University of Rhode Island

Department of Computer Science and Statistics
CSC445, Models of Computation, Section 1, Spring 2016

Instructor:	Lutz Hamel
Office Location:	Tyler Hall 251
Telephone:	401-480-9499
Email:	hamel@cs.uri.edu
Office Hours:	TBA
Class Days/Time:	TueTh 11:00am – 12:15pm,
Classroom:	Bliss Hall Rm 211
Prerequisites:	CSC340
Webpage:	http://homepage.cs.uri.edu/faculty/hamel/courses/2016/spring2016/csc445

Course Description

Computation and algorithms seem to occur naturally in our daily lives. Consider counting change or following a recipe to make your favorite dish. The art of programming takes this to the limit by formally encoding computations and algorithms for a machine to follow. What are the limits of this algorithmic approach? Are there mathematical objects that cannot be computed by a machine via an algorithm? If we can express a computation via an algorithm how long would it take to compute? How much space would the computation consume?

In this course we will investigate many of these interesting questions. We start with simple models of computations such as the finite state machine and the push down automaton. As our main tool we will use an idealized general purpose computer invented by Alan Turing: the Turing Machine. This idealized machine allows us to study the limits of computability and the complexity of computations without having to worry too much about actual hardware.

Course Goals

- To be exposed to the terminology of the theory of computing
- Familiarity with some of the major results in computability and complexity theory.

- A basic understanding of the major models of computations.

Upon successful completion of this course, each student will be able to:

- understand the formal notion of a language
- understand language hierarchies
- understand the idea of language generators and processors
- solve basic problems in theoretical computer science

Required Texts/Readings

Textbook

Formal Language: A Practical Introduction, Adam Brook Webber, Franklin, Beedle & Associates, Inc., 2007.

Classroom Protocol

- Check the website (often)! I will try to keep the website as up-to-date as possible.
- Class **attendance, promptness, participation,** and **adequate preparation** for each class are expected. If you are absent, it is your responsibility to find out what you missed (e.g. handouts, announcements, assignments, new material, etc.)
- **Late assignments** will **not** be accepted.
- **Make-up quizzes** and **exams** will **not** be given without a valid excuse, such as illness. If you are unable to attend a scheduled examination due to valid reasons, please inform myself, or the department office in Tyler Hall, prior to the exam time. Under such circumstances, you are not to discuss the exam with any other class member until after a make-up exam has been completed.
- All work is to be the result of your own individual efforts unless explicitly stated otherwise. **Plagiarism, unauthorized cooperation or any form of cheating** will be handled according to the University Man- ual section 8.27.10 through 8.27.21 (see www.uri.edu/facsen/8.20-8.27.html). The penalty for cheating or plagiarism can range from a zero score on the assignment to a failing grade for the course.
- **Software piracy** will be dealt with exactly like stealing of university or departmental property. Any abuse of computer or software equipment will subject to disciplinary action.

Exams, Assignments, and Grading Policy

Course Grade Composition:

Homework	50%
Midterm	25%
Final	25%

Grading Scale:

95 - 100 A

90 - 94.9 A-

85 - 89.9 B+
80 - 84.9 B
75 - 79.9 B-
70 - 74.9 C+
65 - 69.9 C
60 - 64.9 C-
55 - 59.9 D+
50 - 54.9 D
0 - 49.9 F

Homework consists of exercises to familiarize you with common tools and concepts in theoretical computer science. Homework is assigned on a weekly basis. The midterm and final are open book/open notes exams and are in style and content similar to the homework assignments.

Disability Accommodations and Opportunities

Any student with a documented disability should contact me early in the semester so that we can make reasonable accommodations to support your success in this course. You should also contact Disability Services for Students, Office of Student Life, 330 Memorial Union, 874-2098

Tentative Course Schedule

week 1	Fundamentals
week 2	Closure Properties For Regular Languages
week 3	Deterministic Finite Automata (DFA)
week 4	Nondeterministic Finite Automata (NFA)
week 5	Regular Expressions
week 6	Grammars
	- Non-Regular Languages
	- Context-Free Languages
week 7	Stack Machines
	- The Context-Free Frontier
	- Stack Machine Applications
week 8	Turing Machines
week 9	Computability
week 10	Uncomputability
week 11	Complexity Classes
	- Deterministic Complexity Classes
	- Nondeterministic Complexity Classes