

CS 341: Foundations of Computer Science II

eLearning Section Syllabus, Spring 2014

Course Info

Instructor: Prof. Marvin K. Nakayama

Office: GITC 4312

Phone: 973-596-3398

E-mail: marvin@njit.edu (**Be sure to include “CS341EL” in the subject line**)

Only send e-mails about personal matters (e.g., your performance in the class).

Post any general questions on moodle (described below).

Office Hours: Wednesday, Thursday, 1:10 – 2:10, and by appointment.

Course Webpage: <http://web.njit.edu/~marvin/cs341>

Description

This course presents some of the most fundamental results in theoretical Computer Science. These results attempt to answer, in a precise mathematical sense, the following two questions, which are of practical as well as philosophical interest:

1. Can a given problem be solved by computation?
2. How efficiently can a given problem be solved by computation?

We often focus on *problems* rather than on specific *algorithms* for solving problems. To answer both questions mathematically, we will need to formalize the notion of “computer” or “machine.” The course outline breaks naturally into three parts:

1. Models of computation (Automata Theory)
 - Finite automata
 - Push-down automata
 - Turing machines
2. What can we compute? (Computability Theory)
3. How efficiently can we compute? (Complexity Theory)

Specifically, the topics covered will include regular languages (finite automata, regular expressions), nonregular languages, context-free languages (context-free grammars, pushdown automata), non-context-free languages, Turing machines and variants, Church-Turing Thesis, undecidability, reducibility, time complexity, and complexity classes P, NP, and NP-complete.

Student Course Outcomes

The student course outcomes of the course are to

- Understand the capabilities and limitations of different models of computation.
- Understand what problems can be solved by computation and which cannot.
- Understand what problems can be solved efficiently and those for which there is no known efficient solution.

The specific learning objectives are that after completing the course, students will be able to

- Classify a particular language as regular, context-free, decidable, Turing-recognizable or non-Turing-recognizable.
- Provide a finite automaton and regular expression for a regular language.
- Prove that a nonregular language is not regular.
- Provide a context-free grammar and pushdown automaton for a context-free language.
- Prove that a non-context-free language is not context-free.
- Provide a description of a Turing machine for a decidable language.
- Prove or disprove closure properties (under union, intersection, complementation, Kleene star) of classes of languages.
- Prove that certain languages are undecidable or non-Turing-recognizable.
- Understand nondeterminism and its role in computation and complexity theory.
- Understand the significance of complexity classes P, NP and NP-complete, and carry out some NP-completeness reductions.

Textbook

Michael Sipser, *Introduction to the Theory of Computation, Third Edition*. Course Technology, 2012, ISBN-10: 113318779X, ISBN-13: 978-1133187790. We will cover Chapters 0–5, and 7, following the schedule given at the end of this handout. Earlier editions of the Sipser book are also acceptable for this class, although the page numbers and sections of the book referenced in the notes and assignments may differ, and some terminology has changed.

Prerequisites

Before taking CS 341, you must complete all of the following with grades of C or better:

1. A 100-series general undergraduate required course in CS
2. CS 241 (Foundations of Computer Science I)
3. CS 280 (Programming Language Concepts).

Grading

Your course grade will be determined by one homework assignment (HW EL), three programming assignments, one midterm and a final exam, which is cumulative. All exams will be closed book and closed notes. The midterm and final exams will be 2.5 hours long. Unless notified otherwise, the dates of the exams and the due dates for the programming assignments are as given in the schedule at the end of this document. Your course grade will be determined by the following weighting scheme:

HW EL	10%
Programming Assignments	30%
Midterm	30%
Final Exam	30%

For each programming assignment, students who do not turn in a minimally working program will get a 0 for the assignment *and* have their course grades at the end of the semester lowered by one step, e.g., from B to C+, or from C to D. For example, if you do not turn in a minimally working program for two assignments, your course grade will be lowered by two steps, e.g., from B to C or from C to F.

Course grades will be assigned on a curve using the following approach. First, I will rank everyone using the cumulative scores with the weights given above, and then assign *preliminary grades* based on that. The top group of students will get a preliminary grade of A, the next group will receive a preliminary grade of B, etc. **Any student who scores less than 20 out of 100 on the final will automatically receive an F for the course.**

After assigning preliminary grades, I will make adjustments for those who did not turn in minimally working programs. For each programming assignment for which you did not turn in a minimally working program, your preliminary grade will be lowered by one step. For example, if your preliminary grade was B and you only turned in one minimally working program out of the three programming assignments, then your course grade is C+; if you did not turn in a minimally working program for two assignments, then your course grade drops to a C; if you did not turn in a minimally working program for three assignments, then your course grade drops to a D. If you turned in minimally working programs for all programming assignments, then your course grade is your preliminary grade based on the ranking of cumulative scores.

Exams

There will be one midterm, which will be given on a Saturday around the end of the 7th week of the semester; tentatively, the midterm date/time is Saturday, 3/8, 9:30am - 12:00pm NJ local time. The final exam will most likely be on Saturday, 5/10, 9:30am - 12:00pm NJ local time. More details of the exact dates, times and locations of the exams will be given during the semester.

If you live within a 75-mile radius of the NJIT Newark campus, you must take the midterm and final exam there on the date and time announced for each. If you live outside of the 75-mile radius of the NJIT Newark campus, then you need to nominate a proctor in Homework Assignment EL.

You must bring 2 forms of photo ID to all exams. All exams will be closed book and closed notes. Also, calculators will not be allowed nor will they be needed.

Course Materials

All of the course handouts (including lecture notes and assignments) are available in PDF format through my CS 341 homepage, whose address is given on the first page. To read the files, you will need to use a software package called Adobe Reader, which you can download for free using a link from the course web page.

Lectures

As a student in an eLearning course, you will not be attending traditional “face to face” lectures but instead will watch on your computer a collection of “modules” that I recorded. The modules are videos of the lecture notes with voiceovers, and they cover the same material that I teach in a face-to-face version of CS 341. All the modules are available as free video podcasts from the NJIT iTunes webpage:

<http://itunes.njit.edu>

You will need iTunes, which you can download from a link on the above webpage, installed on your computer to view the videos. Once you have iTunes installed, to access the lecture videos, you must start up iTunes through the NJIT iTunes page. After logging into iTunes through the NJIT iTunes webpage, go to Store → iTunes Store, and you should see a link in iTunes for CS 341 in the Course Lectures section. The preface of the lecture notes describes how the recorded modules correspond to the pages in the lecture notes.

moodle

For this eLearning class, we will be using a computer-based group-communication system called moodle, which you can access at

<http://moodle.njit.edu/>

The website also provides instructions on using moodle. You must visit the moodle each day as I will post important announcements there in the News forum. Also, I encourage you to post general questions about the course or material there; if you don’t understand some concept or find something confusing, it’s likely that other students have the same problem, so having questions posted and answered in an open forum will benefit everyone in the class. I will respond to all questions posted on moodle.

All HW and programming assignments must be submitted through moodle.

Course Policies

As a general rule, I do not give makeup exams or quizzes, I do not allow allow students to take exams or quizzes on alternate dates, nor do I allow students to turn in assignments late. Of course, if someone has a legitimate reason (e.g., jury duty, serious medical problem, conflict with a religious holiday), I will make allowances as long as you **contact me**

beforehand (whenever it is feasible to do so) and provide proper documentation (e.g., a doctor's note) to the Dean of Students, who will let me know if your absence can be excused. I will not accept excuses such as having too heavy a workload, having too many exams the same week, or simply forgetting.

If upon getting back graded material (e.g., exam, programming assignment) you think that you deserve more points on a particular part, I will regrade the entire thing. Thus, you may get more points on the one part, but you may lose points on other parts. **You must ask for a regrade within one week of when the graded material is returned to the class or you are informed of your score, whichever is earlier.**

There are no extra-credit assignments available. So you need to make sure that you perform well on the assignments and exams.

All portable electronic devices, such as cellphones and laptops, must be turned off during exams.

Students must check the course homepage, moodle, and email each day. I will make announcements there.

Students will be informed of any modifications or deviations from the syllabus throughout the course of the semester.

Homework Assignments

You must download the homework assignments from the course webpage. Except for HW EL, do not turn in any of the homework assignments.

HW EL is required and will count towards 10% of your final grade. If you do not complete HW EL by the due date given in the semester schedule below, you will lose all of the points for the assignment.

All homework solutions are posted on the web at the beginning of the semester. However, it is important that you try to work out the problems on your own without looking at the solutions first because this is the only way you will learn the material.

When working on the homework problems, be sure to show all work and give reasons (e.g., proofs) for your answers. If your proof relies on a theorem or result from the book, be sure to either state the theorem number or page number from the book. Writing out complete solutions will help you prepare for the exams, which often contain questions that are either straight from the homework or slight variations. Thus, it is important to do the homework.

Programming Assignments

The programming assignments are to be submitted by the due date/time. Late programming assignments will be penalized at a rate of 25 points (out of 100) per 24-hour period. For example, since Programming Assignment 1 is due by 2:30pm NJ local time on 2/18, you must turn it in by that date/time to not be penalized. If you turn it in after the due date/time but up to 24 hours later, then you will automatically lose 25 points of the assignment. If you turn in an assignment between 24 hours and 48 hours late, then you will automatically lose 50 points of the assignment. If you turn in an assignment between 48 hours and 72 hours late, then you will automatically lose 75 points of the assignment. No assignments over 72 hours late will be accepted.

After the first two weeks of lectures, we will have covered enough material for you to do the first program. Expect to spend at least 5–10 hours on each programming assignment, so do not wait until the last minute to try to complete it.

Academic Integrity

Students must obey the University Code on Academic Integrity, which you can access from

<http://www.njit.edu/academics/integrity.php>

Any student caught violating this will be reported immediately to the Dean of Students. Violations include, but are not limited to,

- communicating with others during exams
- using unauthorized materials during exams
- copying/giving a computer program from/to another person.

Semester Schedule

Unless I announce otherwise, the schedule for the semester is as below. Although you do not need to turn in the homework, you should complete the assignments according to the schedule below. The reading assignments are from the course textbook.

Week	Topic	Modules	Reading	Homework	Complete By
1	Intro, Languages	0, 0a to 0f	Chapter 0	HW 1	1/28
2	Regular Languages, DFA	0g to 1c	Chapter 1	HW 2	2/4
3	NFA, Reg Exp, Kleene's Thm	1d to 1i	Chapter 1	HW 3	2/11
4	Nonregular Lang, CFL	1j to 2a	Chapter 1, 2	HW 4	2/18
5	CFG, PDA	2b to 2g	Chapter 2	HW 5	2/25
6	Non-CFL, Turing Machines	2h to 3c	Chapter 2, 3	HW 6	3/4
7	Church-Turing Thesis	3d to 3i	Chapter 3	HW 7	3/11
8	Decidability	4a to 4e	Chapter 4	HW 8	3/25
9	Halting Problem, Reductions	4f to 5c	Chapter 4, 5	HW 9	4/1
10	Undecidable Problems, Big-O	5d to 5f, 7a, 7b	Chapter 5, 7	HW 10	4/8
11	Time Complexity, Class P	7c to 7h	Chapter 7	HW 11	4/15
12	Classes P and NP	7i to 7m	Chapter 7	HW 12	4/22
13	Class NP-Complete	7n to 7r	Chapter 7	HW 13	4/29
14	Review	Review1 to 6			5/6

Important Dates

The dates/times of exams and assignments are as below, subject to change.

- 2/4: HW EL due
- 2/18, 2:30pm NJ local time: Programming Assignment 1 due

- Saturday, 3/8, 9:30am - 12:00pm NJ local time: Midterm
- 3/25, 2:30pm NJ local time: Programming Assignment 2 due
- 4/29, 2:30pm NJ local time: Programming Assignment 3 due
- Saturday, 5/10, 9:30am - 12:00pm NJ local time: Final Exam