages, review of deterministic finite automata (DFAs), nondeterministic finite automata (NFAs), equivalence of DFAs and NFAs, review of regular expressions and their equivalence to finite automata, closure properties, and proving languages non-regular, via the pumping lemma or alternative means)	12
		elective: context-free languages (push-down automata (PDAs), relationship of PDAs and context-free grammars, properties of context-free languages)	11
		elective: Turing machines or an equivalent formal model of universal computation, nondeterministic Turing machines, and the Church-Turing thesis	9

4. Course Outline (Tentative)

1. Mathematical Preliminaries and Introductory Material
   Source: Lecture Notes, and [Sip12] Chapter 0

2. Regular Languages: Finite Automata (FAs) and Regular Expressions
   Source: Lecture Notes, and [Sip12] Chapter 1
   2.1 Deterministic Finite Automata (DFAs); and Nondeterministic Finite Automata (NFAs)
   2.2 The Equivalence of DFAs and NFAs
   2.3 Regular Expressions and Regular Languages
   2.4 The Equivalence of Regular Expressions and Finite Automata
   2.5 The Pumping Lemma for Regular Languages

3. Context-Free Languages: Context-Free Grammars (CFGs) and Pushdown Automata (PDAs)
   Source: Lecture Notes, and [Sip12] Chapter 2
   3.1 CFGs and Context-Free Languages (CFLs)
   3.2 PDAs
   3.3 The Equivalence of CFGs and PDAs
   3.4 The Pumping Lemma for CFLs

4. Computability
   Source: Lecture Notes, and [Sip12] Chapters 3 and 4 (and 5, if time permits)
   4.1 Turing Machines (TMs); and Their Variations
   4.2 The Church-Turing Thesis
   4.3 Introductory Computability/Decidability