CS 330: Formal Methods and Models, Section 003, Spring 2017

Prerequisites

- CS211 and Math 125 (C or better in both).
- The exposure to Discrete Mathematics (in Math 125 or a similar course) for most students is important for success in this course.

Description

This course is an introduction to two kinds of formal systems - languages and logics - with important applications to computer science. The study of formal languages underlies important aspects of compilers and other language processing systems, as well as the theory of computation. Various systems of logic and automatic reasoning are put to use in artificial intelligence, database theory and software engineering. The entire course will give you practice in precise thinking and proof methods that play a role in the analysis of algorithms. The programming assignments in Lex and Prolog provide practical experience with some theoretical topics.

Course Outcomes

- Will understand the concepts and relevance of logic, formal languages and automata theory, and computability.
- Will be able to able to do mechanical formal proofs, program correctness proofs and solve problems in first-order logic.
- Will be able to solve problems in elementary machine models: designing finite-state, pushdown and Turing machines.
- Will be able to solve problems in formal languages: writing regular expressions, regular grammars, and context-free grammars.

Textbook

- Title: Logic and Languages Models for Computer Science
- Authors: Dana Richards and Henry Hamberger, 3rd edition.
    - Available from GMU bookstore

Class Details

- Section 03:
- Class Times: Tuesday, Thursday 3.00-4.15
- Location: Merten Hall 1200

Course Personnel

Instructor

- Name: Duminda Wijesekera
- Contact Email: dwijesek (at) gmu (dot) edu
- Contact Phone: 703-993-5030
- Office Hours: Tuesdays and Thursdays 4.30 pm to 5.30 pm or by appointment
- Office Location: Room 436, Research Building
- Best way to reach me: email

Teaching Assistant

- TA Name: TBD
- TA Email: TBD
- Office hrs: TBD
- Office Location: TBD

Syllabus

Topic	Week	Chapter/ Parts
Introduction	1	1
Propositional Logic and Proofs	1-2	2-3
Predicate Logic and Proofs	3-4	4-5
Applications: Prolog and Verification	5-6	Appendix A, 6
Exam #1	6	1-7
Finite Automata, Regular Expressions	7-9	8-9
AWK	10	Appendix B
Context-Free Grammars & Applications	11-12	10-11
Turing Machines & Solvability	13-14	12

Grading

Quizzes 25%
Programs 25%
Exams 50%

The two exams, including the final, each cover about a half of the semester. The final is not cumulative.

Notes

- The two exams, including the final, each cover about a half of the semester.
- The final is not cumulative.
- Homework is ungraded.
- Quizzes will test homework, typically every other class.
- The lowest quiz grades will be dropped.
- There will be small programming assignments in Prolog and in Lex (or AWK).

All testing is closed book, but limited notes are permitted, as follows for exams (but not for quizzes). One sheet of notes (8.5 by 11 inches, 1 side only). NO COPYING is allowed. That means no photocopying of anything, even the textbook, though you may write out material from it verbatim. It also means no copying of anyone else's notes, even by hand. You may use a computer for editing your own notes. The sheet must be turned in with your exam. Violations of these rules for creating the notes is considered a violation of the Honor Code.

There is to be NO group work on the programs. Receiving direct contributions to the code that is submitted is considered a violation of the Honor Code. (See cs.gmu.edu/wiki/pmwiki.php/HonorCode for the GMU and Computer Science guidelines.)

Lateness

Programs will be marked down 25% each class they are late; in particular, it is marked down 25% after the due date.

IN ORDER TO GET A PASSING GRADE each student will also meet at least once with his/her academic advisor during the semester.

Honor Code

All violations are reported. GMU honor code