CS 135 Discrete Structures – Spring 2015

January 20, 2015

Instructor:

Michael Engling (Pipeline ID: mengling)

Office Hours:

Lieb 314 Tuesday 12–3 also by appointment and by chance

Teaching Assistants:

Maryam Vatankhah	(mvatankh@stevens.edu)
Kevin Furlong	(kfurlong@stevens.edu)
Dominik Jedruszczak	(djedrusz@stevens.edu)

Lectures:

Section B	MWF	9:00–9:50	EAS 330
Section A	MWF	10:00–10:50	EAS 330

Lab:

Lab A	Kevin Furlong	Tuesday	9:00–10:40	Burchard 714
Lab B	Dominik Jedruszczak	Tuesday	11:00–12:40	Burchard 514
Lab C	Kevin Furlong	Tuesday	3:00–4:40	Babbio 220
Lab D	Maryam Vatankhah	Tuesday	10:00–11:40	Babbio 210
Lab E	Dominik Jedruszczak	Tuesday	3:00–4:40	Morton 105
Lab F	Maryam Vatankhah	Tuesday	2:00–3:40	EAS 229A

Required textbooks:

Discrete Mathematics and Its Applications, 7th edition 2011, by Kenneth Rosen, ISBN-10: 0-07-338309-0; ISBN-13: 978-0-07-338309-5
You must have the seventh edition!

The Little Schemer - 4th Edition, by Daniel P. Friedman and Matthias Felleisen, ISBN 0-262-56099-2

Software: The course employs *DrRacket* which is free and available for many platforms. Students will download it during the first lab session. http://racket-lang.org/

Prerequisite: There is no course prerequisite, but previous programming experience should prove helpful.

Outcomes:

1. **Formal logic** – Unfold a definition, apply a theorem, use deductive rules, for example in proving that a given relation is a function or a function is injective.
2. **Relations** – Define basic properties of relations such as transitivity and injectivity, in words and using logical formulas.
3. **Scheme** – Implement recursive functions on lists, such as append or reverse. For relations represented by lists of pairs, implement operations such as transposition and intersection.
4. **Induction** – Use induction and equational reasoning to prove equations about recursive functions, such as associativity of append.
5. **Modular arithmetic** – Simplify expressions by using laws of modular arithmetic.
6. **Primes** – Define basic properties such as relative primality. Implement Euclid's algorithm in Scheme.

Grading: The course score is a weighted average of the following categories.

- lab 20% (including pop quizzes from lecture)
- homework 20%
- first exam 15%
- second exam 20%
- final exam 25%

- Each lab will consist of several parts. Your lab grade will be based on how many parts you are able to complete correctly during lab time.
- The course score is on a scale of 100 and letter grades (including plusses and minuses). Final letter grades will be assigned according to class-wide grade clustering.
- Any dispute about an assignment grade must be made and resolved within 5 days of receiving your grade. After this period your grade cannot be adjusted.

Policies:

- You, your instructor, and the TAs are bound by the Stevens Honor Code. Students are responsible for reading and understanding the course policies in this syllabus and for announcements made in class and in the course email list.
- You will be permitted to use the textbooks and course notes for programming assignments (homework and labs). During exams, you are **not** permitted to use notes, books, computing or communication devices unless a different policy is specifically announced by the instructor.
- Except when groups are explicitly allowed, work must be done individually. You are encouraged to discuss the problems with your classmates but you must not share details of the solutions. If you are unsure whether you have shared too much, discuss the situation with the TA or instructor; it is your obligation to avoid even the appearance of cheating.
- During lecture and lab sessions please refrain from using mobile phones or otherwise being impolite.
- Attendance will be taken at both lectures and labs (attendance is mandatory). **You are permitted 1 missed lab/quiz** (which will be adjusted for at the end of the course).
- Labs are closed: the assignment is given, completed, and graded during the lab session. Group work is at the discretion of the TA.
- There are no make-ups for labs or exams. The only possible exceptions are in the case of death in the students immediate family or near-death experience of the student; advance notice is required.
- There may be short, unannounced quizzes in class, which count in the lab/quiz grade category. The purpose is to motivate attendance and to help both you and your instructor gauge your progress.
- Homework will **NOT BE ACCEPTED LATE**. It doesn't matter if it is 1 second late or 2 weeks late. I highly suggest you submit your work **hours** before it is due and download it yourself to make sure it submitted correctly. However I understand that things happen and therefore your lowest homework grade will be dropped at the end of the course.
- Homework will be due **Sunday night at 11:59pm** unless otherwise stated.

Week	Dates	Topic	Reading	Comment
0	Jan 19–23	Preliminaries Propositional Logic Lab 0: DrRacket/Scheme	1.1–1.3; LS 1	Martin Luther King Day No Class Monday HW 0 Due
1	Jan 16–30	Propositional Logic Predicate Logic Lab 1: Logical Operations	1.4–1.5	HW 1 Due
2	Feb 2–6	Arguments Introduction to Proofs Lab 2: Operations on Lists	1.6–1.8; LS 2	HW 2 Due
3	Feb 9–13	Sets, Functions, and Sequences Lab 3: Recursion on Lists	2.1–2.6; LS 3	HW 3 Due
4	Feb 16–20	More Sets? Recursion NO LAB	5.4; LS 4	Presidents' Day No Class Monday Tuesday is Monday
5	Feb 23–27	Recursion Induction Lab 4: Sets	5.1,5.3	EXAM I
6	Mar 2–6	Number Theory Integers, Primes, Bases Lab 5: Modular Arithmetic	4.1–4.3	HW 4 Due
7	Mar 9–13	Relations Lab 6: Lists with Ordered Pairs	9.1–9.3	
8	Mar 16–20	SPRING RECESS		NO CLASS
9	Mar 23–Mar 27	Compositions of Relations Closures of Relations Lab 7: Relations	9.4	HW 5 Due
10	Mar 30–Apr 3	Equivalence Relations Partial Orders Lab 8: Closures	9.5 & 9.6	Good Friday No Class Friday
11	Apr 6–Apr 10	Number Theory Integer Algorithms Lab 9: Topological Sort	4.2 (again)	HW 6 Due
12	Apr 13–Apr 17	Chinese Remainder Theorem Fermat's Little Theorem Lab 10: Integer Algorithms	4.5 & 4.6	HW 7 Due
13	Apr 20–24	Graphs Lab 11: Congruences	10.1–10.3	EXAM II
14	Apr 27–May 1	Trees Lab 12: Symmetric Crypto	11.1–11.3	
15	May 4–8	Review for Final Exam Lab 13: RSA	Review Classes End Wednesday	Final Exam TBD Date / Time / Room Wednesday is Friday

Catalog description: The aim of this course is to integrate knowledge of basic mathematics with problems involving specification, design, and computation. By the end of the course, the student should be able to: use sets, functions, lists, and relations in the specification and design of problems; use properties of arithmetic, modular arithmetic (sum, product, exponentiation), prime numbers, greatest common divisor, factoring, Fermat's little theorem; use binary, decimal, and base-b notation systems and translation methods; use induction to design and verify recursive programs; and implement in Scheme all algorithms considered during the course.