COT 4210-01 – Discrete Structures II - Fall 2015 Syllabus

Class Time: Tuesday, Thursday 1:30-2:45pm **Class Location:** MSB - 359

Lecturer: Arup Guha
Office: HEC – 240
Email: dmarino@cs.ucf.edu
Phone Number: 407-823-1062
Office Hours: TBA (http://www.cs.ucf.edu/~dmarino/ucf/OH/html)
Course Web Page: http://www.cs.ucf.edu/courses/cot4210/fall2015/
Teaching Assistant: Kevin Joslyn
TA Office Hours & Location: HEC-308, TR 10:30 - 12:30 pm

Course Objective

The true title of this course should be "Theory of Computation." We will examine different models of computation and determine their computational ability. For each model, we will look at specific problems that can be solved within a particular model of computation and other problems that can not be solved within that model. Our ultimate model will be the Turing Machine, which as far as any one knows, is a model as powerful as any computer that can be built, at least in terms of types of problems it can solve. (It's possible for a standard computer to solve problems *faster* than they could be solved on certain Turing Machine models, for example.) We will study which problems are solvable by a Turing Machine and which ones are not. Furthermore, we will differentiate between problems that can be solved on a standard Turing Machine in Polynomial time versus problems that can be solved on a non-deterministic Turing Machine in Polynomial time. For many of the latter problems, it is not known whether or not a standard Turing Machine exists to solve them in Polynomial time also. This background will lead to a solid understanding of the P=NP? question. Finally, we will briefly discuss the practical ramifications of the complexity class NP.

Recommended Textbook

Introduction to the Theory of Computation Third Edition by Sipser ISBN-13: 9781133187790
(Note: You may obtain the first or second edition as well. The content covered in this course is adequately covered in both versions.)

Grading

The final letter grade will be based upon the items listed below. Study group meetings will be graded based on write-ups about the meetings submitted by each group. Programming assignments (3) will be due over WebCourses. All programs must be done in Java and must be done INDIVIDUALLY. The essay (both rough and final draft) will be submitted via WebCourses. Quizzes and the final exam will occur in class. Note: plus/minus grades will be issued, when deemed appropriate.

Item	Number	Percentage of Grade	Total Percentage
Study Group Meetings	6	2	12
Programming Assignments	3	4	12
Essay	1	10	10
Quiz 0 (Fin Aid Asgn)	1	1	1
Quizzes	6	5	30
Final Exam	1	35	35

The goal of these problems is to help you learn the material and for challenging material, such as the material in this class, it's helpful to work in groups. Typically, the group as a whole can come to an understanding that no single individual can. Each student must do their own write up in their own words and submissions should not be carbon copies of one another. The act of writing without copying actually helps strengthen neural connections relevant to learning. Groups should meet weekly and each group member should do some work on their own before the meeting so that the meeting can be effective.

Community Service Opportunity

In lieu of one of the programming assignments, you may perform 5 hours (or more) of community service **with a registered 501 (c)(3) organization.** This option will replace your lowest grade on a programming assignment with a 100. In order to get this credit, you must complete the community service and turn in the requisite form signed directly to me (in my hand) by **December 3, 2015 in class.**

Study Groups, Homework and Quizzes

Questions for this course are difficult to create. As such, nearly all the homework questions I use have solutions easily found on the internet. Much to my chagrin, students in past semesters (a majority of the class), just copied from these solutions rather than use them as valuable tools for learning. My research has shown me that collaboration in study groups run the right way can greatly aid student learning. As such, I haven't given up on this potentially useful idea. This semester I'll ask the students to form the groups instead of me choosing them. I will match any students left without a group. All groups must contain in between 2 and 4 students, inclusive. All groups must be set by the end of class on September 1st. If you haven't chosen your own group by then, I will assign you one. **Note: Quiz #0 will be one question and will be used for the financial aid requirement. Please see me if you can't make it to class that day.**

I will post all of my past homework questions and solutions to use as study guides. Groups will be required to meet all together (either physically or virtually via Skype or another resource) at least once before each quiz. The group will have to produce meeting notes detailing what the group did in preparation for the upcoming quiz. Groups are encouraged to use any online resources in addition to my posted homework questions and solutions. I will give ideas of how I believe the groups ought to run in class. Each group is required to turn in a summary of their meeting(s) (groups may meet more than once before each quiz) at the beginning of each quiz. Credit will be given based on the quality of the notes as well as the quality of the study session(s) described. More importantly, if the groups use their time wisely, it's likely that all students in the group will do better on the quiz, which ought to be the real reason to take the group meetings seriously.

Programming Assignments

Most instructors don't give programming assignments with this course, but I feel since most CS students enjoy coding, programming assignments are an excellent way of tangibly showing the relevance of the material to students. I will require that Java be used for the assignments for the ease of grading and because Java tends to do a better job of helping programmers catch errors. To incentivize correctness, for each assignment at least 50% of the points will simply be based on correctness on test cases. In addition, any program that crashes, no matter how trivial the reason, will be given a 70% or lower.

Essay

I've found that this class is a wonderful forum to give students practice in argumentative writing. In many workplace situations, the ability to articulate a point of view and provide justification for it is often more important than just writing code. Rarely do CS students get an opportunity to do this. After participating in a Writing across the Curriculum grant, I decided to add an essay to the course with great results. Students all got practice with argumentative writing and most students improved their writing and communication skills, which I feel are very important. Typically, improvement occurs when students get an opportunity to improve their essay based on feedback. Thus, you will turn in a rough draft, which will be worth 3% of the course grade, get my feedback on that and then have an opportunity to make improvements and submit a final draft, worth 7% of the course grade.

Quizzes

Each quiz will only test 1 chapter worth of material (except quiz 3, since chapter 3 is very short). As such, no aids will be allowed for the quizzes and will be given the last 45 minutes of class.

Final Exam

Each student will be allowed 4 sheets of notes (typed or written), front and back, on 8.5" x 11" paper as exam aids. The exam is worth a high percentage of the course grade and is ***the cumulative benchmark*** of the whole course. Due to the exam's weight, no student will be able to earn an A in the course with a poor performance on the final exam. ***Students must earn a 40% on the exam to earn a C or higher in the course.***

Tentative Schedule

Week	Tuesday Class	Thursday Class
Aug 24	1.1	1.2, **quiz #0**
Aug 31	1.3	**FOOTBALL GAME**
Sept 8	1.4, **quiz #1**	1.4
Sept 14	Minimal DFAs	2.1
Sept 21	2.2	2.3, **prog #1 due**
Sept 28	3.1, **quiz #2**	3.2, 3.3
Oct 5	4.1	4.2, **prog #2 due**
Oct 12	4.2	5.1, **quiz #3**
Oct 19	5.2	5.3, **prog #3 due**
Oct 26	5.3	7.1, 7.2, **quiz #4**
Nov 2	7.3, **essay rough draft due**	7.4
Nov 9	7.5	7.5
Nov 16	8.1, **quiz #5**	8.2
Nov 23	8.3, **essay final draft due**	**THANSKGIVING**
Nov 30	**quiz #6**	Final Exam Review **COM SERV DUE**
Dec 7	**READING DAY**	**NO CLASS**
Dec 14	**Final Exam (1pm – 4pm)**	

All sections listed in this chart refer to sections of the textbook. Note that this is a **tentative** schedule. Changes may be made in class. For this reason, attending class is important. Also, some material may be pulled from other sources in lecture. Thus, it will be beneficial for you to take your own notes AND read the posted course notes.

Late/Make Up Exams and Assignments

If a student is unable to complete an assignment on time or take an exam on time due to a serious family, medical or work situation, he or she must contact the instructor **BEFORE** the due date and ask for an extension. Extensions will be granted in situations the instructor deems reasonable. If an emergency occurs that prevents contacting the instructor before the due date, then the student should contact the instructor as soon as possible and reasonable accommodations will be made.

Incompletes

Incompletes are reserved for students who have been disabled in the middle of the course (typically for a medical or family reason) and have successfully completed a majority of the course, but will need extra time to get well and complete the remainder of the work. **INCOMPLETES WILL NOT BE ISSUED TO STUDENTS WHO ARE DOING POORLY IN THE CLASS WHO WISH TO RAISE THEIR GRADE.** If you are doing poorly in the course and are worried about not passing, make an appointment with the instructor, and if necessary, drop the course, **BEFORE THE WITHDRAWAL DEADLINE, NOVEMBER 2, 2015!!!**